Optimization Algorithms for Machine Learning

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Anant Raj

aus Lakhisarai, India

Tübingen

2020

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	07.06.2021
Stellvertretender Dekan:	Prof. Dr. József Fortágh
1. Berichterstatter/-in:	Prof. Bernhard Schölkopf
2. Berichterstatter/-in:	Prof. Philipp Hennig

To my *Parents* and my sisters *Anisha* and *Amrita*.

“In exactly the same way, ... scatter your body, your feeling, your perception, your predispositions, your discriminative consciousness, break them up, knock them down, cease to play with them, apply yourself to the destruction of craving for them. Verily, ... the extinction of craving is Nirvana.”

Buddha

“All things appear and disappear because of the concurrence of causes and conditions. Nothing ever exists entirely alone; everything is in relation to everything else” - Buddha

Image Source: pixelbay.com

Preface

The significant portion of the work presented in this dissertation has been carried out between August 2015 and August 2020 at the Max-Planck institute for Intelligent Systems, Tübingen in the Empirical Inference Department headed by Dr. Bernhard Schölkopf. A small but important portion of the work included in this dissertation has been done while I was visiting the group led by Prof. Martin Jaggi at EPFL, Lausanne and by Prof. Francis Bach at SIERRA-Inria, Paris to do research in the theory of convex/non-convex optimization for machine learning.

This thesis is organized overall in 11 chapters out of which there are 3 main chapters which contain introduction, objective and contribution made in this thesis. 8 of the remaining chapters are included in the appendix which have been taken directly from my research manuscripts.

- Chapter 1 is a brief introduction of the topic and provides background on coordinate descent and stochastic gradient descent optimization methods. It also contains a discussion on the major research challenges in mini batch stochastic gradient descent and coordinate descent algorithms.

- Chapter 2 briefly discusses the objective of this thesis. It contains a concise description of the major research questions which have been partially/fully answered in this thesis.

- Chapter 3 contains the specific contributions made in thesis. It further mentions all the research manuscripts included in this thesis and describes the contribution made by me in each manuscript. In the final part of this chapter starting from Chapter 3.4, I discuss the background and main results of each manuscript in a concise manner.

- Appendix A contains the copy of manuscript titled "Screening Rules for Convex Problems" which has been presented in Optimization for Machine Learning Worksop at Neurips 2016, held in Barcelona.

- Appendix B contains the copy of the manuscript titled "Approximate Steepest Coordinate Descent" published at ICML, 2017 held in Sydney, Australia.

- Appendix C contains the copy of the manuscript titled "Safe Adaptive Importance Sampling" published at Neurips, 2017 held in USA.

- Appendix D contains the copy of the manuscript titled "On Matching Pursuit and Coordinate Descent" published at ICML, 2018 held in Stockholm, Sweden.
- Appendix E contains the copy of the manuscript titled "k-SVRG: Variance Reduction for Large Scale Optimization" which is an arXiv Manuscript.
- Appendix F contains the copy of the manuscript titled "A Simpler Approach to Accelerated Stochastic Optimization" published at ICML, 2020 held online.
- Appendix G contains the copy of the manuscript titled "Importance Sampling via Local Sensitivity" published at AISTATS, 2020 held online.
- Appendix H contains the copy of the manuscript titled "Explicit Regularization of Stochastic Gradient Methods through Duality" submitted at AISTATS, 2021.

Acknowledgements

Upon completion of this work, I would like to take this opportunity to express my sincere gratitude to the people who have helped me get through this journey.

First and foremost, I am wholeheartedly thankful to my supervisor Prof. Dr. Bernhard Schölkopf. He has been an extraordinary Ph.D supervisor. His scientific knowledge, constant support, optimism, and outstanding cooperation have helped me to advance my scientific career. His honest reviews of my research ideas and scientific writings has made me a better researcher over the time. I am very grateful to him for providing me an opportunity to pursue Ph.D in the Empirical Inference Department at Max-Planck Institute of Intelligent Systems, Tübingne and for giving me the freedom to pursue my own research agenda as well as to form my own collaborations. I also want to thank Prof. Dr. Philipp Hennig who kindly agreed to accept the role of the reviewer for my Ph.D. dissertation. He has been very supportive in the entire process and has always been helpful.

I am incredibly thankful to my collaborator Prof. Dr. Martin Jaggi. I have learnt the basics of optimization theory from him and have done my first research project in the area of optimization theory under his guidance. Theory of Optimization later became the main topic of my research during my Ph.D. and hence, this thesis wouldn't have been possible without him. He has been a tremendous mentor throughout my Ph.D. I have been going to him with my academic as well as non-academic problems to seek his advice and have always come back wiser after discussing with him.

I would also like to express my deepest gratitude to Prof. Dr. Francis Bach for his support and mentorship towards the ends of my Ph.D. He has been very kind to host me in his research group as a visiting Ph.D student. I have improved my technical skills as well as my capabilities to come up with important research questions while working in his guidance. This experience has inspired me to try to become a good mentor when I start an academic position in future. He has already set the bar too high for young researchers like me who only can hope to touch it someday.

This thesis would be incomplete without thanking my other amazing collaborators, colleagues and friends who have been part of my journey and this thesis would be incomplete without expressing gratitude towards them. I have been going to them with almost all my technical and non-technical problems and I never came back empty-handed. Dr. Stefan Bauer has been one of those people whom I turned to whenever in trouble. No matter what the situation be, he has been always helpful to me. I can not thank him enough for his kind support. I will also have to mention Dr. Rohit Babbar and Dr. Nidhi who had supported me in my initial days of my stay at Tübingen when I knew no one else

here. Since then, I have gone to them asking for advice on anything and everything. I am extremely thankful to them for their support.

I am wholeheartedly thankful to my colleagues from Max-Planck Institute for Intelligent Systems, Tübingen who had to bear my poor jokes and stupidity for such a long time. I will have to write an entire chapter about them if I start to write about everyone who helped me go through this phase. I can not thank enough to my colleagues from IBM Research, Microsoft Research, MLO Lab at EPFL, Google DeepMind and SIERRA-Inria who left no wheels unturned to make me feel one of them even though I was visiting these research labs only for a short duration. I extend my special thanks to my mentors during my internship phase, Dr. Abhishek Kumar and Dr. Youssef Mroueh from IBM Research, Dr. Nicolo Fusi and Dr. Lester Mackey from Microsfot Research, and Dr. Pooria Joulani, Dr András György and Prof. Csaba Szepesvári from Google DeepMind. They have been amazing mentors and I could not have asked for a better experience. During my short stay at IBM Research, Microsoft Research and Google DeepMind, I have learned a lot from them. I wish I could stay longer at all of these places to maximize my interaction with them. During my visit to the MLO lab at EPFL as a visiting PhD student, I was also mentored by Dr. Sebatian Stich. His contribution in shaping my research prospect can not be overlooked. I am thankful to him for everything I have learned from him. For me, it was a great fun collaborating with him. I am also thankful to my other collaborators who became my co-authors for one or more paper. Each one of you have taught me something invaluable for which I am in debt to you. Let's also not forget those collaborators with whom the projects never reached the final destination. Even though the outcomes of these collaboration were not research manuscripts but the learning part was never missing in these projects as well. I am thankful to them to bear with me and for teaching me a great deal about various research topics.

During my stay in Tübingen as a Ph.D student, the city could not be more kind. In this beautiful university town, I have met some exceptionally nice, humble and smart people across different disciplines. I have learned so much about different disciplines from them. This thesis can not be complete without thanking two very special person Dr. Vinod and Dr. Vishal who have supported me in my thick and thin. At the very first sight of a problem, I reached out to them for solutions and have never been disappointed. I have learned a great deal of neuroscience, politics and humanity from them. I wish I could ever become as humble, as helpful and as down to earth as they are. That would however be a long struggle. In this very moment, I also want to be thankful to all the people who have literally fed me. Whenever they cooked, they offered me their food. This list is so long that I will definitely forget some of the names and it would an injustice to them. So, I thank them collectively. I can never repay whatever they have done for me. I am also thankful to my friends here in Tübingen and away (too many to list here but you know who you are!) for providing support and friendship that I needed to survive this period. They made themselves available whenever I needed them. You guys are the unsung heroes of my research career.

Now is the time to show gratitude towards my family who have made all kind of

sacrifices, so that I could come this far. I can never thank my parents and my sisters enough for that. Irrespective of coming from a conservative background, my parents never undermined the value of the education. They always supported me in my decision to pursue higher education. My sisters made the foremost level of sacrifices to ensure my education. They are the real superheroes of my life, they are the dark knights of my life. I always know that no matter what happens, they will always be there for me. Their contribution in my life can not be expressed in words. I can not ask for more from my family, I am very much thankful to them for always being there for me. I am also thankful to all my teachers who have ever taught me. There is a part of everyone of you in me. I have managed to come this far because of the solid foundations you provided. You all will always be there in my heart.

Last but not the least, I am also thankful to those people whose contribution we never really acknowledge, but nothing works without them. They keep doing their job silently so that we don't have to worry about any thing other than research. I am very thankful to our department's secretary Sabrina, all the administrative staffs and staffs from IT support to make my research life in Tübingen very smooth. Even though I have met only few of you, I am thankful to all of you for your contributions in my research life.

Abstract

With the advent of massive datasets and increasingly complex tasks, modern machine learning systems pose several new challenges in terms of scalability to high dimensional data as well as to large datasets. In this thesis, we consider to study scalable descent methods such as coordinate descent and stochastic coordinate descent which are based on the stochastic approximation of full gradient.

In the first part of the thesis, we propose faster and scalable coordinate based optimization which scales to high dimensional problems. As a first step to achieve scalable coordinate based descent approaches, we propose a new framework to derive screening rules for convex optimization problems based on duality gap which covers a large class of constrained and penalized optimization formulations. In later stages, we develop new approximately greedy coordinate selection strategy in coordinate descent for large-scale optimization. This novel coordinate selection strategy provavbly works better than uniformly random selection, and can reach the efficiency of steepest coordinate descent (SCD) in the best case. In best case scenario, this may enable an acceleration of a factor of up to n, the number of coordinates. Having similar objective in mind, we further propose an adaptive sampling strategy for sampling in stochastic gradient based optimization. The proposed safe sampling scheme provably achieves faster convergence than any fixed deterministic sampling schemes for coordinate descent and stochastic gradient descent methods. Exploiting the connection between matching pursuit where a more generalized notion of directions is considered and greedy coordinate descent where all the moving directions are orthogonal, we also propose a unified analysis for both the approaches and extend it to get the accelerated rate.

In the second part of this thesis, we focus on providing provably faster and scalable mini batch stochastic gradient descent (SGD) algorithms. Variance reduced SGD methods converge significantly faster than the vanilla SGD counterpart. We propose a variance reduce algorithm k-SVRG that addresses issues of SVRG [98] and SAGA[54] by making best use of the *available* memory and minimizes the stalling phases without progress. In later part of the work, we provide a simple framework which utilizes the idea of optimistic update to obtain accelerated stochastic algorithms. We obtain accelerated variance reduced algorithm as well as accelerated universal algorithm as a direct consequence of this simple framework. Going further, we also employ the idea of local sensitivity based importance sampling in an iterative optimization method and analyze its convergence while optimizing over the selected subset. In the final part of the thesis, we connect the dots between coordinate descent method and stochastic gradient descent method in the interpolation regime. We show that better stochastic gradient based dual algorithms with fast rate of

convergence can be obtained to optimize the convex objective in the interpolation regime.

Kurzfassung

Das Aufkommen massiver Datensätze und immer komplexerer Aufgaben stellt moderne maschinelle Lernsysteme vor zahlreiche neue Herausforderungen bezüglich der Skalierbarkeit für hochdimensionale Daten und große Datensätze. In dieser Arbeit betrachten wir die Untersuchung skalierbarer Abstiegsmethoden wie den Koordinatenabstieg und den stochastischen Koordinatenabstieg, die auf stochastischen Approximationen des vollen Gradienten basieren. Im ersten Teil der Arbeit schlagen wir eine schnellere und skalierbare koordinatenba- sierte Optimierung vor. Als ersten Schritt zum Erreichen skalierbarer koordinatenbasierter Abstiegsansätze schlagen wir einen neu- en Ansatz zur Ableitung von Screening-Regeln für konvexe Optimierungsprobleme vor, der auf der Dualitätslücke basiert und eine große Klasse Optimierungsproblemen mit Nebenbedingungen abdeckt. Anschließend entwickeln wir eine neue Auswahlregel für die Koordinatenauswahl in Koordinatenabstiegsverfahren für große Datensätze. Die Konvergenz dieses Algorithmus ist nachweislich schneller als für stochastischen Koordinatenabstieg und kann die Konvergenzrate des steilsten Koordinatenabstiegs (SCD) erreichen, was eine Beschleunigung um einen Faktor von bis zu n, der Anzahl der Koordinaten, ermöglicht. Weiterhin schlagen wir adaptive Sampling-Strategie für die stochastischen gradientenbasierten Optimierung vor. Das vorgeschlagene safe Sampling erreicht nachweislich eine schnellere Konvergenz als alle festen deterministischen Sampling-Strategien für Koordinatenabstiegs- und stochas- tische Gradientenabstiegsmethoden. Unter Ausnutzung des Zusammenhangs zwischen matching Pursuit, bei der ein verallgemeinerter Richtungsbegriff verwendet wird, und greedy Koordinatenabstieg, bei dem alle Bewegungsrichtungen orthogonal sind, schlagen wir eine einheitliche Analyse für beide Ansätze vor und erweitern sie, um die bessere Konvergenzrate zu erhalten. Im zweiten Teil dieser Arbeit beschäftigen wir uns mit der Diskussion von beweisbar schnelleren und skalierbaren Algorithmen für den stochastischen Abstieg in Mini-Batches (SGD-Algorithmen). Varianzreduzierte SGD-Methoden konvergieren deutlich schneller als das Standard SGD Pendant. Wir schlagen einen varianzreduzierenden Algorithmus k-SVRG vor, der die Probleme von SVRG [99] und SAGA[54] angeht, indem er den verfügbaren Speicher optimal nutzt und die Stalling-Phasen ohne Fortschritt minimiert. In einem späteren Teil der Arbeit diskutieren wir ein einfaches Framework, das die Idee der optimistischen Updates nutzt, um schnellere stochastische Algorithmen zu erhalten. So erhalten wir sowohl einen schnelleren varianzreduzierten Algorithmus als auch einen schnelleren allgemeinen Algorithmus. Darüber hinaus untersuchen wir ein auf lokaler Sensitivität basierenden Sampling-Schemas für eine iterative Optimierungsmethode und analysieren dessen Konvergenz während der Optimierung über die ausgewählte Teilmenge. Im letzten Teil der Arbeit verbinden

wir Koordinatenabstiegsmethoden und sto- chastische Gradientenabstiegsmethoden im Interpolationsregime. Wir zeigen, dass wir auf stochastischen Gradienten basierende Algorithmen für das duale Problem mit schnellerer Konvergenzrate definiren können, die eine konvexe Funktion im Interpolationsregime minimieren.

Contents

Chapter 1

Introduction

Intelligent systems and machines have become an integral part of the modern civilization. These machines/systems are now being heavily utilized in various tasks which are useful in our daily life for example *search engines like google, recommendation systems, autonomous cars* and *robots*. The performance of these systems have improved a lot on these tasks which were considered hard to perform for machines a decade ago. Modern machine learning algorithms and the advent of modern data collection methods are the most essential building blocks in building these intelligent systems. On a negative side, modern intelligent systems are data hungry and require a huge amount to data to train the predictive model. Hence, for the same reason, modern machine learning applications also face new obstacles in terms of scalability and efficiency of the algorithms for huge scale data, privacy of the data and other ethical concerns. Addressing these challenges is very critical to the advancement of machine learning and artificial intelligence. There are two major steps of developing modern machine learning applications : (i) choose a model that approximately generates the observable data from underlying data distribution. (ii) learn model parameters using the finitely observable data which are often obtained after minimizing a finite sum objective. Numerical optimization lies at the heart of second step. The goal of this thesis is to develop fast and efficient mathematical optimization methods especially first order stochastic gradient based optimization methods and coordinate descent methods to address problems in modern ML applications.

1.1 Convex Machine Learning Problems

For the purpose of our discussion and introduction, let us consider the classical problem of hinge loss SVM classification. The samples $\{(\mathbf{a}_i, y_i)\}_{i=1}^n$ where $\mathbf{a}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ for all $i \in [n]$ are independent and identically distributed samples from the joint data distribution which form the dataset where $\mathbf{a}_i$ is referred to as feature vector and y_i the corresponding class label in the case of classification task or output in the case of regression task. The optimization problem for hinge loss SVM classification is written as

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{a}_i^\top \mathbf{x}) + \frac{\lambda}{2} \|\mathbf{x}\|^2. \tag{1.1}$$

The term $\max(0, 1 - y_i \mathbf{a}_i^\top \mathbf{x})$ which is popularly known as hinge loss. It is the loss with respect to the i^{th} sample. The remaining term is referred as regularizer which controls the capacity of the optimal solution. However, for different tasks, different loss functions and different regularizers are used. The more general formulation of machine learning optimization problem can be written as :

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \mathbf{a}_i^\top \mathbf{x}) + h(\mathbf{x}) \tag{1.2}$$

where ℓ is the loss function and $\ell(y_i, \mathbf{a}_i^\top \mathbf{x})$ measures the loss occurred in the prediction of y_i using the model parameter x and features z_i. Further, the problem described in Equation (1.2) can be seen as a subset of far more general optimization formulation given below,

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{x}). \tag{1.3}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$. The optimization problem in Equation (1.3) is widely referred as finite sum optimization problem. In this thesis, we focus on optimization problems with convex objectives that means ℓ and f_i's are convex functions. Our main focus is to optimize convex functions because of the fact that convex problems are their ubiquitous in application and are relatively easy to solve provably. Some examples of the problems which have convex objectives are Logistic regression, least-squares, support vector machines, conditional random fields and tree-weighted belief propagation. Also, the optimization techniques developed to optimize convex function often works well for the task of non-convex optimization as well. Even though, global convergence is NP-hard for non-convex optimization, optimization methods developed for optimizing convex function often provably reach stationary point of non-convex optimization as well. From the empirical perspective, many optimization algorithms which have fast rate of convergence while optimizing convex functions have also good empirical performance while optimizing non-convex functions.

First-order numerical optimization methods (i.e., based on first order information of the function) are particularly methods of choice while solving the optimzation problems discussed in Equations (1.2) and (1.3) due to their scalable nature as well as for their provable convergence guarantee. Gradient descent, stochastic gradient descent and randomized coordinate descent methods are amongst the most widely used optimization methods. As the modern data collection methods have improved, the size of dataset used while training machine learning models have increased tremendously and more often than not computing full gradient on the entire dataset for making single first order update is computationally a very expensive task to perform. This computational difficulty boosted the use of stochastic first order methods while optimizing the objective function given in Equations (1.2) and (1.3). *Stochastic Gradient Descent* and *Randomized Coorrdinate*

Descent based methods are the most widely used stochastic first order methods to optimize the objectives arising while training machine learning models. The original stochastic approximation approach was proposed by Robbins and Monro [206] and since then a large amount of work has been done to understand and as well as to improve stochastic first order optimization methods [see, e.g., 158, 165, 191, 192, and references therein]. In past few years, Coordinate descent (CD) methods have become very popular and attracted a vast interest in the optimization community [170, 204]. Due to their computational efficiency, scalability, state of the art empirical performance as well as their ease of implementation, stochastic gradient methods and coordinate descent methods are the most commonly used optimization algorithms in machine learning and signal processing applications [74, 93, 262]. In thesis, we will be mostly focussing on improved algorithms for stochastic gradient descent algorithms and coordinate descent methods. Before going into the detail of stochastic gradient methods and coordinate descent methods, we will first discuss the relevant background.

1.2 Background

Before going into further details, we briefly review the background which will be required to discuss for this thesis. This includes common assumptions, definitions and notations which would be used in most part of the thesis except where it is mentioned otherwise. As discussed previously, let us consider the general finite sum optimization problem as presented in equation (1.3),

$$\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})\,.$$

Throughout this thesis, we will be working with smooth functions until and unless specified. We would first define the smoothness of the function as following:

Definition 1.2.0.1 (L-smooth functions [173])**.** We say a function $f:\mathbb{R}^d\to\mathbb{R}$ is L-smooth if there exists constant L such that the gradient of the function f is L-Lipschitz *i.e.*

$$\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\|\leq L\|\mathbf{x}-\mathbf{y}\|\ \forall\, x,y\ \in\mathbb{R}^d.$$

Smoothness assumption is very common in the analysis of first-order optimization methods. In some parts of the thesis, we will also require to assume that all the components of finite sum optimization f_i's are L_i smooth for $i\in[n]$. The definition in 1.2.0.1 also give rise to the following condition which is utilized in the convergence analysis of first order methods:

$$f(\mathbf{x})\leq f(\mathbf{y})+\langle\nabla f(\mathbf{y}),\mathbf{x}-\mathbf{y}\rangle+\frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2\,.$$

In the similar spirit, next we define the strong convexity of a function below.

Definition 1.2.0.2 (μ-strongly convex functions [173])**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be μ-strongly convex if there exists a constant $\mu > 0$ such that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2 .$$

For an L-smooth and μ strongly function f, the quantity $\kappa = \frac{L}{\mu}$ is known as the condition number of f . When the strong convexity parameter μ is 0 then the function f is said to be non-strongly convex. It is well studied that first order methods converge faster on strongly convex and smooth function and the rate of convergence depends on the condition number κ. Many popular regularizers $h(\mathbf{x})$ (Eq. (1.2)) used in machine learning problem formulation are usually strongly convex in nature which essentially makes the entire objective in Eq. (1.2) strongly convex and thus easier to optimize.

The *proximal operator* of a function g is defined as

$$\text{prox}_{\eta g} := \arg\min_{\mathbf{y} \in \mathbb{R}^d} \left(g(y) + \frac{1}{2\eta} \|\mathbf{y} - \mathbf{x}\|^2 \right)$$

for some parameter $\eta > 0$. Proximal Operators are useful while dealing with nonsmooth optimization *i.e.*, problems where the objective function is not differentiable at finitely many points. The non-differentiability usually comes from the regularization function $h(x)$. Proximal operators are the most easy to use in practice when (i) the objective function can be written as a sum of smooth $(f(\mathbf{x}))$ and a nonsmooth function $(h(\mathbf{x}))$ and (ii) the computation of proximal operator of the non-smooth part is easy.

1.2.1 Convergence Criteria

In most part of thesis, we would be devising provably convergent optimization algorithms. For convex functions, it is common practice to use primal optimality gap $f(\mathbf{x}) - f(\mathbf{x}^\star)$ or the distance between the iterate $\mathbf{x}$ to the optimal iterate $\mathbf{x}^\star$ *i.e.* $\|\mathbf{x} - \mathbf{x}^\star\|$ as the convergence criterion where $\mathbf{x}^\star = \arg\min f(\mathbf{x})$. In some cases, a *Lyapunov function* is defined combining the two criteria discussed above ($f(\mathbf{x}) - f(\mathbf{x}^\star)$ and $\|\mathbf{x} - \mathbf{x}^\star\|$) which further is used as a convergence criterion. However, any criterion which involves the global optimal point $\mathbf{x}^\star$ can not be used for the non-convex case. It is suggested [76] to use $\|\nabla f(\mathbf{x})\|^2$ as the convergence criteria while analyzing the convergence for non-convex functions.

Definition 1.2.1.1 (ε-accurate point [198])**.** A point $\mathbf{x}$ is called ε-accurate if $\|\nabla f(\mathbf{x})\|^2 \leq \varepsilon$. For the stochastic iterative algorithm, we say a point $\boldsymbol{x}$ ε-accurate if $\mathbb{E}\|\nabla f(\mathbf{x})\|^2 \leq \varepsilon$.

The ε-accurate point based criterion is applicable to both while optimizing convex as well non-convex objectives. However, another more conservative criterion can be used for convex objectives which we define below.

Algorithm 1 Randomized Coordinate Descent[170]

Input: f, $\boldsymbol{x}_0$, T
for $t = 0$ **to** T **do**
  Choose index i_t with uniform probability
  Set $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x} - \eta_t [\nabla f(\boldsymbol{x}_t)]_{i_t} \mathbf{e}_{i_t}$ for $\alpha > 0$.
end for

Definition 1.2.1.2 (ε-suboptimal point [198])**.** A point $\mathbf{x}$ is called ε-suboptimal point if $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \varepsilon$ where $\mathbf{x}^\star = \arg\min_{\mathbf{x}\mathcal{X}} f(\mathbf{x})$ and $\mathcal{X}$ is the domain of $\mathbf{x}$. For the stochastic iterative algorithm, we say a point $\boldsymbol{x}$ ε-accurate if $\mathbb{E}[f(\mathbf{x})] - f(\mathbf{x}^\star) \leq \varepsilon$.

Now that we have discussed the general background to understand the result presented in our paper, we will discuss coordinate descent and (mini batch) stochastic gradient descent in the next two sections where the most of contributions are made in this thesis.

1.3 Coordinate Descent

Coordinate descent algorithms performs successive approximate minimization along coordinate directions or coordinate hyperplanes to optimize an objective function. Recently, the rate of convergence for randomized coordinate descent and randomized accelerated coordinate descent was proved in Nesterov [170]. Let us consider the following unconstrained optimization problem,

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \tag{1.4}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a continuous convex function. $[\mathbf{x}]_i$ denotes the i^{th} entry of vector $\mathbf{x}$ and $\mathbf{e}_t$ denotes a d-dimensional vector with all the entries set to be zero except $[\mathbf{e}]_t = 1$. In algorithm 1, we describe a simplest version of randomized coordinate descent method for unconstrained optimization of smooth function f. The convergence analysis for the Algorithm 1 was proposed in [170, 262]. For the analysis of Algorithm 1, coordinate wise Lipschitz continuous gradient is define which is the main key in the analysis. A convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ with coordinate-wise L_i-Lipschitz continuous gradients satisfies $|\nabla_i f(\boldsymbol{x} + \eta \boldsymbol{e}_i) - \nabla_i f(\boldsymbol{x})| \leq L_i |\eta|, \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \eta \in \mathbb{R}$ for constants $L_i > 0$, $i \in [n] := \{1, \ldots, n\}$ which essentially satisfies by the standard reasoning

$$f(\boldsymbol{x} + \eta \boldsymbol{e}_i) \leq f(\boldsymbol{x}) + \eta \nabla_i f(\boldsymbol{x}) + \tfrac{L_i}{2} \eta^2 \tag{1.5}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\eta \in \mathbb{R}$. A function is coordinate-wise L-smooth if $L_i \leq L$ for $i = 1, \ldots, n$. A simplified version of the result presented in Nesterov [170] was presented in Wright [262] which we restate here below and discuss the implications before addressing the challenges in coordinate descent algorithms.

Theorem 1.3.0.1 (Theorem 1, [262]). *Suppose that the function f is convex and uniformly Lipschitz continuously differentiable, and attains its minimum value $f^\star$ on a set $\mathcal{S}$. Also, there exists a finite R_0 such that the level set for f defined by $\mathbf{x}_0$ is bounded, that is*

$$\max_{\mathbf{x}^\star \in \mathcal{S}} \max_{\mathbf{x}} \{\|\mathbf{x} - \mathbf{x}^\star\| : f(\mathbf{x}) \leq f(\mathbf{x}_0)\} \leq R_0$$

then given than $\eta_t = \frac{1}{L}$ in Algorithm 1, we have

$$\mathbb{E}[f(\mathbf{x}_t)] - f^\star \leq \frac{2nLR_0^2}{t}.$$

When f is $\mu > 0$-strongly convex as well then

$$\mathbb{E}[f(\mathbf{x}_t)] - f^\star \leq \left(1 - \frac{\mu}{nL}\right)^t (f(\mathbf{x}_0) - f^\star).$$

Proof Sketch. The detailed proof is given in Wright [262]. However, the key aspect of the proof comes from the coordinate wise Lipschitz condition.

$$\begin{aligned} f(\mathbf{x}_{t+1}) &= f(\boldsymbol{x} - \eta_t [\nabla f(\boldsymbol{x}_t)]_{i_t} \mathbf{e}_{i_t}) \\ &\leq f(\boldsymbol{x}) + \frac{1}{L}[\nabla f(\mathbf{x}_t)]_{i_t}^2 - \frac{1}{2L}[\nabla f(\mathbf{x}_t)]_{i_t}^2 \\ &\leq f(\boldsymbol{x}) - \frac{1}{2L}[\nabla f(\mathbf{x}_t)]_{i_t}^2. \end{aligned}$$

Second last equation comes from using the coordinate wise smoothness condition and putting $\eta_t = \frac{1}{L}$. Now, one can take expectation of both sides over random index i_t to get

$$\mathbb{E}[f(\mathbf{x}_{t+1})] \leq f(\boldsymbol{x}) - \frac{1}{2nL}\|\nabla f(\mathbf{x}_t)\|^2. \tag{1.6}$$

Inequality in Equation (1.6) provides a lower bound on the decrease in the objective function in each iteration. This expression resembles the gain in gradient descent. Rest of the proof follows the proof technique for convergence of gradient descent as in [173]. □

Similar results can be obtained for randomized coordinate descent with composite objective as well as for accelerated version of randomized coordinate descent. The above described result holds when the sampling probability of each and every coordinate is kept fixed and same which is $\frac{1}{n}$. In Nesterov [170], the authors have provided the convergence with L_i-based sampling of coordinates that means probability of selecting i^{th} coordinate $p_i = \frac{L_i}{\sum_{i=1}^n L_i}$. L_i-based sampling as studied in Nesterov [170] improves the performance of randomized coordinate descent algorithm in theory and practice over uniformly random sampling.

1.3.1 Challenges in Randomized Coordinate Descent

Now that we have discussed the background related to randomized coordinate descent, we will now discussed the issues related to coordinate descent algorithms. Some of these issues, we will try to address in this thesis.

It is clear from the result presented in Theorem 1.3.0.1 that the rate of convergence depends on the number of active coordinates. Hence, as the number of active coordinates increases, the algorithm converges slowly to the optimal solution. In the case of high dimensional sparse optimization problem, the subspace where the optimal solution lies is really small that means most of the entries in the optimal solution vector are zeros. If we denote the number of non-negative entries in d-dimensional optimal solution vector as s then more often than not $s << d$. In that case, paying for all the variables in computations as well as in convergence bound is suboptimal as we know that only very few number of variables are non-zero in the end. Hence, one essential goal of sparse optimization problem while applying coordinate descent algorithm is to screen out as many variables as possibles during the course of the optimization process or as a part of preprocessing step which are guaranteed to be zero at the optimal point. The process is termed as *Screening of Variables* [77] in the literature.

As discussed previously, it was shown in Nesterov [170] that L_i-based sampling achieves better rate of convergence in theory and in practice for coordinate descent algorithms. However, another open question remains that can we design an improved time dependent or fixed sampling scheme which would further improve the performance of coordinate descent algorithm further ? If at all it is possible to design a better time dependent coordinate sampling scheme for coordinate descent algorithm, would that be a computationally efficient algorithms to run in practice ? Steepest coordinate descent can also be seen as a time dependent sampling scheme for coordinate descent algorithm. Steepest coordinate descent performs the coordinate selection by choosing the direction which has the maximum coordinate wise absolute gradient value at any instant of time. However, it is pretty clear that to choose the exact steepest coordinate, one has to compute the full gradient and hence, steepest coordinate descent algorithm is a computationally expensive algorithm to run. It has been evident from the experiments that steepest coordinate descent converges fastest to the optimal solution when compared with respect to number of coordinate wise gradient updates has been made. One immediate research question arises from here if one can design a deterministic coordinate selection or coordinate sampling algorithm which carries the best properties from both of the two approaches (i) steepest coordinate descent and (ii) randomized coordinate descent.

1.4 Mini-Batch SGD

Stochastic gradient descent (SGD) algorithms are amongst the most popular optimization algorithms used while training machine learning models. Stochastic gradient descent

(SGD) methods have enabled to run machine learning algorithm on huge scale dataset. Unlike the full gradient methods, stochastic gradient methods does not require to compute the full gradient to make an update instead one has to only compute a stochastic gradient to make an update. Considering the following optimization problem as in equation (1.4)

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}).$$

At any iterate $\mathbf{x}$, we represent $g(\mathbf{x})$ as the stochastic gradient such that $g(\mathbf{x}) = \mathbb{E}[\nabla f(\mathbf{x})]$. The update at ant time instant t, we have following first order update,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \mathbf{g}_t. \tag{1.7}$$

Due the computational scalability of SGD, there has been a lot of works in understanding the convergence guarantee of SGD methods [158, 165, 191, 192]. The variance of the random vector $\boldsymbol{v}$ is defined by $\mathrm{Var}(\boldsymbol{v}) = \mathbb{E}[\|\boldsymbol{v}\|^2] - \|\mathbb{E}[\boldsymbol{v}]\|^2$. The rate of convergence of SGD for Lipschitz convex function is given in the theorem below.

Theorem 1.4.0.1 (Moulines and Bach [158], Nemirovski *et al.* [165]). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a L-Lipschitz convex function and $\mathbf{x}^\star = \arg\min_{\mathbf{x}} f(\mathbf{x})$. Consider an iterate of of SGD update where the estimator g_t has a bounded variance for all t i.e. $Var(\mathbf{g}_t) \leq \sigma^2$. Then for any $T > 1$ and step size $\eta_t \leq \frac{1}{L}$ for all t, SGD satisfies the following guarantee ,*

$$\mathbb{E}[f(\bar{\mathbf{x}}_T)] - f(\mathbf{x}^\star) \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^\star\|^2}{2\eta_T T} + \frac{1}{T}\sum_{t=1}^{t} \frac{\eta_T \sigma^2}{2}$$

where $\bar{\mathbf{x}}_t = \frac{1}{t}\sum_{i=1}^{t} x_i$ for all t.

In the above presented theorem statement, if we choose $\eta_t \approx \mathcal{O}(1/\sqrt{T})$ then we would have,

$$\mathbb{E}[f(\bar{\mathbf{x}}_T)] - f(\mathbf{x}^\star) \approx \mathcal{O}(1/\sqrt{T}).$$

Similarly, if the optimization objective function f is $\mu > 0$ strongly convex function then we get faster rate of convergence as compared to Lipschitz convex function.

Theorem 1.4.0.2 (Lacoste-Julien *et al.* [121]). *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a $\mu > 0$-strongly convex function and $\mathbf{x}^\star = \arg\min_{\mathbf{x}} f(\mathbf{x})$. Consider all the stochastic gradients g_t for all $t > 0$ are bounded i.e. $\|g_t\| \leq G$ then for any $T > 1$, SGD with step size $\eta_t = \frac{1}{\mu t}$ for all t satisfies,*

$$\mathbb{E}[f(\bar{\mathbf{x}}_T)] - f(\mathbf{x}^\star) \leq \frac{G^2}{\mu T}(1 + \log T)$$

where $\bar{\mathbf{x}}_t = \frac{1}{t}\sum_{i=1}^{t} \mathbf{x}_i$ for all t.

In the case of finite sum optimization problem, the stochastic approximation of the gradient g_t is obtained by uniformly random selection of one component amongst n of them (See Eq. (1.3)) with replacement. Another way to make the unbiased stochastic approximation of the gradient is via importance sampling. In Zhao and Zhang [271], the authors analyse the rate convergence of stochastic mirror descent for general sampling distribution p_t for all t. However, there was no new sampling approach proposed in [271] which improves over the existing results of uniform sampling and is computationally cheaper to compute at any time t. If the probability of sampling component i at time instant t is denoted by $p_i^{(t)}$ then SGD with importance sampling makes the following update,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \frac{1}{(np_{i_t}^{(t)})} \nabla f_{i_t}(\mathbf{x}_t). \tag{1.8}$$

Theorem 1.4.0.3 (Theorem 1, [271]). *Let $\mathbf{x}_t$ is be generated by Equation (1.8) and f is L-smooth, $\mu \geq 0$ strongly convex function then if $\eta_t = \frac{1}{\alpha+\mu_t}$ where $\alpha \geq L - \mu$ the following inequality holds for any $T > 1$,*

$$\mathbb{E}[f(\bar{\mathbf{x}}_T)] - f(\mathbf{x}^\star) \leq \frac{1}{T}\left[\alpha\|\mathbf{x}_0 - \mathbf{x}^\star\|^2 + \mathbb{E}\left[\sum_{t=1}^{T} \frac{V_t}{\alpha + \mu t}\right]\right] \tag{1.9}$$

where the variance is defined as $V_t = Var[(np_{i_t}^{(t)})^{-1}\nabla f_{i_t}(\mathbf{x}_t)] = \mathbb{E}\|(np_{i_t}^{(t)})^{-1}\nabla f_{i_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2$.

For the maximum reduction in the objective function, one should choose $p_i^{(t)}$ for $i \in [T]$ which solves the following optimization problem,

$$\min_{p_i^{(t)}, \sum_{i=1}^n p_i^{(t)} = 1} \text{Var}[(np_{i_t}^{(t)})^{-1}\nabla f_{i_t}(\mathbf{x}_t)]. \tag{1.10}$$

One can easily verify that the solution of the above optimization problem is

$$p_i^{(t)} = \frac{\|f_i(\mathbf{x}_t)\|}{\sum_{i=1}^n \|f_i(\mathbf{x}_t)\|}. \tag{1.11}$$

However, it is not practical to compute $p_i^{(t)}$ for all i at any instant t as it requires the computation of the full gradient which is computationally very expensive.

Variance Reduction in SGD: Stochastic gradient descent is the most widely used algorithm for large scale optimization amongst all the optimization methods because of its per iteration computational efficiency. However, the major drawback of SGD optimization algorithm is that it has slow convergence asymptotically due to the inherent variance.

The major breakthrough to accelerate stochastic gradient descent method came when Johnson and Zhang [99] proposed to introduce an explicit variance reduction method for stochastic gradient descent by computing full gradient for snapshot points. This method was named as stochastic variance reduced gradient (SVRG) method. Similar method of variance reduction was proposed in SAGA [54]. SAGA improves upon the previous variance reduced methods like SVRG and SAG by providing tight theoretical convergent rate and extending the analysis to composite objectives where a proximal operator is used on the regularizer. Variance reduced stochastic optimization works on the principle that if the current iterate and previous iterates are close then the gradient information from the previous iterates might be useful in providing better current gradient estimates which will eventually reduce the variance. A unified framework for the variance reduced can be written as in given below. Let $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T$ denote the iterates of the algorithm, where $\mathbf{x}_0 \in \mathbb{R}^d$ is the starting point. For each component f_i, $i \in [n]$, of the objective function, we denote by $\boldsymbol{\theta}_i \in \mathbb{R}^d$ the corresponding snapshot point. The updates of the algorithms take the form

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{x}_t - \eta \mathbf{g}_{i_t}(\mathbf{x}_t), \qquad \text{with} \\ \mathbf{g}_{i_t}(\mathbf{x}_t) &:= \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\boldsymbol{\theta}_{i_t}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_i), \end{aligned} \tag{1.12}$$

where $\eta > 0$ denotes the stepsize, and $i_t \in [n]$ an index (typically selected uniformly at random from the set $[n]$). We now reiterate the result from Defazio *et al.* [54] for strongly convex finite sum optimization. For simplicity, we do not assume the proximal update in this result however, the result also holds for proximal functions.

Theorem 1.4.0.4 (Theorem 1, [54])**.** *With $x^\star$ the optimal solution, define the Lyapunov function T as:*

$$T_k = T(\mathbf{x}_k, \{\boldsymbol{\theta}_i^{(k)}\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta}_i^{(k))} - f(\mathbf{x}^\star) + c\|\mathbf{x}_k - \mathbf{x}^\star\|^2.$$

Then with $\eta = \frac{1}{2(\mu n + L)}$, $c = \frac{1}{2\eta(1-\eta\mu)n}$ and $\kappa = \frac{1}{\eta\mu}$, we have the following expected change in the Lyapunov function between steps of the SAGA algorithm (conditioned on T_k)

$$T_{k+1} \leq \left(1 - \frac{1}{\kappa}\right) T_k.$$

The result presented in Theorem 1.4.0.4 holds for strongly convex function however, the result can also be extended for smooth but non-strongly convex functions.

Sensitivity Based Sampling for Subset Selection: So far we have discussed that how stochastic gradient methods can reduce the computation of a machine learning task.

However, one can also choose to reduce the burden of solving a large scale finite sum optmization by minimizing an approximation to the finite sum objective f formed by independently subsampling data points with appropriate reweighting. Let us consider the optimization objective similar to that of Equation (1.2),

$$\min_{\mathbf{x}} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{a}_i^\top \mathbf{x}) . \tag{1.13}$$

Let us consider a target of sample size m and a probability distribution $\mathbf{P} = \{p_1, \cdots, p_n\}$ over simplex in n-dimension where p_i is the probability of selecting $\mathbf{a}_i$, subsampled Finite Sum Problem of the finite sum problem in Equation (1.13) can be defined as,

$$\min_{\mathbf{x}} \frac{1}{mn} \underbrace{\sum_{i=1}^{m} \frac{f(\mathbf{a}_{i_j}^\top \mathbf{x})}{p_{i_j}}}_{:= f^{(P,m)}(\mathbf{x})} . \tag{1.14}$$

Here $i_1, \cdots, i_m$ has been selected *i.i.d* from P. It can be easily verify that for any x, $\mathbb{E}[f^{(P,m)}(\mathbf{x})] = f(\mathbf{x})$. If the sampled function approximates $f(\mathbf{x})$ uniformly well, then the sampled function can serve as an effective proxy/surrogate for minimizing f. Trivially, one can set P to be the uniform distribution. However, if the contribution of few large $f_i(\mathbf{a}_i^T \mathbf{x})$ is the most in $f(\mathbf{x})$ then uniform subsampling will miss these important data points having large contributions and $f^{(P,m)}(\mathbf{x})$ will often underestimate $f(\mathbf{x})$. A possible solution to the drawbacks of uniform subsampling is to use importance sampling: sample the functions $f_i(\mathbf{a}_i^T \mathbf{x})$ that contribute most significantly to $f(\mathbf{x})$ with more probability and normalize the sum with the probability. Typically the relative the importance of each point, $\frac{f_i(\mathbf{a}_i^T \mathbf{x})}{\sum_{i=1}^{n} f_i(\mathbf{a}_i^T \mathbf{x})}$, will depend on the choice of of the data point $\mathbf{x}$. This motivates the definition of *sensitivity* [126].

Definition 1.4.0.1 (Sensitivity [126])**.** For $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, the *sensitivity* of point $\mathbf{a}_i$ with respect to a finite sum function f (Equation (1.3)) with domain $\mathcal{X} \subseteq \mathbb{R}^d$ is

$$\sigma_{f,\mathcal{X}}(\mathbf{a}_i) = \sup_{\mathbf{x} \in \mathcal{X}} \frac{f_i(\mathbf{a}_i^T \mathbf{x})}{\sum_{j=1}^{n} f_j(\mathbf{a}_j^T \mathbf{x})}.$$

The *total sensitivity* is defined as $\mathcal{G}_{f,\mathcal{X}} = \sum_{i=1}^{n} \sigma_{f,\mathcal{X}}(\mathbf{a}_i)$.

The following general approximation theorem is presented in [159] for sensitivity sampling with the notion of range space in place.

Theorem 1.4.0.5 (Theorem 9 [159])**.** *Consider the setting of finite sum problem as in Equation (1.13). For all $i \in [n]$, let $s_i \geq \sigma_{f,\mathcal{X}}(\mathbf{a}_i)$, $S = \sum_{i=1}^{n} s_i$, and $P = \left\{\frac{s_1}{S}, \ldots, \frac{s_n}{S}\right\}$. For*

some finite c and all $\varepsilon, \delta \in (0, 1/2)$*, if*

$$m \geq c \cdot \frac{S}{\varepsilon^2}\left(\Delta \log S + \log\left(\frac{1}{\delta}\right)\right),$$

then, with probability at least $1-\delta$*,*

$$(1-\varepsilon) f(\mathbf{x}) \leq f^{(P,m)}(\mathbf{x}) \leq (1+\varepsilon) f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{X}$$

Here, Δ *is an upper bound on the VC-dimension* $\Delta(\mathcal{R}_{\mathcal{F}})$ *where* $\mathcal{F}$ *is the set* $\left\{\frac{f_1(\mathbf{a}_1^T \mathbf{x})}{mn \cdot p_1}, \ldots, \frac{f_n(\mathbf{a}_n^T \mathbf{x})}{mn \cdot p_n}\right\}$.

In this way, one has to optimize modified objective in Equation (1.14) to get the approximately correct optimizer of $f(\mathbf{x})$ which essentially reduce the computational burden of optimizing the original objective.

1.4.1 Challenges in Stochastic Gradient Methods

We have discussed the backgrounds related to stochastic gradient methods we will now discussed the issues related to Stochastic Gradient Methods. Some of these issues, we will try to address in this thesis. Although, the distribution given in Equation (1.11) minimizes the variance of the t-th stochastic gradient, the full gradient of n-components must be computed in the process. In that sense, designing an efficient sampling algorithm for importance samples is an important research problem. In [271], authors have suggested static sampling schemes based on smoothness constant of each component and have provided the rate of convergence for importance sampled SGD. However, the optimal sampling scheme as in Equation (1.11) is dynamic. For the same reason, it is important to design a sampling scheme which is dynamic, improves upon existing static sampling scheme and is computationally cheaper to compute for every time steps.

SVRG is an iterative algorithm, where in each each iteration only stochastic gradients. In order to attain variance reduction a full gradient, full gradient is computed at a snapshot point in every few epochs. There are three issues with SVRG: i) the computation of the full gradient requires a full pass over the dataset. No progress (towards the optimal solution) is made during this time. On large scale problems, where one pass over the data might take several hours, this can yield to wasteful use of resources; ii) the theory requires the algorithm to restart at every snapshot point, resulting in discontinuous behaviour and iii) on strongly convex problems, the snapshot point can only be updated every $\Omega(\kappa)$ iterations (cf. [34, 98]), where $\kappa = L/\mu$ denotes the condition number (see (1.2.0.2)). When the condition number is large, this means that the algorithm relies for a long time on "outdated" deterministic information. In practice—as suggested in the original paper by Johnson and Zhang [98]—the update interval is often set to $O(n)$, without theoretical justification. *SAGA* on the other hand, circumvents the stalling phases by treating every iterate as a partial snapshot point. Hence, intuitively, in SAGA the gradient information at

partial snapshot point does have more recent information about the gradient as compared to SVRG. A big drawback of this method is the memory consumption: unless there are specific assumptions on the structure of the objective function. For large scale problems it is impossible to keep all data available in fast memory (i.e. cache or RAM) which means we can not run SAGA on large scale problems which do not have GLM structure. Also, extension of variance reduction algorithms to get nestrov's accelerated rate. However, the analysis of variance reduced accelerated methods differs vastly from traditional analysis of stochastic gradient methods. Hence, there is a need to obtain a unified but simpler framework to understand accelerated stochastic gradient methods.

Theorem G.1.2.2 is quite important and powerful: it can be utilized to achieve approximation algorithms which are based on sensitivity-sampling and have provable guarantees for a wide range of problems [65, 95, 146, 159]. However, two major issues of sensitivity sampling which have hindered more widespread use of sensitivity sampling are following: (i) Computation of the sensitivity score $\sigma_{f,\mathcal{X}}(\mathbf{a}_i)$ for all $i \in [n]$ is a hard task as it is not understood how to compute the supremum over all $\mathbf{x} \in \mathcal{X}$ in the expression of Definition 1.4.0.1. (ii) The sensitivity score considers the supremum of $\frac{f_i(\mathbf{a}_i^T \mathbf{x})}{\sum_{j=1}^n f_j(\mathbf{a}_j^T \mathbf{x})}$ over all $\mathbf{x} \in \mathcal{X}$. This also includes those $\mathbf{x}$ that may be very far from the true minimizer of f. Hence, it is usually a high value and a very worst case importance metric. Now since all the sensitivities are large, hence, the total sensitivity $\mathcal{G}_{f,\mathcal{X}}$ is large which essentially makes sample complexity too large to be useful in practice.

Coordinate descent and stochastic gradient descent, both the methods can be considered under the umbrella of stochastic gradient methods. However, coordinate descent methods converge faster than the stochastic gradient descent. Though, it is widely believed that coordinate descent on the dual objective has some connection with primal stochastic gradient update, however the connection is not explicitly well understood. One important research aspect is to understand this connection while training machine learning models especially overparametrized models as it has been studied that SGD converges faster in the overparametrized regime.

Chapter 2

Objectives

In the previous chapter, the general introduction of stochastic optimization methods and it's challenges were discussed. In this chapter, I would like to discuss the main objective of this thesis. In a broader picture, the main objective of the works presented in this thesis is to develop provably faster and improved optimization algorithms for optimizing machine learning objectives. More precisely, the main objective of this thesis is to address the research challenges in the following more broadly categorized areas of optimization research.

1. *Screening Rules:* For most data-analysis and machine learning task, one often has to work with optimization techniques in high dimensions. With the rise in advent of big data, one of the major challenges of the optimization method is to scale the optimization methods for very high dimensional data as the number of optimization variables/size of parameters grows beyond the capacity of current computing systems. The idea of screening the variables/parameters refers to remove optimization variables/parameters that for sure do not have any contribution to the optimal solution and hence can be safely eliminated from the problem. An improvement in the screening techniques can be hugely impactful in high dimensional optimization. In first part of this thesis, a new framework is proposed which enables screening on general convex optimization problems, using tools from convex duality such as frank wolfe gap and duality gap, instead of any geometric arguments.

2. *Sampling in Stochastic Optimization:* Stochastic gradient methods relies on random sampling of coordinates (in random coordinate descent) or data points (in stochastic gradient descent). The general convention is to use fixed sampling scheme such as uniform sampling or non-uniform sampling based on a fixed distribution. These fixed sampling schemes depends on the input data but are not adaptive in nature. That means the sampling distribution does not take into account of the current parameters or the local curvature of the optimization landscape. In contrast to these schemes, adaptive importance sampling schemes based on the current full gradient information constantly re-evaluate the probability of sampling each data point/coordinate and hence the relative importance of each data point during training is updated at every step. Thereby, gradient based adaptive sampling schemes often

surpass the performance of static algorithm. The major drawback of adaptive sampling strategies is that often it is computationally expensive to compute the optimal adaptive sampling distribution. In a part of this thesis, an efficient approximation of the gradient-based sampling is proposed which can efficiently be computed in each iteration and is provably better than uniform or any fixed importance sampling scheme.

Greedy coordinate descent can also be considered as a special case of adaptive gradient based sampling. An approximate greedy coordinate descent approach is also proposed which is provably better than uniform random sampling and is computationally more efficient than exact greedy coordinate descent.

3. *Variance Reduction in Stochastic Gradient Descent:* Stochastic gradient descent (SGD) (Robbins and Monro [205]) is frequently used to solve large scale optimization problems in machine learning due to its computation efficiency.However, one major drawback of SGD is that the rate of convergence of SGD algorithm to the optimal solution are often slow and far from the optimal on many problem classes. *Variance reduced methods* have been introduced to overcome this challenge. The variance reduced methods can roughly be divided in two classes, namely i) methods that achieve variance reduction by computing (non-stochastic) gradients of f from time to time at snapshot points, as for example done in SVRG, and ii) methods that maintain a table of previously computed stochastic gradients, such as done in SAGA. In a part of this thesis, we propose a variance reduction method which has shorter stalling phases of only order $\mathcal{O}(n/k)$ at the expense of only $\tilde{\mathcal{O}}(kd)$ additional memory where k can be chosen by the user. An accelerated version of variance reduced method is also proposed under a unifying analysis of accelerated stochastic gradient algorithms.

4. *Faster Stochastic Gradient Methods:* Stochastic gradient descent (SGD) is the method of choice to perform the optimization on large scale machine learning problems. However, the convergence of stochastic gradient descent is usually slow. Optimization error while optimizing with stochastic gradient descent consists of two terms (i) *bias term* : forgetting initialization point and (ii) *variance term*. In a part of this thesis, a new optimization algorithm would be introduced which allows the bias term to vanish quickly (accelerated rates).

Also, SGD is known to converge faster in the interpolation regime. Last part of this thesis would be devoted to obtain faster, better and improved stochastic gradient methods to perform optimization in the interpolation regime.

Chapter 3

Results and Contributions

3.1 Contributions Made in the Thesis

In Chapter 1, introduction and challenges for the problems considered in this thesis work were discussed. Later in Chapter 2, the main objective of this thesis was addressed. In this section, I specifically discuss the main contributions and results which resulted in this thesis. The main contributions made in this thesis are as follows:

- In this thesis, a new framework to derive screening rules for a large class of problems with a simple primal-dual structure is proposed. With the help of this framework, we are able to derive screening rules for a large set of machine learning problems for which no screening rules were known before. Furthermore, we were able to recover many existing screening rules as the side products of our screening framework. The proposed rules are dynamic in nature and are safe (only eliminates truly unimportant variables which does not contribute to the optimal solution) which allows it to be used with any existing algorithm. These screening rules are most suitable to use while using coordinate descent methods or frank-wolfe optimization methods for optimizing the objective function. **(Section 3.4)**

- An approximate steepest coordinate descent (ASCD), a new scheme of coordinate selection which combines ideas from the uniform coordinate descent (UCD) and from the steepest coordinate descent (SCD) strategies is proposed in this work. The existing convergence result for steepest coordinate descent for smooth and strongly convex functions is extended to the setting of smooth non-strongly convex functions. A novel lower bound which shows that the complexity estimates for steepest coordinate descent and uniform coordinate descent can be equal in the worst case is proved as well in this work. As the final algorithm, coordinate selection rules under ASCD for composite functions is proposed in this work and we prove that ASCD provable performs better than UCD and it can reach the performance of SCD in the best case scenario. **(Section 3.5)**

- An efficient adaptive sampling scheme which is the approximation of the gradient-based sampling is proposed in this work in the sense that it can be easily computed

in each iteration for little computational cost, (ii) is provably better than all fixed importance sampling including uniform sampling and (iii) behaves like the gradient-based sampling in special case if the gradient information is known accurately. Gradient-based sampling in CD methods are shown to outperform classical fixed sampling theoretically in this work. In the best case, the the algorithm can be faster up to a factor of the dimension n over the uniform sampling. As the main contribution, we propose an efficient adaptive importance sampling strategy which takes the approximate gradient information as the input to re-evaluate the sampling distribution. This sampling approach can be applied in CD as well as in SGD methods. **(Section 3.6)**

- An affine invariant convergence analysis for *Matching Pursuit* algorithms is presented in this work. The approach is tightly related to steepest coordinate descent update with non-orthogonal basis. The convergence analysis we propose in this work is also related to the analysis of coordinate descent and relies mostly on the properties of the atomic norm in order to generalize from orthogonal coordinates to non-orthogonal atoms. As a direct consequence of this analysis, a tighter rate of convergence for steepest coordinate descent is obtained. Extending the work to accelerated, we provide the first known accelerated MP algorithms as well as semi-steepest coordinate descent algorithm which provably converges to the optimal solution optimally. **(Section 3.7)**

- k-SVRG is variance reduced optimization algorithm which takes the good properties of the algorithms SAGA and SVRG and can be efficiently implemented in limited memory. The memory requirement of k-SVRG is of the order of $\tilde{\mathcal{O}}(kd)$ to store $\tilde{\mathcal{O}}(k)$ vectors. The algorithm is flexible in the choice of k and one can choose k depending on the available memory resource. We prove the convergence result for two variants of k-SVRG algorithm. This algorithm removes the barrier constraints of SAGA which requires to store n past gradient vectors. The propose algorithm also removes long stalling phase unlike SVRG. We also improved the analysis of SVRG in this work and show that SVRG algorithm converges for arbitrary sizes of inner loops for strongly convex and smooth functions.**(Section 3.8)**

- In this work, we propose a direct and simple template to analyze accelerated stochastic/deterministic convex optimization. We use the results from online learning literature and optimistic update for online learning to derive simple stochastic algorithms with fast rate of convergence. We derive accelerated results for smooth non-strongly convex as well strongly convex functions using the same template. A new *universal* algorithm (in the sense of 174) is also derived in this work for composite non-strongly-convex objectives using the same template. The new adaptive/universal algorithm simultaneously achieves the optimal rate of convergence for smooth and non-smooth f. We further extend our framework to get accelerated variance reduced algorithms with optimal rate of convergence. **(Section 3.9)**

- In Section 1.4 of Chapter 1, the issues with sensitivity sampling have been discussed which are (i) Computability and (ii) Pessimistic Bounds. In this work, a simple idea of *local sensitivity* is proposed to overcome the above barriers. Instead of sampling with the sensitivity over the full domain $\mathcal{X}$, the proposed algorithm considers the sensitivity over a small ball. Sampling by this local sensitivity gives a function which approximates the true function f well on the entire ball. We also relate the sensitivity scores of the second order approximation to a function with the leverage scores of a slightly modified matrix. These sensitivity scores of the local second order approximation are used to provide upper bound on the local sensitivity scores of the actual objective. **(Section 3.10)**

- In this part of the work, theoretical results for stochastic optimization methods in the interpolation regime for convex objectives are discussed. The main theoretical result of this work is to relate dual convergence guarantees with the primal convergence guarantees in terms of Bregman divergences of iterates via a generic equality. This allows us to use dual stochastic algorithms more specifically dual coordinate based algorithms for solving the primal objectives. The obtained vanilla dual coordinate ascent algorithm has the strong similarity with the (non-averaged) stochastic mirror descent on specific functions f_i's. The algorithm comes with the benefit of using an explicit regularizer and the algorithm converges to the minimum value of the regularizer which also interpolates the data. For accelerated coordinate ascent, a new algorithm is obtained which has the same order of computational complexity as that of SGD but has optimal rate of convergence in the interpolating regime. **(Section 3.11)**

3.2 List of Appended Papers (* denotes Joint First Authorship)

Here are the list of papers, manuscript and submitted papers which are appended in the Appendix of this thesis.

1. **Anant Raj**, Olbrich, J., Gärtner, B., Schölkopf, B., and Jaggi, M. (2016). Screening rules for convex problems. *Optimization for Machine Learning Workshop (OPT 2016), arXiv preprint arXiv:1609.07478*

2. Stich, S. U., **Anant Raj**, and Jaggi, M. (2017a). Approximate steepest coordinate descent. In *ICML 2017 - Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pages 3251–3259

3. Stich, S. U., **Anant Raj**, and Jaggi, M. (2017c). Safe adaptive importance sampling. In *Advances in Neural Information Processing Systems*, pages 4381–4391

4. Locatello*, F., **Anant Raj***, Praneeth Karimireddy, S., Rätsch, G., Schölkopf, B., Stich, S., and Jaggi, M. (2018). On matching pursuit and coordinate descent. In *35th International Conference on Machine Learning (ICML)*, pages 3204–3213. PMLR

5. **Anant Raj** and Stich, S. U. (2018). k-svrg: Variance reduction for large scale optimization. *arXiv preprint arXiv:1805.00982 (Manuscript)*

6. Joulani*, P., **Anant Raj***, György, A., and Szepesvari, C. (2020). A simpler approach to accelerated stochastic optimization: Iterative averaging meets optimism. In *ICML 2020- Proceedings of the 37th International Conference on Machine Learning*, PMLR

7. **Anant Raj**, Musco, C., and Mackey, L. (2020). Importance sampling via local sensitivity. In *International Conference on Artificial Intelligence and Statistics*, pages 3099–3109. PMLR

8. **Anant Raj** and Bach, F. (2020). Explicit regularization of stochastic gradient methods through duality. *arXiv preprint arXiv:2003.13807 (Submitted to AISTATS 2021)*

3.3 Delineation of Contribution to Collective Work

1. *Screening Rules for Convex Problems [239]*- Most of the theories in the paper were developed by Anant Raj and Jakob. Anant derived screening rules for constrained problem especially for L_1 constrained problem, elastic net constrained problems and simplex constrained problems. Jakob derived screening rules for box-constrained problems and found applications of derived screening rules in multiple real world problems. Illustrative experiment was set up by Anant. Most part of the paper was written by Anant and Martin Jaggi. Other authors contributed in the discussion and helped in writing the paper.

2. *Approximate Greedy Coordinate Descent [231]*- In this work, Anant derived the rate of convergence for steepest coordinate descent when the objective is smooth but non-strongly convex. Anant first provide the idea of approximate coordinate wise gradient oracle using Hessian which was later used by Anant and Sebastian to come up with the algorithm in the paper. Sebastian then proved the lower bound for steepest coordinate descent and went on to prove the sandwich theorem which states that the performance of our algorithm lies between uniformly random coordinate descent and greedy coordinate descent. Experiments were performed by Anant. The writing of the paper was done together by Anant, Sebastian and Martin. Martin was involved in all the discussions.

3. *Safe Adaptive Importance Sampling [233]*- In this work, Anant proved the convergence of coordinate descent and stochastic gradient descent with importance sampling. Later by joint effort of Sebastian and Anant, an optimization formulation was done to come up with the best possible sampling scheme at any instant which minimize the variance given the approximate gradient. Sebastian later proved lower bound and properties of the sampling distribution. Experiments were performed by Anant. The writing of the paper was done together by Anant, Sebastian and Martin. Martin was involved in the discussions.

4. *On Matching Pursuit and Coordinate Descent [144]*- Most of the theoretical results in this paper were jointly derived by Anant and Frenchesco while discussing on the white board. Later Praneeth joined the effort in proving the accelerated rate for greedy coordinate descent with the insights of decoupling the two updates of Nestrov's acceleration and use semi greedy sampling approach instead. Sebastian provided important insights with his simpler existing proof for accelerated coordinate descent. Toy experiment was performed by Franchesco in the paper. Most of the paper was written by Anant and Franchesco. Other co-authors helped in writing the paper and were involved in the discussing the ideas.

5. *k-SVRG : Variance Reduction for Large Scale Optimization [238]*- Theoretical results related to convex problems in this paper were developed by Anant and Sebatian jointly. Anant later proved convergence result for non-convex problems. Experiments were jointly performed by Anant and Sebastian. Both the authors were involved in writing the paper.

6. *A Simple Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism [104]*- Most of the theoretical results in this paper were jointly derived by Anant and Pooria while discussing on the white board. Pooria later derived results which came as a direct consequence of the result proved by Anant and Pooria together. Anant later extend the framework to derived accelerated rate of convergence for variance reduced methods. Andras and Csaba were involved in the discussion throughout the paper. They also contributed significantly in writing the paper.

7. *Importance Sampling via Local Sensitivity [240]*- Most of the theoretical results in this paper were jointly derived by Anant and Cameron. Cameron derived result about leverage score sampling. Anant derived the results related to optimization (approximate proximal point method) in the paper. In the early discussion, it was Lester who proposed to use local sensitivity instead. All the experiments were performed in by Anant and most of the part of the paper was written by Anant. Lester and Cameron also contributed in writing the paper.

8. *Explicit Regularization of Stochastic Gradient Methods through Duality [237]*- Many results in this paper were derived jointly by Francis and Anant. Experiments

were performed by Anant. Paper outline was written by Francis and then later Anant completed the paper writing as well.

3.4 Results in "Screening Rules for Convex Problems [239]"

3.4.1 Background

In this work, we consider the optimization problem of the form given in Equation (A) which can also be written in the dual form as given in Equation (B). This primal-dual relationships are very useful in theory and practice as it can be used for the computation of duality gap. Duality gap acts as a certificate for the approximation quality of the optimal solution. A vast range of machine learning optimization problems can be formulated as (A) and (B), which are bind with the primal-dual relationship as discussed above (are dual to each other):

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \left[\; \mathcal{O}_A(\boldsymbol{x}) := \; f(A\boldsymbol{x}) \; + \; g(\boldsymbol{x}) \qquad \right] \tag{A}$$

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \left[\; \mathcal{O}_B(\boldsymbol{w}) := \; f^*(\boldsymbol{w}) \; + \; g^*(-A^\top \boldsymbol{w}) \right] \tag{B}$$

The matrix $A \in \mathbb{R}^{d\times n}$ is known as data matrix, and the functions $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are arbitrary closed convex functions. The functions f^*, g^* in (B) represents the convex conjugate functions of their corresponding counterparts f, g in (A).

The duality gap is often considers as a strong optimality certificate as for the case of convex functions, the duality gap a primal iterate and at its corresponding dual iterate can only be zero iff the iterates are optimal iterates. Duality gap certificates plays an important role in deriving the screening rules for convex problems. This basically allows us to screen at the optimal point. The duality gap is defined as $G(\boldsymbol{w},\boldsymbol{x}) := \mathcal{O}_A(\boldsymbol{x}) + \mathcal{O}_B(\boldsymbol{w})$ for any pair of primal and its corresponding dual variable.

Since, the duality gap is seen a certificate of approximation quality, hence, the true optimal values $\mathcal{O}_A(\boldsymbol{x}^*)$ and $-\mathcal{O}_B(\boldsymbol{w}^*)$ always lie within the duality gap.

The Gap Function. Consider the case of differentiable function f. In this case, a simpler duality gap can be studied

$$G(\boldsymbol{x}) := \mathcal{O}_A(\boldsymbol{x}) + \mathcal{O}_B(\boldsymbol{w}(\boldsymbol{x})) \tag{3.1}$$

which is purely defined as a function of $\boldsymbol{x}$, using the first order optimality relation $\boldsymbol{w}(\boldsymbol{x}) := \nabla f(A\boldsymbol{x})$.

The Wolfe-Gap Function. General duality gap in the case of constrained optimization problem defined over a bounded set $\mathcal{C}$ and $\boldsymbol{x} \in \mathcal{C}$ reduces to a specific kind of gap function which is called the "Wolfe-Gap function". The Wolfe gap function is defined as follows,

$$GW(\boldsymbol{x}) := \max_{\boldsymbol{y} \in \mathcal{C}} (A\boldsymbol{x} - A\boldsymbol{y})^\top \nabla f(A\boldsymbol{x}). \tag{3.2}$$

One can easily see that for g being the indicator function of the constraint set $\mathcal{C}$, and $\boldsymbol{w}(\boldsymbol{x}) := \nabla f(A\boldsymbol{x})$, Wolfe gap function becomes the general duality gap.

3.4.2 Main Results

Simplex Constrained Problems: Optimization problems over unit simplex $\triangle := \{\boldsymbol{x} \in \mathbb{R}^n \mid x_i \geq 0, \quad \sum_{i=1}^n x_i = 1\}$are very important constrained optimization problems. This class of problems includes all the optimization problems over any finite polytope. The vertices are represented by the columns of A in this case, and $\boldsymbol{x}$ represents barycentric coordinates corresponding to the point $A\boldsymbol{x}$. Formally, $g(\boldsymbol{x})$ can be seen as the indicator function of the unit simplex $\mathcal{C} = \triangle$ in this case. In the following Theorem 3.4.2.1, we provide our first main result of this work for screening on simplex constrained optimization problems for any arbitrary iterate $\boldsymbol{x}$ without knowing $\boldsymbol{x}^\star$. Function f are assumed to be smooth and strongly convex so that the distance between the arbitrary iterate and the optimal iterate can be related with the duality gap. Only main theorem statements will be mentioned here and details of the proof and other results are provided in the Appendix.

Theorem 3.4.2.1 (**Anant Raj** *et al.* [239]). *Let us consider f to be an L-smooth and μ-strongly convex function over the unit simplex $\mathcal{C} = \triangle$. Then for simplex constrained optimization problem $\min_{\boldsymbol{x} \in \triangle} f(A\boldsymbol{x})$, the screening rule can be stated as following for any $i \in [n]$*

$$(\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) > L\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} \|\boldsymbol{a}_i - A\boldsymbol{x}\| \;\Rightarrow\; x_i^\star = 0\,.$$

The screening rules for simplex constrained problems as in Theorem 3.4.2.1 is very general and have a lot of practical implications. For example, one can easily verify that new screening rules for squared loss SVM can be derived by a simple application of result presented in Theorem 3.4.2.1.

Corollary 3.4.2.2 (Square Hinge-Loss SVM, **Anant Raj** *et al.* [239]). *For the squared hinge loss SVM, the screening rule can be stated as follows,*

$$(\boldsymbol{a}_i - A\boldsymbol{x})^\top A\boldsymbol{x} > \sqrt{\max_i (A\boldsymbol{x} - \boldsymbol{a}_i)^\top A\boldsymbol{x}}\, \|\boldsymbol{a}_i - A\boldsymbol{x}\| \Rightarrow\; x_i^\star = 0. \tag{3.3}$$

L_1-Constrained Problems: L_1-constrained optimization problems are again very popular in machine learning applications. This kind of optimization formulation are used

in order to induce sparsity in the optimal solution. Here below, we state the results for screening on general L_1-constrained optimization problems. The L_1-constrained optimization can be written as $\min_{\boldsymbol{x}\in\mathcal{C}} f(A\boldsymbol{x})$ for $\mathcal{C} = L_1 \subset \mathbb{R}^n$ (or a scaled version of the L_1-ball). We provide screening rules only i terms of current iterate and does not involve the optimal iterate. For a smooth and strongly function f, one can obtain the following screening rule for L_1-constrained optimization problem.

Theorem 3.4.2.3 (**Anant Raj** *et al.* [239]). *Let us consider f to be an L-smooth and μ-strongly convex function over the L_1-ball. Then for L_1-constrained optimization problem $\min_{\boldsymbol{x}\in L_1} f(A\boldsymbol{x})$, the screening rule can be stated as follows for any $i \in [n]$*

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) + L(\|\boldsymbol{a}_i\|_2 + \|A\boldsymbol{x}\|_2)\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} < 0 \Rightarrow \boldsymbol{x}_i^\star = 0 \tag{3.4}$$

Going further in sparse optimization problem, elastic net constrained/regularization is also very often used as an alternative to L_1 constrained/regularization. At times, elastic net based solution outperforms the Lasso and still has sparse representation [276]. The expression for the elastic net is the following:

$$\alpha\|\boldsymbol{x}\|_1 + (1-\alpha)\tfrac{1}{2}\|\boldsymbol{x}\|_2^2 \,.$$

Here below we discuss the screening rule obtained for elastic net constrained optimization problem using our framework. Similar to the previous optimization formulation, elastic net constrained optimization formulation can be written as $\min_{\boldsymbol{x}\in\mathcal{C}} f(A\boldsymbol{x})$ for $\mathcal{C}$ being the elastic net constraint, or a scaled version of it. We use only the current iterate $\boldsymbol{x}$ and not optimal point in order to derive screening rule for elastic net constrained problem. Below we provide the screening rule for elastic net constrained problem given the the function f is smooth and strongly convex function.

Theorem 3.4.2.4 (**Anant Raj** *et al.* [239]). *Let us consider f to be an L-smooth and μ-strongly convex function over the elastic net norm ball. Then for elastic net constrained optimization $\min_{\boldsymbol{x}\in L_E} f(A\boldsymbol{x})$, the screening rule can be stated as follows for any $i \in [n]$*

$$\begin{aligned}\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x})\left[\frac{2\alpha}{3-\alpha}\right] \\ + L(\|\boldsymbol{a}_i\|_2 + \|A\boldsymbol{x}\|_2\left[\frac{2\alpha}{3-\alpha}\right])\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} < 0 \Rightarrow \boldsymbol{x}_i^\star = 0\end{aligned} \tag{3.5}$$

It can be easily verified that the results given in Theorem 3.4.2.4 recovers the result for L_1 constrained case as a special case, when $\alpha \to 1$.

Box Constrained Problems: Box-constrained problems are another widely used problem in several machine learning applications, including SVMs. It can very well be

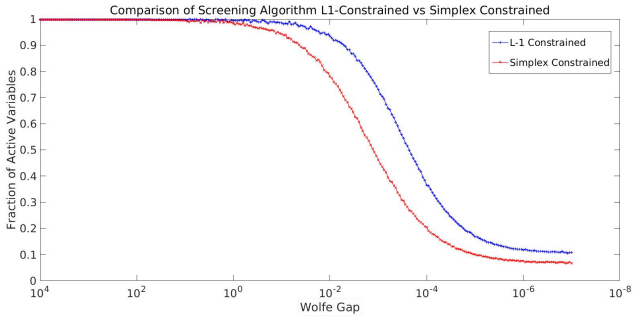


Figure 3.1: Simplex- vs L_1-constrained Screening

screened as well using the framework discussed in this work. One can always assume the constraint set $\mathcal{C} = \square := \{\boldsymbol{x} \in \mathbb{R}^n \mid 0 \leq x_i \leq 1\}$ without the loss of generality. Our framework provides screening rules to predict both if a variable will take the upper or lower constraint.

Theorem 3.4.2.5 (**Anant Raj** *et al.* [239]). *Let us consider f to be an L-smooth and μ-strongly convex function. Then for box-constrained optimization* $\min_{\boldsymbol{x}\in\square} f(A\boldsymbol{x})$*, the screening rule can be stated as follows for any $i \in [n]$*

$$\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} > 0 \;\Rightarrow\; \mathbf{x}_i^\star = 0\,, \;\; \textit{and}$$
$$\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) + \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} < 0 \;\Rightarrow\; \mathbf{x}_i^\star = 1\,.$$

Hinge loss SVM can be seen as one of many special cased of box-constrained optimizaion problem. All the results provided above have been derived using the same framework developed in the paper. The framework works in two stages. In first stage, we obtain the relation between optimal primal and dual variable using the optimality conditions ([239, Lemma 5]). In the later stage, we bound the distance between any (feasible) current dual iterate and the optimal dual solution $\mathbf{w}^\star$. This framework is much more general and can be applied to a very wide range of problems as compared to the earlier proposed problem specific geometric frameworks. Using similar tools and techniques, dynamic screening rules can be derived for penalized problems as well which has also been discussed in previous works.

Experiments: The main contribution of this work is on the theoretical generality to derive new set of screneing rules for a wide range of constrained/non-constrained optimization problem. We also evaluate the performance of simplex constrained and L1-constrained problems on a toy example. The fraction of active variables and the Wolfe-Gap function as optimization algorithm progress are plotted for both the case. We provide the details of the experimental setting and data generation process in the the main paper (Appendix). In Fig 3.1, the blue curve represents the screening performance for

the L_1-constrained screening case, while the red curve represents the performance for simplex constrained screening. The theorems 3.4.2.3 and 3.4.2.1 are well in line with the phenomena in Fig 3.1. It can be seen in the plot that screening rule is not effective at the start of the optimization as the duality gap is large. However, as the algorithm progress, the duality gap becomes considerably smaller, and our screening rules start to screen out the varibles. For both variants, screening becomes slow towards the end because of the fact that gradient also becomes smaller.

3.5 Results in "Approximate Greedy Coordinate Descent [231]"

3.5.1 Background

In this work, we consider the composite convex functions $F : \mathbb{R}^n \to \mathbb{R}$ of the form

$$F(\boldsymbol{x}) := f(\boldsymbol{x}) + \Psi(\boldsymbol{x}) \tag{3.6}$$

where f is coordinate-wise L-smooth and Ψ is convex and separable, that is that is $\Psi(\boldsymbol{x}) = \sum_{i=1}^{n} \Psi_i([\boldsymbol{x}]_i)$. For simplicity, $\Psi \equiv 0$ is assumed here however, . Coordinate descent methods have the following update rule given constant step size:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \tfrac{1}{L} \nabla_{i_t} f(\boldsymbol{x}) \boldsymbol{e}_{i_t} . \tag{3.7}$$

In *Uniform Coordinate Descent* (UCD) the active coordinate i_t is selected uniformly at random from the set $[n]$, $i_t \in_{u.a.r.} [n]$. *Steepest Coordinate Descent* (SCD) selects the coordinate following the Gauss-Southwell (GS) rule which is a greedy selection procedure:

$$i_t = \underset{i \in [n]}{\arg\max} \nabla_i \left| f(\boldsymbol{x}_t) \right| . \tag{3.8}$$

Convergence of Steepest Coordinate Descent

Earlier, the notion of coordinate wise smoothness has been discussed. It implies

$$f(\boldsymbol{x} + \eta \boldsymbol{e}_i) \leq f(\boldsymbol{x}) + \eta \nabla_i f(\boldsymbol{x}) + \tfrac{L_i}{2} \eta^2 \tag{3.9}$$

With the quadratic upper bound (3.9) and the coordinate descent update, one can easily verify that we have the following lower bound on the one step progress

$$\mathbb{E}\left[f(x_t) - f(x_{t+1}) \mid x_t \right] \geq \mathbb{E}_{i_t} \left[\tfrac{1}{2L} \left| \nabla_{i_t} f(\boldsymbol{x}_t) \right|^2 \right] . \tag{3.10}$$

Let us use $\tau_{\text{UCD}}(\boldsymbol{x}_t)$ and $\tau_{\text{SCD}}(\boldsymbol{x}_t)$ to denote the right hand side expressions For UCD and SCD. The expression on the right hand for UCD and SCD evaluates to

$$\begin{aligned} \tau_{\text{UCD}}(\boldsymbol{x}_t) &:= \tfrac{1}{2nL}\|\nabla f(\boldsymbol{x}_t)\|_2^2 \\ \tau_{\text{SCD}}(\boldsymbol{x}_t) &:= \tfrac{1}{2L}\|\nabla f(\boldsymbol{x}_t)\|_\infty^2 \end{aligned} \tag{3.11}$$

By aoplying Cauchy-Schwarz inequality, one can find that

$$\tfrac{1}{n}\tau_{\text{SCD}}(\boldsymbol{x}_t) \leq \tau_{\text{UCD}}(\boldsymbol{x}_t) \leq \tau_{\text{SCD}}(\boldsymbol{x}_t)\,. \tag{3.12}$$

From the above expression, it is clear that one step progress of SCD is always at least as good as that of one step UCD on average. Moreover, in the best case the gain of SCD over UCD per iteration can be as large as by a factor of n. However, due to the technical complexity, one can not prove this linear speed for more than one iteration as it is almost impossible to track down the gain in expressions (3.12) which depend on the sequence of iterates.

Let us consider the case of smooth but non-strongly convex function f. In this case, the analysis of greedy coordinate descent from [180] does not apply as it only works for smooth and strongly convex functions. In this work, we extend the analysis from [180] to smooth non-strongly convex functions.

Theorem 3.5.1.1 (Stich *et al.* [232])**.** *Let us assume $f\colon \mathbb{R}^n \to \mathbb{R}$ to be a convex and coordinate-wise L-smooth function. Then for the sequence of iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD, the following convergence bound holds:*

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \frac{2LR_1^2}{t}\,, \tag{3.13}$$

for $R_1 := \max\limits_{\boldsymbol{x}^\star \in X^\star} \left\{ \max\limits_{\boldsymbol{x}\in\mathbb{R}^n} [\|\boldsymbol{x}-\boldsymbol{x}^\star\|_1 \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)] \right\}$.

Note that the R_1 is the diameter of the level set at $f(\boldsymbol{x}_0)$ measured in the 1-norm. For the case of UCS, R_1^2 in (3.13) is replaced by nR_2^2, where R_2 is the diameter of the level at $f(\boldsymbol{x}_0)$ measured in the 2-norm (cf. Nesterov [170], Wright [262]). By simple application of cauchy shwartz inequality, one can verify that

$$\tfrac{1}{n}R_1^2 \leq R_2^2 \leq R_1^2\,, \tag{3.14}$$

Hence, the upper bound of the optimization error SCD can be tighter than that of *UCD* and for the same reason we might loosely claim that SCD can have faster convergence up to a factor of n over to UCD.

3.5.2 Main Results

Lower Bound: So far we have discussed the best case scenario when steepest coordinate descent can improve upon the uniform coordinate descent. Now, we show that in the worst case scenario the convergence of SCD method is same as that of UCD. In Theorem 3.5.2.1 below, we consider a function $q\colon \mathbb{R}^n \to \mathbb{R}$, for which the one step progress of SCD and UCD update is approximately same *i.e.* $\tau_{\text{SCD}}(\boldsymbol{x}_t) \approx \tau_{\text{UCD}}(\boldsymbol{x}_t)$ up to a constant factor, for all iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD.

By the same reasoning, it is also possible to construct a family of function where the gain is maximum. For instance, let us consider a functions which is separable and has a low dimensional structure. If we fix the integers s, n such that $\frac{n}{s} \approx \lambda$, then one can define the function $f\colon \mathbb{R}^n \to \mathbb{R}$ as

$$f(\boldsymbol{x}) := q(\pi_s(\boldsymbol{x})) \tag{3.15}$$

where π_s denotes the projection to $\mathbb{R}^s$ (being the first s out of n coordinates) and $q\colon \mathbb{R}^s \to \mathbb{R}$ is the function from Theorem 3.5.2.1. In that case, it is very clear that

$$\tau_{\text{SCD}}(\boldsymbol{x}_t) \approx \lambda \cdot \tau_{\text{UCD}}(\boldsymbol{x}_t)\,, \tag{3.16}$$

for all iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by steepest coordinate descent. Here below we state the lower bound result.

Theorem 3.5.2.1 (Stich *et al.* [232])**.** *Consider the function $q(\boldsymbol{x}) = \frac{1}{2}\langle Q\boldsymbol{x}, \boldsymbol{x}\rangle$ for $Q := I_n - \frac{99}{100n}J_n$, where $J_n = 1_n 1_n^T$, $n > 2$. Then there exists $\boldsymbol{x}_0 \in R^n$ such that for the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD it holds*

$$\|\nabla q(\boldsymbol{x}_t)\|_\infty^2 \leq \tfrac{4}{n}\|\nabla q(\boldsymbol{x}_t)\|_2^2\,. \tag{3.17}$$

Algorithm and Complexity: It is very obvious that exact GS selection rule is a very expensive task to compute and has the same computational complexity as that of gradient descent. However, efficiency of coordinate descent comes from the fact that one needs to compute the direction wise gradient only and hence each update is n-times cheaper than that of gradient descent update (cf. Nesterov [170]). The same argument also holds for SCD by Theorem 3.5.2.1. A class of functions which has this property and also is widely used in machine learning domain can be represented by functions $F\colon \mathbb{R}^n \to \mathbb{R}$

$$F(\boldsymbol{x}) := f(A\boldsymbol{x}) + \sum_{i=1}^{n} \Psi_i([\boldsymbol{x}]_i) \tag{3.18}$$

where A is a $d \times n$ data matrix, and $f\colon \mathbb{R}^d \to \mathbb{R}$, and $\Psi_i\colon \mathbb{R}^n \to \mathbb{R}$ are convex and simple functions which have linear time complexity of computing the gradient. This class of functions includes most popular loss functions widely used in machine learning

applications such as least squares, logistic regression, Lasso, and SVMs (when solved in dual form). The main idea behind the algorithm proposed in this work is following: we would like to track the evolution of the gradients along all the coordinate even though we only compute the coordinate wise gradient. Once, we have this evolution with us for all the coordinates, we can construct an active set which contains the coordinates that have provably larger coordinate wise gradient and also contains the steepest coordinate. Now in the coordinate selection step, one can choose the coordinate from the active set of coordinates which provably has coordinates with large coordinate wise gradient. Selecting coordinates from this active set uniformally at random will provably have faster rate of convergence than UCD. This algorithm we refer as ASCD (approximately steepest coordinate descent) in this work.

Convergence Rate Guarantee. The main result of this work is to prove that the performance of ASCD is at least as good as that of UCD on avrage and can reach upto SCD in the best case. As explained previously, one can think of this as the coordinates with smaller value of coordinate wise gradient are left out of this active set and hence on average per iterate gain in ASCD is more than as that of UCD. When, we know the exact full gradient, then the active set contains only one coordinate which has the maximum value of coordinate wise gradient in absolute. Here below, we state the sandwich theorem of our work.

Theorem 3.5.2.2 (Stich *et al.* [232])**.** *Consider $f : \mathbb{R}^n \to \mathbb{R}$ to be a convex and coordinate-wise L-smooth function, let τ_{UCD}, τ_{SCD}, τ_{ASCD} denote the expected one step progress* (3.11) *of UCD, SCD and ASCD, respectively, and suppose all methods use the same step-size rule. Then*

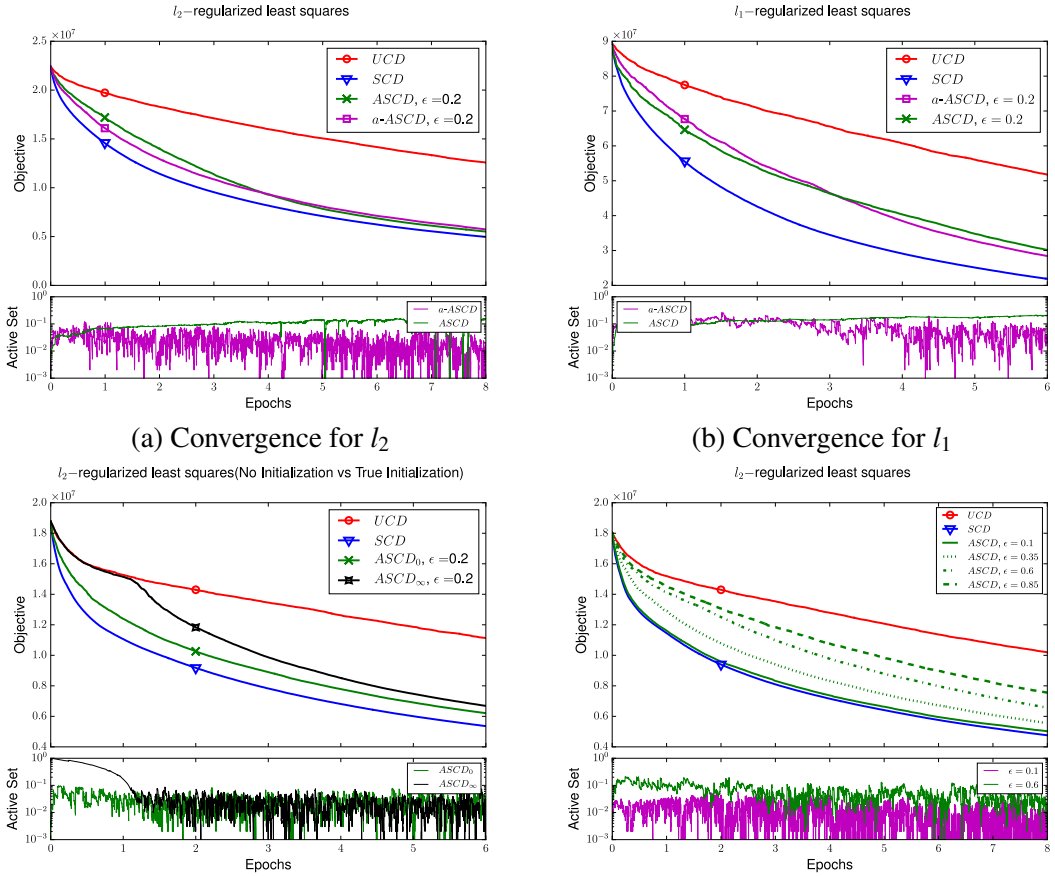
$$\tau_{\text{UCD}}(\boldsymbol{x}) \leq \tau_{\text{ASCD}}(\boldsymbol{x}) \leq \tau_{\text{SCD}}(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathbb{R}^n . \tag{3.19}$$

Experiments: Details about the heuristic variants of approximate steepest coordinate descent (a-ASCD), data generation and experimental set-up are provided in the Appendix (main paper). Here, the observations from the empirical evaluation are discussed.

Here are the highlights of the experimental study:

1. Irrespective of the initialization of the gradient vector, the algorithm performs well in the task on learning the active set and eventually in optimizing the function.

2. Even with very crude gradient approximation obtained by the gradient oracle, the algorithm performs very well and the convergence is excellent. The size of the active set is also very small.



(a) Convergence for l_2

(b) Convergence for l_1

(c) True vs No Initialization for l_2

(d) Error Variation (ASCD)

Figure 3.2: Experimental results on synthetically generated datasets

3.6 Results in "Safe Adaptive Importance Sampling [233]"

3.6.1 Background

Adaptive Sampling for Coordinate Descent: Let us again consider the optimization problem $\min_{\boldsymbol{x}} f(\boldsymbol{x})$. We assume that the objective function $f\colon \mathbb{R}^n \to \mathbb{R}$ is a convex function with coordinate-wise L_i-Lipschitz continuous gradients. We have the following update for coordinate descent,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma_k \nabla_{i_k} f(\boldsymbol{x}_k) \boldsymbol{e}_{i_k}\,. \tag{3.20}$$

The coordinate i_k can either be selected in deterministic fashion (cyclic descent, steepest descent) or in random picking fashion where i_k is randomly picked according $\boldsymbol{p}_k \in \Delta^n$ which assigns the sampling probabilities for each coordinates. Generally, the step size is

chosen as $\gamma_k = L_{i_k}^{-1}$ which also minimizes the quadratic upper bound. However, we set $\gamma_k = \alpha_k [\boldsymbol{p}_k]_{i_k}^{-1}$ where α_k is not dependent on the direction i_k in this work. This results in directionally-unbiased updates as in the case of importance sampling SGD update. It holds

$$\mathbb{E}_{i_k\sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k\right] \overset{(C.1)}{\leq} \mathbb{E}_{i_k\sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_k) - \frac{\alpha_k}{[\boldsymbol{p}_k]_{i_k}}\left(\nabla_{i_k} f(\boldsymbol{x}_k)\right)^2 + \frac{L_{i_k}\alpha_k^2}{2[\boldsymbol{p}_k]_{i_k}^2}\left(\nabla_{i_k} f(\boldsymbol{x}_k)\right)^2 \mid \boldsymbol{x}_k\right]$$
$$= f(\boldsymbol{x}_k) - \alpha_k \|\nabla f(\boldsymbol{x}_k)\|_2^2 + \sum_{i=1}^{n} \frac{L_i \alpha_k^2}{2[\boldsymbol{p}_k]_i}\left(\nabla_i f(\boldsymbol{x}_k)\right)^2 . \tag{3.21}$$

While applying adaptive sampling strategies, one can select both variables α_k and $\boldsymbol{p}_k$ as one wishes to. Hence, to get the maximum per iterate average progress, both the variables α_k and $\boldsymbol{p}_k$ are selected in such a way that they give *minimum* value to the upper bound (3.21) . The optimal choice of the probability vector $\boldsymbol{p}_k$ in (3.21) does not depend on α_k, however, the optimal α_k parameter depends on the probability vector $\boldsymbol{p}_k$. The following observation is trivial to observe but is very important.

Lemma 3.6.1.1 (Stich *et al.* [233]). *Assum that $\alpha_k = \alpha_k(\boldsymbol{p}_k)$ minimizes the upper bound of (3.21), then $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \frac{\alpha_k}{[\boldsymbol{p}_k]_{i_k}} \nabla_{i_k} f(\boldsymbol{x}_k) \boldsymbol{e}_{i_k}$ satisfies*

$$\mathbb{E}_{i_k\sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k\right] \leq f(\boldsymbol{x}_k) - \frac{\alpha_k(\boldsymbol{p}_k)}{2}\|\nabla f(\boldsymbol{x}_k)\|_2^2 . \tag{3.22}$$

Example 3.6.1.1 (Optimal sampling [233]). *For probabilities $[\boldsymbol{p}_k^\star]_i = \frac{\sqrt{L_i}|\nabla_i f(\boldsymbol{x}_k)|}{\|\sqrt{\mathbf{L}}\nabla f(\boldsymbol{x}_k)\|_1}$ and $\alpha_k(\boldsymbol{p}_k^\star) = \frac{\|\nabla f(\boldsymbol{x}_k)\|_2^2}{\|\sqrt{\mathbf{L}}\nabla f(\boldsymbol{x}_k)\|_1^2}$, the upper bound in Equation (3.21) attends its minimum. This immediately gives rise to the following observation, $\frac{1}{\mathrm{Tr}[\mathbf{L}]} \leq \alpha_k(\boldsymbol{p}_k^\star) \leq \frac{1}{L_{\min}}$, where $L_{\min} := \min_{i\in[n]} L_i$.*

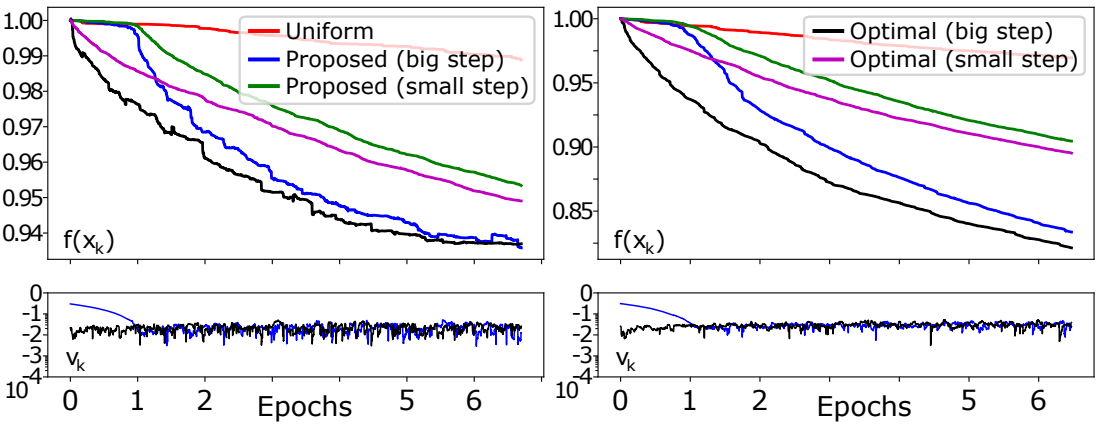
The ideal adaptive algorithm. The stepsize and the sampling distribution for CD under gradient based adaptive sampling scheme are selected as in Example 3.6.1.1. This also gives us new bound on the expected one-step progress of the CD under gradient based adaptive sampling scheme which will be later used to derive rate of convergence of this algorithm following the standard techniques in optimization literature.

SGD with Adaptive Sampling: SGD methods are widely used in practice to optimize finite sum functions which decompose as a sum

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) \tag{3.23}$$

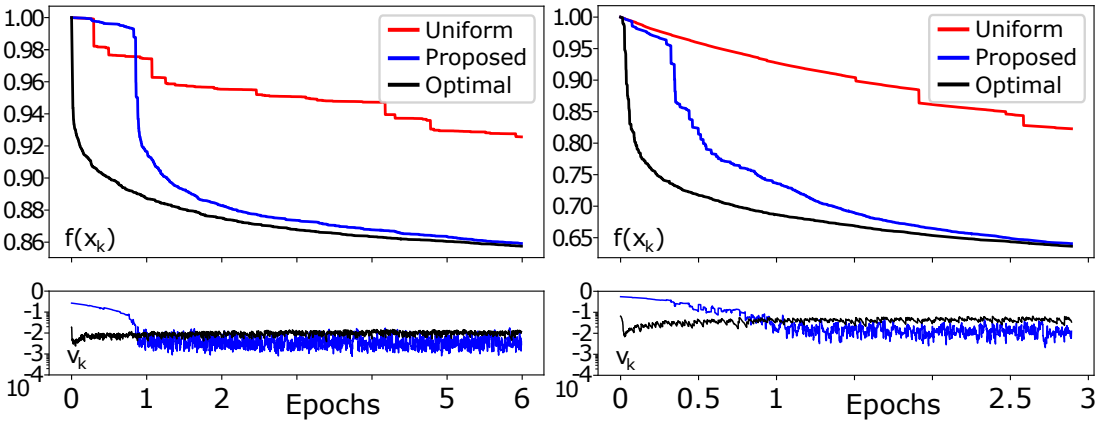
with each $f_i\colon \mathbb{R}^d \to \mathbb{R}$ be a convex function. Previously [185, 272, 273], the authors argued that the following gradient-based sampling $[\tilde{\boldsymbol{p}}_k^\star]_i = \frac{\|\nabla f_i(\boldsymbol{x}_k)\|_2}{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}$ maximizes the the expected progress (3.21) by minimizing the variance of the stochastic gradient estimate.

3.6.2 Main Results



(a) *rcv1'*, $L1$ reg. (b) *rcv1'*, $L2$ reg.

Figure 3.3: (CD, square loss) Fixed vs. adaptive sampling strategies.



(a) *rcv1'*, $L1$ reg. (b) *real-sim'*, $L2$ reg.

Figure 3.4: (CD, squared hinge loss) Function value vs. number of iterations for optimal stepsize.

An Optimization Formulation of Sampling: We now discuss the main result of this work. Before moving into the result, we first discribe one crucial assumption made in this paper. We assume that in each iteration we have the access to two vectors $\boldsymbol{\ell}_k, \boldsymbol{u}_k \in \mathbb{R}^n_{\geq 0}$ the contains the safe upper and lower bounds on the component wise gradient norms

($[\boldsymbol{\ell}_k]_i \leq \|\nabla f_i(\boldsymbol{x}_k)\|_2 \leq [\boldsymbol{u}_k]_i$). for stochastic gradient descent or on the absolute value of coordinate wise gradient($[\boldsymbol{\ell}_k]_i \leq |\nabla_i f(\boldsymbol{x}_k)| \leq [\boldsymbol{u}_k]_i$) for coordinate descent algorithms.

One can minimize the upper bound (3.21) to get the following problem as in Stich *et al.* [233],

$$\min_{\alpha_k} \min_{\boldsymbol{p}_k \in \Delta^n} \left[-\alpha_k \|\boldsymbol{c}_k\|_2^2 + \frac{\alpha_k^2}{2} V(\boldsymbol{p}_k, \boldsymbol{c}_k) \right] \quad \Leftrightarrow \quad \min_{\boldsymbol{p}_k \in \Delta^n} \frac{V(\boldsymbol{p}_k, \boldsymbol{c}_k)}{\|\boldsymbol{c}_k\|_2^2} \tag{3.24}$$

where $\boldsymbol{c}_k \in \mathbb{R}^n$ is the *unknown* true gradient. That means, it is possible to write $\boldsymbol{c}_k \in C_k := \{\boldsymbol{x} \in \mathbb{R}^n : [\boldsymbol{\ell}_k]_i \leq [\boldsymbol{x}]_i \leq [\boldsymbol{u}_k]_i, i \in [n]\}$ with respect to the bounds $\boldsymbol{\ell}_k, \boldsymbol{u}_k$. Hence, we solve the following min-max problem to obtain the best sampling strategy with respect to C_k.

$$v_k := \min_{\boldsymbol{p} \in \Delta^n} \max_{\boldsymbol{c} \in C_k} \frac{V(\boldsymbol{p}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2}, \qquad \text{and to set} \qquad (\alpha_k, \boldsymbol{p}_k) := \left(\tfrac{1}{v_k}, \hat{\boldsymbol{p}}_k\right), \tag{3.25}$$

where $\hat{\boldsymbol{p}}_k$ denotes a solution of (3.25).

Theorem 3.6.2.1 (Stich *et al.* [233]). *Let $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{c}}) \in \Delta^n \times \mathbb{R}^n_{\geq 0}$ be a solution of the optimization problem in (3.25). Then $L_{\min} \leq v_k \leq \operatorname{Tr}[\mathbf{L}]$ and*

1. $\max_{\boldsymbol{c} \in C_k} \frac{V(\hat{\boldsymbol{p}}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \leq \max_{\boldsymbol{c} \in C_k} \frac{V(\boldsymbol{p}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2}, \forall \boldsymbol{p} \in \Delta^n$; *(in the worst case)*
2. $V(\hat{\boldsymbol{p}}, \boldsymbol{c}) \leq \operatorname{Tr}[\mathbf{L}] \cdot \|\boldsymbol{c}\|_2^2, \forall \boldsymbol{c} \in C_k$. *(in the worst case)*

Remark 3.6.2.1 (Stich *et al.* [233]). *In the special case when all the coordinate wise lipschitz constants are same i.e. $L_i = L$ for all $i \in [n]$, then L_i-based sampling reduces to uniform sampling strategy . From the above result it is clear that $\hat{\boldsymbol{p}}$ is provably better than uniform sampling even in the worst case: $V(\hat{\boldsymbol{p}}, \boldsymbol{c}) \leq Ln\|\boldsymbol{c}\|_2^2, \forall \boldsymbol{c} \in C_k$.*

Optimization problem (3.25) can also be denotes as $\sqrt{v_k} = \max_{\boldsymbol{c} \in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}{\|\boldsymbol{c}\|_2} = \max_{\boldsymbol{c} \in C_k} \frac{\langle \sqrt{\boldsymbol{l}}, \boldsymbol{c} \rangle}{\|\boldsymbol{c}\|_2}$, where $[\boldsymbol{l}]_i = L_i$ for $i \in [n]$. Hence, the maximum of the optimization objective is obtained when vectors $\boldsymbol{c} \in C_k$ minimize the angle with the vector $\boldsymbol{l}$.

Theorem 3.6.2.2 (Stich *et al.* [233]). *Let $\boldsymbol{c} \in C_k$, $\boldsymbol{p} = \frac{\sqrt{\mathbf{L}}\boldsymbol{c}}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}$ and denote $m = \|\boldsymbol{c}\|_2^2 \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^{-1}$. If*

$$[\boldsymbol{c}]_i = \begin{cases} [\boldsymbol{u}_k]_i & \text{if } [\boldsymbol{u}_k]_i \leq \sqrt{L_i} m, \\ [\boldsymbol{\ell}_k]_i & \text{if } [\boldsymbol{\ell}_k]_i \geq \sqrt{L_i} m, \\ \sqrt{L_i} m & \text{otherwise}, \end{cases} \qquad \forall i \in [n], \tag{3.26}$$

then $(\boldsymbol{p}, \boldsymbol{c})$ is a solution to (3.25). This solution is computable in $O(n \log n)$ time.

Experiments: Details about the experimental set-up and data generation are given in the main paper. The main findings of the experimental study can be summarized as follows:

- From the empirical evaluations it is clear that the proposed sampling converges almost as good as the optimal gradient-based sampling in terms of number of iterations. However, optimal gradient-based sampling is computationally infeasible to compute due to the requirement of computation of full gradient information. The computational overhead of computing sampling probability vector for our algorithms is small and hence it performs better than fixed importance sampling in terms of computation time.
- Algorithm with the adaptive stepsize strategies converges much faster than fixed-stepsize strategies.
- To compute the safe lower and upper bound on the gradient, we only need approximate gradient oracle which provides reasonably good estimates of the gradient. It does need to be very precise.

3.7 Results in "On Matching Pursuit and Coordinate Descent [144]"

3.7.1 Background

In this work, we consider the problem of optimizing convex function over linear spaces. More pontifically, we have following optimization problem:

$$\min_{\mathbf{x}\in\text{lin}(\mathcal{A})} f(\mathbf{x}), \tag{3.27}$$

where f is a convex function and $\text{lin}(\mathcal{A})$ denotes the linear space formed by the linear combination of the elements of $\mathcal{A}$. These elements of set $\mathcal{A}$ are popularly known as *atoms*. Generally, $\mathcal{A}$ is a compact but not necessarily finite subset of Hilbert space. In the literatures, this problem is addressed with the name of matching pursuit. This can be seen as a generalization of coordinate descent where the coordinates do not form orthogonal basis and hence it is allowed for the coordinates/atoms to be linearly dependent. Consider an Hilbert space $\mathcal{H}$ which is associated with inner product $\langle \mathbf{x}, \mathbf{y} \rangle, \forall \mathbf{x}, \mathbf{y} \in \mathcal{H}$. The norm is induced by the inner product as follows, $\|\mathbf{x}\|^2 := \langle \mathbf{x}, \mathbf{x} \rangle, \forall \mathbf{x} \in \mathcal{H}$. We also assume that $\mathcal{A} \subset \mathcal{H}$ is a compact and symmetric set, $f\colon \mathcal{H} \to \mathbb{R}$ is a convex function which is L-smooth.We assume the existence of a linear minimization oracle (LMO) from which MP queries to find the steepest descent direction among the elements of set $\mathcal{A}$ In each iteration:

$$\text{LMO}_{\mathcal{A}}(\mathbf{y}) := \arg\min_{\mathbf{z}\in\mathcal{A}} \langle \mathbf{y}, \mathbf{z} \rangle, \tag{3.28}$$

for a vector $\mathbf{y} \in \mathcal{H}$. Each update looks like the following,

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle}{L\|\mathbf{z}_t\|^2} \mathbf{z}_t$$

where $\mathbf{z}_t := \mathrm{LMO}_{\mathcal{A}}(\nabla f(\mathbf{x}_t))$. However, the above presented algorithm in not affine invariant. We call a method affine invariant if the algorithm is invariant under affine transformation of the input. One can use the atomic norm to define an affine invariant notion of smoothness as follows

$$L_{\mathcal{A}} := \sup_{\substack{\mathbf{x},\mathbf{y} \in \mathrm{lin}(\mathcal{A}) \\ \mathbf{y} = \mathbf{x} + \gamma \mathbf{z} \\ \|\mathbf{z}\|_{\mathcal{A}} = 1, \gamma \in \mathbb{R}_{>0}}} \frac{2}{\gamma^2} \left[f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \right] \tag{3.29}$$

an update of the algorithm looks like

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle}{L_{\mathcal{A}}\|\mathbf{z}_t\|^2} \mathbf{z}_t$$

where $\mathbf{z}_t := \mathrm{LMO}_{\mathcal{A}}(\nabla f(\mathbf{x}_t))$. Following the similar ideas as that of affine invariant smoothness, we can also define the affine invariant notion of strong convexity similarly which is defined below,

$$\mu_{\mathcal{A}} := \inf_{\substack{\mathbf{x},\mathbf{y} \in \mathrm{lin}(\mathcal{A}) \\ \mathbf{x} \neq \mathbf{y}}} \frac{2}{\|\mathbf{y} - \mathbf{x}\|_{\mathcal{A}}^2} D(\mathbf{y}, \mathbf{x}) \,.$$

where $D(\mathbf{y}, \mathbf{x}) := f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$. Based on above defined notion of smoothness, each update of the affine invariant algorithm looks as following,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle}{L_{\mathcal{A}}} \mathbf{z}_t \,. \tag{3.30}$$

We have already mentioned that the matching pursuit algorithm is generalized greedy coordinate descent algorithm. This perspective allows us to use the analysis from accelerated coordinate descent method and extend it for matching pursuit algorithm. In the process, we obtain an accelerated matching pursuit algorithm. The major difficulty however is that it is not known that if accelerate greedy coordinate descent even converges to the global optimum. Hence, we need to make further slight modification in the existing greedy scheme to accelerate *matching* pursuit algorithm. This is achieved by decoupling the updates for $\mathbf{x}$ and $\boldsymbol{b}$ allowing them to be selected from different distributions. Here in this work, we propose to update $\mathbf{x}$ using the greedy update (or the MP update), and use a random direction selected uniformaly at random to update $\boldsymbol{b}$.

3.7.2 Main Results

In the previous section, the relevant background was discussed on the relationship between and coordinate descent. Now, the convergence rate for affine invariant matching pursuit is provided and later will be extended this result to accelerated matching pursuit.

Theorem 3.7.2.1 (Locatello* *et al.* [144]). *Consider $\mathcal{A} \subset \mathcal{H}$ to be a closed and bounded set and $\|\cdot\|_{\mathcal{A}}$ to be norm over* $\text{lin}(\mathcal{A})$. *Let f be a convex function which is $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$ over* $\text{lin}(\mathcal{A})$, *and let $R_{\mathcal{A}}$ be the radius of the level set of $\mathbf{x}_0$ measured with the atomic norm. Then, the optimization error after $t \geq 0$ iterations goes down as following*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star) \leq \frac{2L_{\mathcal{A}}R_{\mathcal{A}}^2}{\delta^2(t+2)},$$

where $\delta \in (0,1]$ is the relative accuracy parameter of the employed approximate LMO.

The rates for coordinate descent can be recovered as a direct corollary from the above theorem statement. Let us consider the case when $\mathcal{A}$ is the L_1-ball in an n dimensional space, then the rate of convergence obtained from Theorem 3.7.2.1 with no approximation in the oracle (exact oracle) can be written as:

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star) \leq \frac{2L_1R_1^2}{t+2} \leq \frac{2L_2R_1^2}{t+2} \leq \frac{2L_2nR_2^2}{t+2},$$

where the first inequality is the rate obtained in this paper, the second inequality is the rate obtained in Stich *et al.* [231] and the last inequality is the rate of vanilla coordinate descent given in [171]. L_2 here denotes the global Lipschitz constant. Therefore, it is clear from the previous argument that measuring the smoothness in atomic norm directly provides a tighter convergence bound. Now, with the notion of affine invariant smoothness and affine invariant strong convexity, one can similarly derive the linear rate of convergence for the matching pursuit algorithm.

Theorem 3.7.2.2 (Locatello* *et al.* [144]). *Consider $\mathcal{A} \subset \mathcal{H}$ to be a closed and bounded set. Let us assume that $\|\cdot\|_{\mathcal{A}}$ is a norm, f be $\mu_{\mathcal{A}}$-strongly convex and $L_{\mathcal{A}}$-smooth function w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$, both over* $\text{lin}(\mathcal{A})$. *Then, optimization error after $t \geq 0$ iterations goes down as following*

$$\varepsilon_{t+1} \leq \left(1 - \delta^2 \frac{\mu_{\mathcal{A}}}{L_{\mathcal{A}}}\right)\varepsilon_t.$$

where $\varepsilon_t := f(\mathbf{x}_t) - f(\mathbf{x}^\star)$.

Accelerated Rates

As we already have discussed the non-accelerated rates, we will now focus on getting accelerated matching pursuit algorithm. So far we have established the insights on the relationship between matching pursuit and coordinate descent algorithm. This insight will

be helpful to us to derive accelerated rate for matching pursuit. Before going directly into discussing the result for accelerated matching pursuit, we first discuss the rate for greedy accelerated coordinate descent algorithm. The rate for greedy accelerated coordinate descent method was not known in the literature. In this work, we propose a semi greedy accelerated coordinate descent by decoupling the two updates of nestrov and choosing one update with the greedy selection while the other one with random selection. Let us assume that the atoms are distributed according to a distribution $\mathcal{Z}$ defined over $\mathcal{A}$. Let us define

$$\tilde{\mathbf{P}} := \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}}[\mathbf{z}\mathbf{z}^\top].$$

Generally here we assume that the distribution $\mathcal{Z}$ is such that $\text{lin}(\mathcal{A}) \subseteq \text{range}(\tilde{\mathbf{P}})$. This condition basically is equivalent to assuming that the probability to sample $\mathbf{z} \sim \mathcal{Z}$ along the direction of every atom $\mathbf{z}_t \in \mathcal{A}$ is not zero for any $\mathbf{z}$ i.e.

$$\mathbb{P}_{\mathbf{z}\sim\mathcal{Z}}[\langle \mathbf{z}, \mathbf{z}_t\rangle > 0] > 0,\ \forall \mathbf{z}_t \in \mathcal{A}.$$

Further, we denote $\mathbf{P} = \tilde{\mathbf{P}}^\dagger$ as the pseudo-inverse of $\tilde{\mathbf{P}}$. With the new psd matrices $\tilde{\mathbf{P}}$ and $\mathbf{P}$, the space can be equipped with the new dot product $\langle \cdot, \mathbf{P}\cdot\rangle$ which results in the following norm $\|\cdot\|_{\mathbf{P}}$. With this new inner product in picture, we have

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{Z}}[\langle \mathbf{z}, \mathbf{P}\mathbf{d}\rangle \mathbf{z}] = \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}}[\mathbf{z}\mathbf{z}^\top]\mathbf{P}\mathbf{d} = \mathbf{P}^\dagger\mathbf{P}\mathbf{d} = \mathbf{d}.$$

The entire algorithm is discussed in the main paper (appendix) and the rate of convergence of the accelerated matching pursuit algorithm is discussed here below.

Theorem 3.7.2.3 (Locatello* *et al.* [144]). *Let f be a convex function and $\mathcal{A}$ be a symmetric compact set. Then the output of (semi-greedy) accelerated matching pursuit algorithm for any $t \geq 1$ converges with the following rate:*

$$\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}^\star) \leq \frac{2L\nu}{t(t+1)}\|\mathbf{x}^\star - \mathbf{x}_0\|^2_{\mathbf{P}}.$$

On similar note, one can derive the convergence rate of the randomized version which is discussed below.

Theorem 3.7.2.4 (Locatello* *et al.* [144]). *Let f be a convex function and $\mathcal{A}$ be a symmetric set. Then the output of the randomized version of accelerated matching pursuit algorithm for any $t \geq 1$ converges with the following rate:*

$$\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}^\star) \leq \frac{2L\nu'}{t(t+1)}\|\mathbf{x}^\star - \mathbf{x}_0\|^2_{\mathbf{P}},$$

where

$$\nu' \leq \max_{\mathbf{d}\in\text{lin}(\mathcal{A})} \frac{\mathbb{E}\left[(\mathbf{z}_t^\top\mathbf{d})^2\|\mathbf{z}_t\|^2_{\mathbf{P}}\right]}{\mathbb{E}\left[(\mathbf{z}_t^\top\mathbf{d})^2/\|\mathbf{z}_t\|^2_2\right]}.$$

3.8 Results in "k-SVRG [238]"

3.8.1 Background

We consider empirical risk minimization problem

$$\mathbf{x}^\star := \arg\min_{\mathbf{x}} f(\mathbf{x}), \qquad \text{with} \qquad f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \tag{3.31}$$

where each $f_i\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth. Relevant background for variance reduced optimization has been discussed in the chapter 1.4. However, for the sake of completeness, we again discuss here the variance reduced update. Afterwards, we will discuss the the result of approach k-SVRG. A unified framework of variance reduced update can be seen as (discussed in [238]) following. Consider $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T$ as the iterates of the algorithm, where $\mathbf{x}_0 \in \mathbb{R}^d$ is the initial point. For each f_i, $i \in [n]$, the corresponding snapshot point is denoted by $\boldsymbol{\theta}_i \in \mathbb{R}^d$. We have the following update,

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{x}_t - \eta g_{i_t}(x_t), \qquad \text{with} \\ g_{i_t}(\mathbf{x}_t) &:= \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\boldsymbol{\theta}_{i_t}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_i), \end{aligned} \tag{3.32}$$

where $\eta > 0$ denotes the stepsize, and $i_t \in [n]$ an index (typically selected uniformly at random from the set $[n]$). SVRG and SAGA both can be seen as a special case of the general template given above.

SVRG As we know that SVRG works by maintaining only one snapshot point $\mathbf{x}$, i.e. $\boldsymbol{\theta}_i = \mathbf{x}$ and its corresponding gradient to the memory. Hence, all the gradient component at this snapshot point can be computed in a cost of one stochastic gradient computation. Hence, this makes the computation cost slightly worse (more by 2-3 times then that of SGD).

SAGA SAGA also takes the same form as given in Equation (3.32). We have $\boldsymbol{\theta}_i \neq \boldsymbol{\theta}_j$ for $i \neq j$ in general. Thus all $\boldsymbol{\theta}_i$ needs to be stored. Hence, the memory requirement of SAGA is large but recomputation can be avoided.

k-SVRG In k-SVRG, we propose to interpolate between SVRG and SAGA by maintaining few snapshot points and saving those gradients at memory. Hence, the memory requirement of k-SVRG is not as worse as that of SAGA. Precisely, we need to maintain few snapshot points $\boldsymbol{\theta} \subset \mathbb{R}^d$ of cardinality $\tilde{\mathcal{O}}(k \log k)$ for k-SVRG. Therefore, it is sufficient to only keep $\boldsymbol{\theta}$ in the memory, and a mapping from each index i to its corresponding element in $\boldsymbol{\theta}$.

In this work, Two variants of k-SVRG are proposed in this work. These variants differ in the way how the snapshot points $\boldsymbol{\theta}_i^m$ are updated at the end of each inner loop.

V1 In k-SVRG-V1, The snapshot points are updated as follows, before moving to the $(m+1)^{th}$ outerloop:

$$\boldsymbol{\theta}_i^{m+1} := \begin{cases} \boldsymbol{\theta}_i^m, & \text{if } i \notin \Phi^m, \\ \tilde{\mathbf{x}}^{m+1}, & \text{otherwise.} \end{cases} \tag{3.33}$$

The set Φ^m is used to keep track of the selected indices in the inner loop. This avoids the storage of $|\Phi^m|$ copies of the the snapshot point $\tilde{x}^{m+1}$ in memory/ Only one point is sufficient.

V2 In k-SVRG-V2(q), q indices are sampled without replacement from $[n]$ at the end of the m^{th} outer loop, which form the set Φ^m, and then update the snapshot points as before in (3.33). The suggested choice of q is $\mathcal{O}(n/k)$, and whenever the argument is dropped, q is simply set to be $q = \ell = \lceil n/k \rceil$.

3.8.2 Main Results

Convex Problems

We consider the μ-strongly convex function f for $\mu > 0$. We study the convergence of the algorithm k-SVRG by studying a suitable Lyapunov function [238]. The Lyapunov function is defined as follows,

$$\mathcal{L}(\mathbf{x}, H) := \|\mathbf{x} - \mathbf{x}^\star\|^2 + \gamma\sigma H\,, \tag{3.34}$$

with $\gamma := \frac{\eta n}{L}$ and $0 \leq \sigma \leq 1$ a constant parameter. We define a sequence of parameters H^m that are updated at the end of each outer loop iteration $\mathbf{x}^m$. Let us also define H_i^m with the property $H_i^m \geq \|\boldsymbol{\alpha}_i^m - \nabla f_i(\mathbf{x}^\star)\|^2$, and thus their sum we call as H^m, *i.e.* $H^m := \frac{1}{n}\sum_{i=1}^n H_i^m$ is an upper bound on $\mathbb{E}\|\boldsymbol{\alpha}_i^m - \nabla f_i(\mathbf{x}^\star)\|^2$. Now, we can properly write H^m as with the help of $h_i^m : \mathbb{R}^d \to \mathbb{R}$ defined below

$$h_i^m(\mathbf{x}) := f_i(\mathbf{x}) - f_i(\mathbf{x}^\star) - \langle \mathbf{x} - \mathbf{x}^\star, \nabla f_i(\mathbf{x}^\star)\rangle\,. \tag{3.35}$$

$\boldsymbol{\alpha}_i^0$ is initialized as 0 *i.e.* $\boldsymbol{\alpha}_i^0 = 0$ and $H_i^0 = \|\nabla f_i(\mathbf{x}^\star)\|^2$ for $i \in [n]$, and then the bounds H_i^m are updated as follows,

$$H_i^{m+1} = \begin{cases} 2Lh_i^m(\tilde{\mathbf{x}}^{m+1}), & \text{if } i \in \Phi^m, \\ H_i^m, & \text{otherwise.} \end{cases} \tag{3.36}$$

Φ^m are the set of indices that are used to compute $\tilde{\mathbf{x}}^{m+1}$. We now provide the convergence result for both the versions of k-SVRG.

Theorem 3.8.2.1 (**Anant Raj** and Stich [238]). *Let $\{\mathbf{x}^m\}_{m\geq 0}$ denote the iterates in the outer loop of k-SVRG-V2(q). If $\mu > 0$, parameter $q \geq \frac{\ell}{3}$, and step size $\eta \leq \frac{1}{3(\mu n+2L)}$ then*

$$\mathbb{E}'_{q,m}\mathcal{L}(\mathbf{x}^{m+1},H^{m+1}) \leq \left(1-\eta\mu\right)^{\ell}\mathcal{L}(\mathbf{x}^m,H^m)\,. \tag{3.37}$$

Theorem 3.8.2.2 (**Anant Raj** and Stich [238]). *Let $\{\mathbf{x}^m\}_{m\geq 0}$ denote the iterates in the outer loop of k-SVRG-V1. If $\mu > 0$, and step size $\eta \leq \frac{2\left(1-\frac{\ell-1}{2n}\right)}{5(\mu n+2L)} < \frac{1}{5(\mu n+2L)}$ then*

$$\mathbb{E}_m\mathcal{L}(\mathbf{x}^{m+1},H^{m+1}) \leq (1-\eta\mu)^{\ell}\mathcal{L}(\mathbf{x}^m,H^m)\,. \tag{3.38}$$

Remark 3.8.2.1 (**Anant Raj** and Stich [238]). *The convergence rate of k-SVRG has the same factor of $(1-\eta\mu)$ as that of SVRG and SAGA. For SAGA, any time convergence can be show. Thus, after ℓ steps, SAGA achieves a decrease of $(1-\eta\mu)^{\ell}$, i.e. of the same order as k-SVRG. On the other hand, the proof for SVRG shows decrease by a constant factor after κ iterations. The same improvement is attained by k-SVRG after $\min\{\lceil n/\ell\rceil,\lceil\kappa/\ell\rceil\}$ inner loops, i.e. $\min\{n,\kappa\}$ total updates. Hence, our rates do not fundamentally differ from the rates of SVRG and SAGA (in case $n \gg \kappa$ we even improve compared to the former method), but they provide an interpolation between both results.*

Non-Convex Problems

For the analysis of non-convex case, the Lyapunov function is chosen as

$$\mathcal{L}^m(\mathbf{x}) := f(\mathbf{x}) + \frac{c^m}{n}\sum_{i=1}^{n}\|\mathbf{x}_0^m - \boldsymbol{\theta}_i^m\|^2\,, \tag{3.39}$$

where $\{c^m\}_{m=0}^M$ denotes a sequence of parameters. If the sequence $\{c^m\}_{m=0}^M$ is defined such that it holds $c^M = 0$ then $\mathcal{L}^M(\mathbf{x}^M) = f(\mathbf{x}^M)$. Quantities $H^m := \frac{1}{n}\sum_{i=1}^n H_i^m$ are defined with $H_i^m := \|\mathbf{x}_0^m - \boldsymbol{\theta}_i^m\|^2$. The sequence $\{c^m\}_{m=0}^M$and an auxiliary sequence $\{\Gamma^m\}_{m=1}^M$ are is initialized that will be used in the proof:

$$c^m := c^{m+1}\left(1-\frac{\ell}{n}+\gamma\eta\ell+4b_1\eta^2L^2\ell^2\right)+2b_1\eta^2L^3\ell\,, \tag{3.40}$$

$$\Gamma^m := \eta - c^{m+1}\frac{\eta}{\gamma} - b_1\eta^2L - 2b_1c^{m+1}\eta^2\ell\,, \tag{3.41}$$

with $b_1 := (1-2L^2\eta^2\ell^2)^{-1}$ and $\gamma \geq 0$ a parameter that will be specified later. As mentioned, $c^M = 0$ will be set to 0 and (3.40) provides the values of c^m for $m = M-1,\ldots,0$. Let us also define the following notation,

$$\mathbb{E}_m\|\nabla F^m\|_F^2 = \sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|\nabla f(x_t^m)\|^2 = \sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2\,. \tag{3.42}$$

Now that Lypapunov function has been defined, the main theoretical result for this section is provided below which show that the algorithm k-SVRG converges to a stationary point for non-convex problems.

Theorem 3.8.2.3 (**Anant Raj** and Stich [238]). *Let* $\{\mathbf{x}_t^m\}_{t=0,m=0}^{\ell-1,M}$ *denote the iterates of* k-SVRG-V2. *Let* $\{c^m\}_{m=0}^{M}$ *be defined as in* (3.40) *with* $c^M = 0$ *and* $\gamma \geq 0$ *and such that* $\Gamma^m > 0$ *for* $m = 0, \ldots, M-1$. *Then:*

$$\sum_{m=0}^{M-1} \mathbb{E}\|\nabla F^m\|_F^2 \leq \frac{f(\mathbf{x}_0^0) - f^\star}{\Gamma}, \tag{3.43}$$

where $\Gamma := \min_{0\leq m\leq M-1} \Gamma^m$. *In particular, for parameters* $\eta = \frac{1}{5Ln^{2/3}}$, $\gamma = \frac{L}{n^{1/3}}$ *and* $\ell = \frac{3}{2}n^{1/3}$ *and* $n > 15$ *it holds:*

$$\sum_{m=0}^{M-1} \mathbb{E}\|\nabla F^m\|_F^2 \leq 15Ln^{2/3}\left(f(\mathbf{x}_0^0) - f^\star\right). \tag{3.44}$$

3.9 Results in "A Simple Approach to Accelerated Stochastic Optimization [104]"

3.9.1 Background

We consider the following composite optimization

$$\text{find } \mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{X}} \ell(\mathbf{x}) = f(\mathbf{x}) + \phi(\mathbf{x}), \tag{3.45}$$

where $\mathcal{X}$ is a convex constraint set in the d-dimensional Euclidean space, f is convex and smooth, and ϕ is a (possibly non-smooth) convex function. *Online linear optimization* (OLO) algorithms have been popularly used to analyze iterative optimization method. An OLO algorithm aims to maintain a small cumulative composite loss $\sum_{t=1}^T \alpha_t \left(\langle \mathbf{u}_t, \mathbf{x}_t - \mathbf{x}\rangle + \phi(\mathbf{x}_t) - \phi(\mathbf{x})\right)$, a.k.a. its *regret* compared to a competitor point $\mathbf{x}$ where $\mathbf{x}_t$ is prediction at time step t and $\langle \alpha_t \mathbf{u}_t, \cdot\rangle$ is the linear loss function. Here $\mathbf{u}_t \in \mathbb{R}^d$ is unknown to the algorithm before selecting $\mathbf{x}_t$, but the non-negative weights α_t are known for all time step t. One can convert an OLO algorithm to an iterative optimization algorithm by using $\mathbf{y}_t = x_t$ to query the oracle, using $u_t = g_t$ where g_t represents the first order gradient information in the linear loss to the OLO algorithm, and employing the average $\bar{\mathbf{x}}_T = \sum_{t=1}^T \frac{\alpha_t}{\alpha_{1:T}}\mathbf{x}_t$ as the final estimate of $\mathbf{x}^\star$.

An alternative, elegant online-to-batch conversion (algorithm 2) was recently proposed by Cutkosky [50], which uses the "online" average $\bar{\mathbf{x}}_t = \sum_{s=1}^t \frac{\alpha_s}{\alpha_{1:t}}\mathbf{x}_s$ as the query point, i.e., $\mathbf{y}_t = \bar{\mathbf{x}}_t$. Cutkosky [50, Theorem 1] showed similar reduction holds under this conversion

Algorithm 2 Anytime Online-to-Batch [50]

Input: Stochastic gradient oracle, non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, online linear optimization algorithm $\mathcal{A}$
Get the initial point $\mathbf{x}_1 \in \mathcal{X}$ from $\mathcal{A}$ and let $\bar{\mathbf{x}}_1 \leftarrow \mathbf{x}_1$
for $t = 1$ **to** $T-1$ **do**
    Get stochastic gradient g_t at the *average* iterate $\bar{\mathbf{x}}_t$
    Send $\langle \alpha_t \mathbf{g}_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
    Let $\mathbf{x}_{t+1}$ be the next iterate from $\mathcal{A}$
    Let $\bar{\mathbf{x}}_{t+1} \leftarrow \frac{\sum_{s=1}^{t+1} \alpha_s \mathbf{x}_s}{\alpha_{1:t+1}}$
end for
return the average iterate $\bar{\mathbf{x}}_T$

scheme as well. Next,the tighter results obtained in the framework of algorithm 2 is discussed that enables to prove accelerated rates.

3.9.2 Main Results

So far, the background and framework have been discussed for this section where the results will be provided. Now, the main results presented in this work will be discussed. The result provided below is crucial in proving all the further results.

Lemma 3.9.2.1 (Joulani* *et al.* [104]). *For $t = 1, 2, \ldots T$, let $\alpha_t > 0$ and $\mathbf{x}_t \in \mathbb{R}^d$, and define $\bar{\mathbf{x}}_t = (\sum_{s=1}^t \alpha_s \mathbf{x}_s)/\alpha_{1:t}$, $B_t = \alpha_t B_f(\mathbf{x}^\star, \bar{\mathbf{x}}_t)$, and $\bar{B}_t^f = \alpha_{1:t-1} B_f(\bar{\mathbf{x}}_{t-1}, \bar{\mathbf{x}}_t), t > 1$. Then, if ϕ is convex,*

$$\alpha_{1:T}\left(\ell(\bar{\mathbf{x}}_T) - \ell^\star\right) \leq \sum_{t=1}^{T} \alpha_t \left(\left\langle f'(\bar{\mathbf{x}}_t), \mathbf{x}_t - \mathbf{x}^\star \right\rangle + \phi(\mathbf{x}_t) - \phi(\mathbf{x}^\star)\right) - B_{1:T} - \bar{B}_{2:T}^f, \tag{3.46}$$

where Bregman-divergence $B_h : \mathcal{D} \times \mathcal{D}^o \to \mathbb{R}$ is defined as $B_h(\mathbf{x}, \mathbf{y}) = h(\mathbf{x}) - h(\mathbf{y}) - \langle h'(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$.

The lemma immediately gives rise to the following generic error bound, which improves upon Theorem 1 of Cutkosky [50] by keeping around the aforementioned $-\bar{B}_t^f$ and $-B_t$ terms which allow to prove accelerated rates for online averaging.

Corollary 3.9.2.2 (Generic Error Bound [104]). *Under the assumptions of Lemma 3.9.2.1, if for all $t = 1, 2, \ldots, T$, $g_t \in \mathbb{R}^d$ satisfies $\mathbb{E} g_t | \bar{\mathbf{x}}_t = f'(\bar{\mathbf{x}}_t)$ and then*

$$\sum_{t=1}^{T} \alpha_t \left(\langle \mathbf{g}_t, \mathbf{x}_t - \mathbf{x}^\star \rangle + \phi(x_t) - \phi(\mathbf{x}^\star)\right) \leq \mathcal{R}_T(x^\star) \tag{3.47}$$

for some upper-bound $\mathcal{R}_T(x^\star)$, then

$$\mathbb{E}\ell(\bar{\mathbf{x}}_T) - \ell(\mathbf{x}^\star) \leq \mathbb{E}\frac{\mathcal{R}_T(\mathbf{x}^\star) - B_{1:T} - \bar{B}^f_{2:T}}{\alpha_{1:T}}. \tag{3.48}$$

The main idea behind deriving accelerated rates is combining the regret bound of optimistic AO-FTRL update withthe generic error bound obtained in the previous lemma, and selecting α_t and $\tilde{g}_t$ appropriately so that the negative terms $-\bar{B}^f_t$ in (3.48) offset the contribution of significant positive term in the final error bound of $\bar{\mathbf{x}}_T$. More formal result is given in the next theorem:

Theorem 3.9.2.3 (Joulani* *et al.* [104]). *In algorithm 2, let the base method $\mathcal{A}$ generate its iterates by the AO-FTRL update, using $\tilde{\mathbf{g}}_t = \mathbf{g}_{t-1}$ as the optimistic prediction of $\mathbf{g}_t$ for $t > 1$ and arbitrary $\tilde{\mathbf{g}}_1$. Suppose that f and ϕ are convex, and there exists a norm $\|\cdot\|$ such that f is 1-smooth w.r.t. $\|\cdot\|$ over $\mathbb{R}^d$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Further suppose that for all $t \in [T]$, $r_{t-1} \geq 0$ is convex, the AO-FTRL update is well-defined with finite value at the optimum $\mathbf{x}_t$, and there exist $\beta_t > 0$ and a norm $\|\cdot\|_{(t)}$ such that $\alpha_{1:t}\phi + r_{0:t-1}$ is 1-strongly-convex w.r.t. $\frac{\beta_t}{2}\|\cdot\|^2 + \frac{1}{2}\|\cdot\|^2_{(t)}$. Then, if $\alpha_t^2\beta_t^{-1} \leq \alpha_{1:t-1}$ for all $t > 1$, then*

$$\mathbb{E}\ell(\bar{\mathbf{x}}_T) - \ell^\star \leq \sum_{t=1}^T \mathbb{E}\frac{r_{t-1}(\mathbf{x}^\star) - r_{t-1}(\mathbf{x}_t) - B_t}{\alpha_{1:T}} + \sum_{t=1}^T \mathbb{E}\frac{\alpha_t^2\|\sigma_t - \sigma_{t-1}\|^2_{(t)*}}{2\alpha_{1:T}} + \mathbb{E}\frac{\alpha_1^2\|f'(\bar{\mathbf{x}}_1) - \tilde{\mathbf{g}}_1\|^2_*}{2\beta_1\alpha_{1:T}}, \tag{3.49}$$

where $\sigma_t = \mathbf{g}_t - f'(\bar{\mathbf{x}}_t), t \in [T]$, and $\sigma_0 = 0$.

The theorem statment discussed above is very important and will be utilized to in most of the results discussed in the paper. This result also provide an intuitive but simple explanation of nestrov's accelerated gradient method. In the next corollary, with appropriately setting α_t and η_t, one can obtain the optimal accelerated rates for the proximal dual averaging update.

Corollary 3.9.2.4 (Accelerated Proximal Dual-Averaging [104]). *Let f and ϕ be convex and assume that either f is L-smooth over $\mathbb{R}^d$. Consider the online-averaged (stochastic) proximal dual averaging algorithm, given by Algorithm 2 with proximal SGD update using $\tilde{\mathbf{g}}_t = \mathbf{g}_{t-1}$ as the optimistic prediction of $\mathbf{g}_t$ for $t > 1$, and $\tilde{\mathbf{g}}_1 = 0$, where the gradient estimates g_t are unbiased, that is, $\mathbb{E}[\mathbf{g}_t|\bar{\mathbf{x}}_t] = f'(\bar{\mathbf{x}}_t)$. Let $\sigma_*^2 = \max_{t=1}^T \mathbb{E}\|\sigma_t\|_2^2$, where $\sigma_t = \mathbf{g}_t - f'(\bar{\mathbf{x}}_t)$, and let $D = \max\{\|\mathbf{x}^\star\|_2, \|\mathbf{x}_1 - \mathbf{x}_f^*\|_2\}$, where $\mathbf{x}_f^*$ is the minimizer of f over $\mathbb{R}^d$. Then the following error bounds hold:*

(i) If $\eta_t = 4L + \eta\alpha_t\sqrt{t}$ for some $\eta > 0$ and $\alpha_t = t$, then

$$\mathbb{E}[\ell(\bar{\mathbf{x}}_T)] - \ell^\star \leq \frac{\left(4L + \frac{L}{4} + \eta T\sqrt{T}\right)D^2 + \frac{4\sigma_*^2}{\eta}T\sqrt{T}}{T(T+1)} = \mathcal{O}\left(\frac{LD^2}{T^2} + \frac{\eta D^2 + \eta^{-1}\sigma_*^2}{\sqrt{T}}\right).$$

(ii) If ϕ_t is μ-strongly-convex then using $\eta_t = 4L$ and $\alpha_t = t$, then

$$\mathbb{E}[\ell(\bar{\mathbf{x}}_T)] - \ell^\star \leq \frac{\left(4L + \frac{L}{4}\right) D^2 + \frac{8\sigma_*^2 T}{\mu}}{T(T+1)} = \mathcal{O}\left(\frac{LD^2}{T^2} + \frac{\sigma_*^2}{\mu T}\right).$$

(iii) If $\mathbf{g}_t = f'(\mathbf{x}_t)$ (i.e., the noiseless case) and ϕ is μ-strongly-convex, then for $\eta_t = 0$ and any sequence of $\alpha_t > 0, t \in [T]$ satisfying

$$\sqrt{c\kappa} \geq \frac{\alpha_{1:t}}{\alpha_t} \geq \sqrt{2\kappa} \qquad t > 1\,, \tag{3.50}$$

for some $c \geq 2$ where $\kappa = (L+\mu)/\mu$ denotes the condition number, then

$$\ell(\bar{\mathbf{x}}_T) - \ell^\star \leq \frac{\|f'(\mathbf{x}_1)\|^2 \left(1 - \frac{1}{\sqrt{c\kappa}}\right)^{T-1}}{2\mu}. \tag{3.51}$$

Next, the universal convergence of algorithm 2 is presented with AdaGrad-style step sizes.

Theorem 3.9.2.5 (Joulani* *et al.* [104])**.** *Suppose that the iterates x_t are given by AO-FTRL with AdaGrad step sizes, i.e., using AO-FTRL update with $r_0 = 0$,*

$$r_t(\mathbf{x}) = \gamma \sum_{j=1}^{d} \frac{\eta_{t,j} - \eta_{t-1,j}}{2} (x_j - x_{t,j})^2, \qquad t \geq 1\,,$$

where $\gamma > 0$, $\eta_{t,j} = \sqrt{\sum_{s=1}^{t} \alpha_s^2 (g_{s,j} - \tilde{g}_{s,j})^2}, t > 0$ and $\eta_0 = 0$. Further suppose that g_t are unbiased estimates of $f'(\bar{\mathbf{x}}_t)$, and $\tilde{\mathbf{g}}_t = \mathbf{g}_{t-1}, t > 1$ as well as $\tilde{\mathbf{g}}_1 = 0$. Let R be an upper-bound on $|x_j^\star - x_{t,j}|^2$. Then the following hold:

(i) If $\mathbb{E}\mathbf{g}_{t,j}^2 \leq G_j^2$ for all $t \in [T]$, then

$$\mathbb{E}\ell(\bar{\mathbf{x}}_T) - \ell^\star \leq \sum_{j=1}^{d} \mathbb{E}\left[\frac{\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)}{\alpha_{1:T}} \sqrt{\sum_{t=1}^{T} \alpha_t^2 G_{t,j}^2}\right] = \mathcal{O}\left(\frac{R \sum_{j=1}^{d} G_j}{\sqrt{T}}\right),$$

for $\gamma = 2/R$, where $G_{t,j} := (g_{t,j} - \tilde{g}_{t,j})$.

(ii) If f is L-smooth over $\mathbb{R}^d$, and $\mathbb{E}\sigma_{t,j}^2 \leq \sigma_j^2$ for all $t \in [T]$ (recall that $\sigma_t = \mathbf{g}_t - f'(\bar{\mathbf{x}}_t)$), then

$$\mathbb{E}\ell(\bar{\mathbf{x}}_T) - \ell^\star \leq \frac{1}{\alpha_{1:T}} \sum_{j=1}^{d} 6L \left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)^2$$

$$+\frac{1}{\alpha_{1:T}}\left(\frac{\gamma R^2}{2}+\frac{2}{\gamma}\right)\left(\Delta+\sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T}6\alpha_t^2\sigma_j^2}\right)$$
$$=\mathcal{O}\left(\frac{LdR^2+\Delta R}{T^2}+\frac{\max_j\sigma_j dR}{\sqrt{T}}\right),$$

for $\gamma=2/R$, *where* $\Delta=\sum_{j=1}^{d}\sqrt{2\mathbb{E}|f'(x_{1,j})|^2}$.

Accelerated Variance-Reduced Methods: The framework is applied to the variance reduced setting. In this setting, $f=\mathbb{E}[F(\cdot,\xi)]$ is assumed to be the expected value of functions $F:\mathbb{R}^d\times\Xi\to\mathbb{R}$, where ξ is a random variable from some set Ξ, with distribution P_Ξ. At time step t, the algorithm receives a realization $\zeta_t\sim P_\Xi$, and can query the gradient oracle $F'(\cdot,\zeta_t)$ at (potentially multiple) points in $\mathcal{X}$. In addition, the algorithm can query the exact (non-stochastic) gradient oracle f' from time to time as discussed in *Variance Reduced Method* section (Section 1.4).

Theorem 3.9.2.6 (Joulani* *et al.* [104])**.** *Suppose that* f, *as well as* $F(\cdot,\zeta)$ *for all* $\zeta\in\Xi$, *are a) convex; and, b) either L-smooth w.r.t.* $\|\cdot\|_2$ *over* $\mathbb{R}^d$. *Further suppose that variance reduced gradient estimate is unbiased. Assume that variance reduction update (Section 1.4) is run with epoch lengths* $T_s=\min\{\tau,2^{s-1}\}$ *for some maximum epoch length* τ *after which snapshot is updated* $\tilde{\mathbf{x}}=\bar{\mathbf{x}}_{t+1}$ *and full gradient is computed,* $\alpha_t=t$, *and* $\mathcal{A}$ *selected as AO-FTRL with regularizer* $r_{1:t-1}=\frac{\eta_t}{2}\|\cdot\|_2^2$ *for* $\eta_t=8L\tau^2$ *and optimistic gradient estimates* $\tilde{\mathbf{g}}_1=0$ *and* $\tilde{\mathbf{g}}_t=f'(\tilde{\mathbf{x}}_{t-1}),t>1$. *Then, for any* $T>\tau$,

$$\mathbb{E}\ell(\bar{\mathbf{x}}_T)-\ell(x^\star)\leq\frac{8L\tau^2\|\mathbf{x}^\star\|_2^2+\frac{\|f'(\bar{x}_1)\|_2^2}{8L\tau^2}}{T(T+1)}.$$

3.10 Results in "Importance Sampling via Local Sensitivity [240]"

3.10.1 Background

In this work, finite sump optimization of specific form (empirical risk minimization) is considered. Let us consider the data points $\mathbf{a}_1,\ldots,\mathbf{a}_n\in\mathbb{R}^d$, non-negative convex functions $f_1,\ldots,f_n:\mathbb{R}\to\mathbb{R}^+$, and a non-negative function $\gamma:\mathbb{R}^d\to\mathbb{R}^+$ which usually the regularizer. We have following optimization problem formulation which would be minimized over $\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^d$

$$f(x):=\frac{1}{n}\sum_{i=1}^{n}f_i(\mathbf{a}_i^T\mathbf{x})+\gamma(\mathbf{x}). \tag{3.52}$$

However, for large data sets, the task of optimizing the finite sum objective given above is computationally cumbersome. Hence, it is necessary to come up with the approximation of f which is easier to minimize and whose minimizer is not very far from the true minimizer of f. This approximation of the finite sum function f can be obtained by independently subsampling data points a_i with some fixed probability weights and sum the weighted $f_i(\mathbf{a}_i^T\mathbf{x})$ for all the subsampled points. More formally, if we have a target sample size m, a probability distribution $P = \{p_1, \ldots, p_n\}$ over $[n] \triangleq \{1, \ldots, n\}$, then we select $i_1, ..., i_m$ i.i.d. from P to form the approximation of the true function which can be written as follows,

$$F^{(P,m)}(\mathbf{x}) := \frac{1}{mn}\sum_{j=1}^{m}\frac{f_{i_j}(\mathbf{a}_{i_j}^T\mathbf{x})}{p_{i_j}} + \gamma(\mathbf{x}). \tag{3.53}$$

We minimize the objective function $F^{(P,m)}(\mathbf{x})$ over $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ to get the approximately optimal solution. This has been discussed already in the section 1.4 of Chapter 1 that for any $\mathbf{x}$, we have $\mathbb{E}[F^{(P,m)}(\mathbf{x})] = f(x)$. If the sampled function stays close to the true function $f(\mathbf{x})$ for all $\mathbf{x}$, then it can serve effectively as a surrogate for minimizing f. One trivial idea to form such approximation is by choosing the data points uniformly at random from the dataset to form the approximate objective function $F^{(P,m)}(\mathbf{x})$. However, uniform sampling assigns same probability weights to all the data points and it is very likely that uniform sampling will miss those data points which have higher contributions in the loss functions. A possible solution to this problem was proposed in the form of sensitivity sampling which we discussed in section 1.4 of Chapter 1. However, there are two major issues which stops this approach from being widely used in practice:

1. It is almost impossible to compute or even approximate the sensitivity $\sigma_{f,\mathcal{X}}(\mathbf{a}_i)$ for most of the loss functions due to the fact that the supremum over all $\mathbf{x} \in \mathcal{X}$ in the expression of Definition 1.4.0.1 can not be computed for most of the loss functions. The knowledge of close form expression of the sensitivity is only limited to few loss functions such as least squares regression.

2. Since, usually the domain $\mathcal{X}$ is huge, the supremum over all $\mathbf{x} \in \mathcal{X}$ of $\frac{f_i(\mathbf{a}_i^T\mathbf{x})}{\sum_{j=1}^{n} f_j(\mathbf{a}_j^T\mathbf{x}) + n\gamma(\mathbf{x})}$ is a large quantity. Because of this 'worst case' importance metric, the sensitivity scores for all the data points are usually high which results in higher sum of total sensitivity which essentially make the sample complexity bound worse *i.e.* large sampling requirement for better approximation.

3.10.2 Main Result

In this work, the idea of *local sensitivity* is proposed to overcome the above bariers. Local sensitivity considers the idea to approximate the function in a small ball of the domain $\mathcal{X}$.

This is achieved by considering the definition of the sensitivity over a small ball instead of the full domain $\mathcal{X}$ as in Definition 1.4.0.1. This approach has two major advantages over the previous sensitivity based sampling approach:
1. Each function f can be often locally approximated by a simple function for which one can can compute the local sensitivities in closed form. This will result to get the approximation of the true local sensitivities. In this work, we considet a local quadratic approximation to f due to the fact that the sensitivities of a quadratic function can be given in closed form by the *leverage scores* of a modified matrix which is trivial to compute.
2. The value of local sensitivity $\sigma_{f,\mathcal{X}\cap B(r,\mathbf{y})}$ is *always* smaller in number than global sensitivity $\sigma_{f,\mathcal{X}}$, and hence the sum of local sensitivities will also be much smaller in comparison to the sum of local which we call as the total sensitivity $\mathcal{G}_{f,\mathcal{X}}$. Since, the number of samples required to find a good approximation directly depends on the sum of sensitivities, hence we need to sample fewer samples to approximately minimize F locally over $B(r,\mathbf{y})$.

Once, we have computed the local sensitivities for all the data points, we would use in the conjunction an iterative optimization algorithm to get the desired rate of convergence. The iterative method presented in this work uses a proximal function, and thus in this section we consider the proximal function function defined below in definition 3.10.2.1. Proximal function reduces to f when $\lambda = 0$:

Definition 3.10.2.1 (Proximal Function)**.** For a function $f : \mathcal{X} \to \mathbb{R}$, define $f_{\lambda,y}(\mathbf{x}) = f(\mathbf{x}) + \lambda \|\mathbf{x}-\mathbf{y}\|_2^2$.

Using the ideas from leverage score sampling, the following result is established.

Theorem 3.10.2.1 (Sensitivity of Quadratic Approximation, [240])**.** *Consider f as a finite sum function (Equation (3.52)) with the quadratic approximation to the proximal function $f_{\lambda,\mathbf{y}}$ around $\mathbf{y} \in \mathcal{X}$ denoted by $\tilde{f}_{\lambda,y}(\mathbf{x})$. If $A \in \mathbb{R}^{n\times d}$ is the data matrix with i^{th} row equal to $\mathbf{a}_i$, then*

$$\begin{aligned}\tilde{f}_{\lambda,y}(\mathbf{x}) &:= \frac{1}{n}\sum_{i=1}^{n}\left[f_i(\mathbf{a}_i^T\mathbf{y}) + a_i^T(\mathbf{x}-\mathbf{y})\cdot f'(\mathbf{a}_i^T\mathbf{y}) + \frac{1}{2}(\mathbf{a}_i^T(\mathbf{x}-\mathbf{y}))^2\cdot f''(\mathbf{a}_i^T\mathbf{y})\right] + \gamma(\mathbf{x}) + \lambda\|\mathbf{x}-\mathbf{y}\|_2^2 \\ &:= f(\mathbf{y}) + (\mathbf{x}-\mathbf{y})^T A^T\alpha_y + \frac{1}{2}(\mathbf{x}-\mathbf{y})^T A^T H_y A(\mathbf{x}-\mathbf{y}) + \gamma(x) + \lambda\|\mathbf{x}-\mathbf{y}\|_2^2\end{aligned} \tag{3.54}$$

where $[\alpha_y]_i = \frac{1}{n}f_i'(\mathbf{a}_i^T\mathbf{y})$, and H_y is the diagonal matrix with $[H_y]_{i,i} = \frac{1}{n}f''(\mathbf{a}_i^T\mathbf{y})$. Assuming that H_y is nonnegative, the sensitivity scores of $\tilde{f}_{\lambda,\mathbf{y}}$ with respect to $B(r,\mathbf{y})$ can be bounded as

$$\sigma_{\tilde{f}_{\lambda,\mathbf{y}},B(r,\mathbf{y})}(\mathbf{a}_i) \leq \beta\cdot\ell_i^{\lambda}(C) + \frac{f_i(\mathbf{a}_i^T\mathbf{y})}{\eta}, \tag{3.55}$$

where $C = [H_y^{1/2}A, \frac{1}{\delta}H_y^{-1/2}\alpha_y]$, $\ell_i^{\lambda}(C)$ *is the leverage score,* $\eta = \min_{\mathbf{x}\in B(r,\mathbf{y})} \tilde{f}_{\lambda,\mathbf{y}}(\mathbf{x})$, $\delta = \min_{\mathbf{x}\in B(r,\mathbf{y})} \gamma(x)$, *and* $\beta = \max\left(1, 1 - \frac{f(\mathbf{y}) - \frac{1}{n}\sum_{i=1}^{n} \frac{f'(\mathbf{a}_i^T\mathbf{y})^2}{4f''(\mathbf{a}_i^T\mathbf{y})}}{\eta}\right)$.

If we consider the size of the ball where we are approximating $f_{\lambda,\mathbf{y}}$ with $\tilde{f}_{\lambda,\mathbf{y}}$ small enough then the approximation quality of $f_{\lambda,\mathbf{y}}$ by $\tilde{f}_{\lambda,\mathbf{y}}$ is very good. In such cases, we hope the following to hold: $\eta = \min_{\mathbf{x}\in B(r,\mathbf{y})} \tilde{f}_{\lambda,\mathbf{y}}(\mathbf{x}) = \Theta(f(\mathbf{y}))$. For this reason, the additive $\frac{f_i(\mathbf{a}_i^T\mathbf{y})}{\eta}$ term will contribute a very little additive factor of $\frac{\sum f_i(\mathbf{a}_i^T\mathbf{y})}{\Theta(f(\mathbf{y}))} = O(1)$ in the total sum of sensitivities which essentially will not affect the final sample complexity bound by a lot.

If the approximation of the $f_{\lambda,\mathbf{y}}$ by $\tilde{f}_{\lambda,\mathbf{y}}$ is good enough then on the ball $B(r,\mathbf{y})$ then one can apply the result of Theorem 3.10.2.1 to approximate the true local sensitivity $\sigma_{f_{\lambda,\mathbf{y}},\mathcal{X}\cap B(r,\mathbf{y})}(\mathbf{a}_i)$. Let's discuss the assumption made to get the rest of the results in this work. For a function f which has a C Lipschitz-Hessian, we have:

$$f(\mathbf{x}) \leq f(\mathbf{y}) + (\mathbf{x}-\mathbf{y})^\top A^\top \alpha_y + (\mathbf{x}-\mathbf{y})^\top A^\top H_y A(\mathbf{x}-\mathbf{y}) + \gamma(\mathbf{x}) + \frac{C}{6}\|\mathbf{x}-\mathbf{y}\|_2^3. \quad (3.56)$$

After the C Lipschitz-Hessian assumption made, the next result to obtain an upper bound on the local sensitivity is discussed below.

Theorem 3.10.2.2 (**Anant Raj** *et al.* [240]). *Consider f as as a finite sum function, $f_{\lambda,\mathbf{y}}$ as in 3.10.2.1, $\mathbf{y} \in \mathcal{X}$, a radius r, and $\alpha = \min_{\mathbf{x}\in B(r,\mathbf{y})} f_{\lambda,\mathbf{y}}(\mathbf{x})$. Then, $\forall\, i \in [n]$,*

$$\sigma_{f_{\lambda,\mathbf{y}},B(r,\mathbf{y})}(\mathbf{a}_i) \leq \sigma_{\tilde{f}_{\lambda,\mathbf{y}},B(r,\mathbf{y})}(\mathbf{a}_i) + \min\left(\frac{C_i r}{6n\lambda}, \frac{C_i r^3}{6n\alpha}\right).$$

From the above result, we have the bound on true local sensitivities which can be utilized to independently sample components resulting to obtaining a $(1+\varepsilon)$ approximation of the function $f_{\lambda,y}(\mathbf{x})$. This approximation will be used repetitively in the sense of approximate proximal point method to find the optimal point $\mathbf{x}^\star$ upto δ accuracy for some $\delta > 0$ which depends on ε. Each black box optimization oracle look like the following while applying approximate proximal point methods.

$$\mathbf{x}_t \leftarrow P_{f_{\lambda_t,\mathbf{x}_{t-1}}}(\mathbf{x})$$

where $P_{f_{\lambda_t,\mathbf{x}_{t-1}}}(\mathbf{x})$ is defined below.

Definition 3.10.2.2 (**Anant Raj** *et al.* [240]). An algorithm $\mathcal{P}_f$ is called *multiplicative* ε-oracle for a given function f if $f(\mathbf{x}^\star) \leq f\big(\mathcal{P}_f(\mathbf{x})\big) \leq (1+\varepsilon)f(\mathbf{x}^\star)$ where $\mathbf{x}^\star$ if the true minimizer of f.

Now that algorithm has been defined, convergence bounds for Approximate Proximal Point Method would be stated below with a blackbox multiplicative ε-oracle.

Theorem 3.10.2.3 (**Anant Raj** *et al.* [240]). *For μ-strongly convex f, consider $\varepsilon_1, \dots \varepsilon_T \in (0,1)$ and $\mathbf{x}_0, \dots, \mathbf{x}_T \in \mathbb{R}^d$ such that $\mathbf{x}_t = P_{f_{\lambda_t, \mathbf{x}_{t-1}}}(\mathbf{x}_t - 1)$ where $P_{f_{\lambda_t, \mathbf{x}_{t-1}}}$ is an ε_t-oracle. Then if $\varepsilon_t \leq \frac{\mu}{\mu+\lambda_t} \forall t \in [T]$, we have $f(\mathbf{x}_t) - f^\star \leq \frac{1}{1-\varepsilon_t}\frac{\lambda_t}{\mu+\lambda_t}(f(\mathbf{x}_t - 1) - f^\star) + \frac{\varepsilon_t}{1-\varepsilon_t} f^\star \; \forall t \in [T]$ and*

$$f(\mathbf{x}_T) - f^\star \leq \rho(f(\mathbf{x}_0) - f^\star) + \delta f^\star$$

where $\rho = \prod_{t=1}^{T} \frac{1}{1-\varepsilon_t}\frac{\lambda_t}{\mu+\lambda_t}$ and $\delta = \sum_{t=1}^{T}\left(\frac{\varepsilon_t}{1-\varepsilon_t}\prod_{j=t+1}^{T}\frac{1}{1-\varepsilon_t}\frac{\lambda_j}{\mu+\lambda_j}\right)$.

Theorem 3.10.2.4 (**Anant Raj** *et al.* [240]). *For a smooth convex function f, let $\varepsilon_1, \dots, \varepsilon_T = \varepsilon$ where $\varepsilon \in (0, 1/2)$ and $\mathbf{x}_0, \dots, \mathbf{x}_T \in \mathbb{R}^d$ be as in Theorem 3.10.2.3. Then, we have*

$$f(\mathbf{x}_T) - f^\star \leq \frac{2}{(1-\varepsilon)} \frac{\|\mathbf{x}^\star - \mathbf{x}_0\|_2^2}{\sum_{t=1}^{T}\frac{2}{\lambda_t}} + \frac{3\varepsilon}{1-\varepsilon} f^\star.$$

3.11 Results in "Explicit Regularization of Stochastic Gradient Methods through Duality [237]"

3.11.1 Background

In the interpolation regime, we have the following finite sum objective

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

with respect to $\mathbf{x} \in \mathbb{R}^d$, where the global minimizer of f is also a global minimizer of *all* functions f_i, for $i \in \{1, \dots, n\}$. Hence, our goal in the interpolation regime reduces to find a point $\mathbf{x} \in \mathbb{R}^d$ in the intersection of all sets of minimizers

$$\mathcal{K}_i = \arg\min_{\eta \in \mathbb{R}^d} f_i(\eta),$$

for all $i \in \{1, \dots, n\}$. Hence, we can enforce a geometric constraint of our own choice on the optimal solution in the form of explicit regularization which has the following optimization problem formulation:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \psi(\mathbf{x}) \text{ such that } \forall i \in \{1, \dots, n\}, \; \mathbf{x} \in \mathcal{K}_i, \tag{3.57}$$

where ψ is a regularization function. In the optimization problem formulation given in Eq. (3.57), explicit regularization in the solution can be introduced via the function ψ.

Considering the case of finite data , one can rewrite more specific version of the problem as following:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \Psi(\mathbf{x}) \text{ such that } \forall i \in \{1,\dots,n\},\ \mathbf{a}_i^\top \mathbf{x} \in \mathcal{Y}_i, \tag{3.58}$$

where:

- Regularizer / mirror map: $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a differentiable as well as μ-strongly convex function with respect to some norm $\|\cdot\|$ (which is not in general the ℓ_2-norm). The associated Bregman divergence with respect to ψ is [33] defined as
$$D_\Psi(\mathbf{x},\eta) = \psi(\mathbf{x}) - \psi(\eta) - \psi'(\eta)^\top(\mathbf{x}-\eta).$$
- Data: $\mathbf{a}_i \in \mathbb{R}^{d\times k}$, $\mathcal{Y}_i \subset \mathbb{R}^k$ are closed convex sets, for $i \in \{1,\dots,n\}$.
- Feasibility / interpolation regime: It is assumed that there exists $\mathbf{x} \in \mathbb{R}^d$ such that $\psi(\mathbf{x}) < \infty$ and $\forall i \in \{1,\dots,n\},\ \mathbf{a}_i^\top \mathbf{x} \in \mathcal{Y}_i$.

3.11.2 Main Results

The support function $\sigma_{\mathcal{Y}_i}$ of the convex set $\mathcal{Y}_i$ would be needed which is defined as, for $\boldsymbol{\alpha}_i \in \mathbb{R}^k$ [30],

$$\sigma_{\mathcal{Y}_i}(\boldsymbol{\alpha}_i) = \sup_{y_i\in\mathcal{Y}_i} y_i^\top \boldsymbol{\alpha}_i.$$

By Fenchel duality:

$$\begin{aligned} & \min_{\mathbf{x}\in\mathbb{R}^d} \psi(\mathbf{x}) \text{ such that } \forall i \in \{1,\dots,n\},\ \mathbf{a}_i^\top \mathbf{x} \in \mathcal{Y}_i && (3.59)\\ = & \min_{\mathbf{x}\in\mathbb{R}^d} \psi(\mathbf{x}) + \frac{1}{n}\sum_{i=1}^n \max_{\boldsymbol{\alpha}_i\in\mathbb{R}^k} \Big\{ \boldsymbol{\alpha}_i^\top \mathbf{a}_i^\top \mathbf{x} - \sigma_{\mathcal{Y}_i}(\boldsymbol{\alpha}_i) \Big\} \\ = & \max_{\forall i,\ \boldsymbol{\alpha}_i\in\mathbb{R}^k} -\frac{1}{n}\sum_{i=1}^n \sigma_{\mathcal{Y}_i}(\boldsymbol{\alpha}_i) - \psi^\star\Big(-\frac{1}{n}\sum_{i=1}^n \mathbf{a}_i\boldsymbol{\alpha}_i\Big), && (3.60) \end{aligned}$$

with, at optimality,

$$\mathbf{x}^\star = \mathbf{x}(\boldsymbol{\alpha}^\star) = \nabla\psi^\star\Big(-\frac{1}{n}\sum_{i=1}^n \mathbf{a}_i\boldsymbol{\alpha}_i\Big).$$

$G(\boldsymbol{\alpha})$ denotes the dual objective function above. In this work, dual algorithms are considered to solve the problem discussed earlier. The first and main result in this work is to provide some primal guarantees from $\mathbf{x}(\boldsymbol{\alpha})$.

Proposition 3.11.2.1 (**Anant Raj** and Bach [237]). *With the assumption, for any* $\boldsymbol{\alpha} \in \mathbb{R}^{n\times k}$*, it gives:*

$$D_{\Psi}(\mathbf{x}^{\star}, \mathbf{x}(\boldsymbol{\alpha})) \leqslant \text{gap}(\boldsymbol{\alpha}).$$

As a direct consequence of previous proposition and coordinate descent result, the following holds

$$\mathbb{E}\Big[D_{\Psi}(\mathbf{x}^{\star}, \mathbf{x}(\boldsymbol{\alpha}^{(t)}))\Big] \leqslant \mathbb{E}\Big[\text{gap}(\boldsymbol{\alpha}^{(t)})\Big] \leqslant \frac{\max_i L_i}{t} \frac{\max\{\|\boldsymbol{\alpha}^{\star}\|^2, \mathcal{R}(0)^2\}}{n}, \tag{3.61}$$

where L_i is smoothness constant and $\mathcal{R}(\boldsymbol{\alpha}) = \max_y \max_{\boldsymbol{\alpha}^{\star}\in A^{\star}} \{\|y - \boldsymbol{\alpha}^{\star}\| \ : \ G(y) \geq G(\boldsymbol{\alpha})\}$.

Relationship to Least Square: Least-squares in the overparametrized regime where the we obtain zero training error (interpolation regime) can be written as a finite sum objective as follows,

$$\min \left[\frac{1}{2n}\sum_{i=1}^{n} \|y_i - \mathbf{a}_i^{\top}\mathbf{x}\|_2^2 = \frac{1}{2n}\sum_{i=1}^{n} d(\mathbf{a}_i^{\top}\mathbf{x}, \mathcal{Y}_i)^2\right]. \tag{3.62}$$

It is evident that primal stochastic mirror descent with constant step-size applied to the problem of least square given in Eq. (3.62) and the formulation of least square provided in Section 3.11.2 are equivalent, as is shown below.

Lemma 3.11.2.2 (**Anant Raj** and Bach [237]). *The mirror descent updates using the mirror map* ψ *for the least-squares problemprovided in Eq. (3.62) converges to minimum* ψ *solution.*

Accelerated Rates: One can also consider to use accelerated proximal randomized coordinate ascent [11, 88, 139] instead of non-accelerated randomized coordinate descent. For the problem, it leads to:

$$\mathbb{E}\Big[D_{\Psi}(\mathbf{x}^{\star}, \mathbf{x}(\boldsymbol{\alpha}^{(t)}))\Big] \leqslant \mathbb{E}\Big[\text{gap}(\boldsymbol{\alpha}^{(t)})\Big] \leqslant \frac{4\max_i L_i}{t^2}\left\{\frac{G(\boldsymbol{\alpha}^{\star}) - G(0)}{\max_i L_i} + \frac{1}{2}\|\boldsymbol{\alpha}^{\star}\|^2\right\}. \tag{3.63}$$

The above written bound in Eq. (3.63) is used in analyzing the general perceptron and details of which was provided in the main paper (Appendix).

As a direct implication of the result provided in Proposition 3.11.2.1, we have the convergence result for SDCA [216] and accelerated stochastic dual coordinate ascent [218] which is provided in Corollary 3.11.2.3 and Corollary 3.11.2.4. For the next two results, $\mathbf{x}_k$ is denoted as $\mathbf{x}(\boldsymbol{\alpha}_k)$.

Corollary 3.11.2.3 (Stochastic Dual Coordinate Ascent [237]). *Consider the regularized empirical risk minimization problem, then if SDCA [216] algorithm is run starting from*

$\boldsymbol{\alpha}_0 \in \mathbb{R}^n$ with a fix step size $1/\max_i L_i$ where $L_i = \frac{\|\mathbf{x}_i\|^2}{\lambda n^2}$, primal iterate after k iterations converges as following:

$$\frac{\lambda}{2}\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2 \leq D(\boldsymbol{\alpha}_{k+1}) \leq \left(1 - \frac{\gamma\lambda}{\max_i \|\mathbf{x}_i\|^2}\right)^k (\mathcal{S}_D(\boldsymbol{\alpha}_0) - \mathcal{S}_D(\boldsymbol{\alpha}^\star)).$$

Corollary 3.11.2.4 (Accelerated Stochastic Dual Coordinate Ascent [237])**.** *Consider the regularized empirical risk minimization problem, then if Accelerated SDCA [218] algorithm is run starting from $\boldsymbol{\alpha}_0 \in \mathbb{R}^n$, the following convergence rate for the primal iterates holds:*

$$\frac{\lambda}{2}\|\mathbf{x}_{k+1} - \mathbf{x}^\star\|^2 \leq D(\boldsymbol{\alpha}_{k+1}) \leq 2\left(1 - \frac{\sqrt{\gamma\lambda}}{\sqrt{\max_i \|\mathbf{x}_i\|^2}}\right)^k (\mathcal{S}_D(\boldsymbol{\alpha}_0) - \mathcal{S}_D(\boldsymbol{\alpha}^\star)).$$

Appendix A

Screening Rules for Convex Problems

Anant Raj[1], Jakob Olbrich[2], Bernd Gärtner[2], Bernhard Schölkopf[1], Martin Jaggi[3]

1 – MPI for Intelligent Systems, Tübingen

2 – ETH Zürich

3 – EPFL, Lausanne

Abstract

We propose a new framework for deriving screening rules for convex optimization problems. Our approach covers a large class of constrained and penalized optimization formulations, and works in two steps. First, given any approximate point, the structure of the objective function and the duality gap is used to gather information on the optimal solution. In the second step, this information is used to produce screening rules, i.e. safely identifying unimportant weight variables of the optimal solution. Our general framework leads to a large variety of useful existing as well as new screening rules for many applications. For example, we provide new screening rules for general simplex and L_1-constrained problems, Elastic Net, squared-loss Support Vector Machines, minimum enclosing ball, as well as structured norm regularized problems, such as group lasso.

A.1 Introduction

Optimization techniques for high-dimensional problems have become the work-horses for most data-analysis and machine-learning methods. With the rapid increase of available data, major challenges occur as the number of optimization variables (weights) grows beyond capacity of current systems.

The idea of screening refers to eliminating optimization variables that are guaranteed to *not* contribute to any optimal solution, and can therefore safely be removed from the problem. Such screening techniques have received increased interest in several machine learning related applications in recent years, and have been shown to lead to very significant computational efficiency improvements in various cases, in particular for many types of sparse methods. Screening techniques can be used either as a pre-processing before passing the problem to the optimizer, or also interactively during any iterative solver (called dynamic screening), to gradually reduce the problem complexity during optimization.

While existing screening methods were mainly relying on geometric and problem-specific properties, we in this paper take a different approach. We propose a new framework allowing screening on general convex optimization problems, using simple tools from convex duality instead of any geometric arguments. Our framework applies to a very large class of optimization problems both for constrained as well as penalized problems, including most machine learning methods of interest.

Our main contributions in this paper are summarized as follows:

1. We propose a new framework for screening for a more general class of optimization problem with a simple primal-dual structure.
2. The framework leads to a large set of new screening rules for machine learning problems that could not be screened before. Furthermore, it also recovers many existing screening rules as special cases.
3. We are able to express all screening rules using general optimization complexity notions such as smoothness or strong convexity, getting rid of problem-specific geometric properties.
4. Our proposed rules are dynamic (allowing any existing algorithm to be additionally equipped with screening) and safe (guaranteed to only eliminate truly unimportant variables).

Related Work. The concept of screening in the sense of eliminating non-influential data points to reduce the problem size has originated relatively independently in at least two communities. Coming from computational geometry, [3] has proposed a screening technique for the minimum enclosing ball problem for a given set of data points. Here screening can be interpreted as simply removing points which are guaranteed to lie in the strict interior of the final ball. Later [107] improve the threshold for this rule in the minimum enclosing ball setting.

Independently, the breakthrough work of [77] gave the first screening rules for the important case of sparse regression, as given in the Lasso. Since then, there have been many extensions and alterations of the general concept. While [77] exploits geometric quantities to bound the Lasso dual solution within a compact region, we recommend the survey paper by [264] for an overview of geometric methods for Lasso screening.

Sphere-region based methods differ from dome-shaped regions as used in [77] in choosing different centers and radii to bound the dual optimal point. Apart from being geometry specific, most existing approaches such as [77, 141, 182, 256, 257] are not agnostic to the regularization parameter used, but instead are restricted to perform screening along the entire regularization path (as the regularization parameter changes). This is known as sequential screening, and restricts its usability to optimization algorithms obtaining paths. In contrast, our proposed framework here allows any internal optimization algorithms to be equipped with screening.

Despite the importance of constrained problems in many applications, much less is known about screening for constrained optimization, in contrast to the case of penalized optimization problems. For the dual of the hinge loss SVM, which is a box-constrained optimization problem, [183] proposed a geometric screening rule based on the intersection region of two spheres, in the sequential setting of varying regularizer. More recently, [275] provided new screening rules for that case in the dynamic setting using a method similar to our approach. However their method is restricted to the SVM case.

As a first step to allow screening for more general optimization objectives, [66, 161, 162] have developed more systematic duality gap based screening rules for several problems, including group lasso, multi-task and multi-class problems (in the penalized setting) under a wider class of objectives f. While the earlier work of [66, 161] assumed separability of f over the group structure, later extensions of [162, 220], have generalized the applicability, but still rely on geometric and application-specific quantities in order to perform screening. The approach of [220] allows screening rules for (sparse) SVM problems on both dimensions, the features as well as the datapoints, but is limited in terms of generality of this specific sparse problem structure. We here provide screening rules for a more general framework of box constrained optimization, while hinge-loss SVM happens to be a special case of this. Our approach here is most similar to the Blitz framework of [100], which provides a general possibility to exploit piece-wise linear structure in an optimization problem in order to do screening. The method of [100] is however tied to a specific L1 algorithm, leading to very efficient active set methods on this smaller problem class. Further, exploitation of peicewise linearity is also discussed in paper [101] by selectively replacing piecewise terms in the objective with corresponding linear subfunctions which is easier to solve. Our proposed approach aims at capturing the largest possible general class of optimization problems allowing for screening. It can be shown to recover many of the other existing rules including e.g. [66, 161, 162] and [275], but significantly generalizing the method to general objectives and constraints as well as regularizers.

The rest of the paper is organized as follows: In Section A.2, we discuss our framework for screening. Section A.3 is devoted to deriving the information about optimal points in terms of gap functions. Sections A.4 and A.5 utilizes the framework and tools derived in previous sections to provide screening rules for the constrained and penalized case respectively. In the end, we provide a small illustrative experiment for screening on simplex and L_1-constrained and also discuss that which of the existing results can be

recovered using our algorithm in Section A.6.

A.2 Setup and Primal-Dual Structure

In this paper, we consider optimization problems of the following primal-dual structure. As we will see, the relationship between primal and dual objectives has many benefits, including computation of the duality gap, which allows us to have a certificate for approximation quality.

A very wide range of machine learning optimization problems can be formulated as (A) and (B), which are dual to each other:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \left[\; \mathcal{O}_A(\boldsymbol{x}) := \; f(A\boldsymbol{x}) \; + \; g(\boldsymbol{x}) \;\right] \tag{A}$$

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \left[\; \mathcal{O}_B(\boldsymbol{w}) := \; f^*(\boldsymbol{w}) \; + \; g^*(-A^\top \boldsymbol{w}) \;\right] \tag{B}$$

The two problems are associated to a given data matrix $A \in \mathbb{R}^{d\times n}$, and the functions $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ are allowed to be arbitrary closed convex functions. The functions f^*, g^* in formulation (B) are defined as the *convex conjugates* of their corresponding counterparts f, g in (A). Here $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{w} \in \mathbb{R}^d$ are the respective variable vectors. For a given function $h : \mathbb{R}^d \to \mathbb{R}$, its conjugate is defined as

$$h^*(\boldsymbol{v}) := \max_{\boldsymbol{u}\in\mathbb{R}^d} \boldsymbol{v}^\top \boldsymbol{u} - h(\boldsymbol{u})\,.$$

The association of problems (A) and (B) is a special case of Fenchel Duality. More precisely, the relationship is called *Fenchel-Rockafellar Duality* when incorporating the linear map A as in our case, see e.g. [29, Theorem 4.4.2] or [21, Proposition 15.18], see the Appendix A.8 for a self-contained derivation. The two main powerful features of this general duality structure are first that it includes many more machine learning methods than more traditional duality notions, and secondly that the two problems are fully symmetric, when changing respective roles of f and g. In typical machine learning problems, the two parts typically play the roles of a data-fit (or loss) term as well as a regularization term. As we will see later, the two roles can be swapped, depending on the application.

Optimality Conditions. The first-order optimality conditions for our pair of vectors $\boldsymbol{w} \in \mathbb{R}^d, \boldsymbol{x} \in \mathbb{R}^n$ in problems (A) and (B) are given as

$$\boldsymbol{w} \in \partial f(A\boldsymbol{x})\,, \tag{A.1a}$$

$$A\boldsymbol{x} \in \partial f^*(\boldsymbol{w})\,, \tag{A.1b}$$

$$-A^\top \boldsymbol{w} \in \partial g(\boldsymbol{x})\,, \tag{A.2a}$$

$$\boldsymbol{x} \in \partial g^*(-A^\top \boldsymbol{w}) \tag{A.2b}$$

see e.g. [21, Proposition 19.18]. The stated optimality conditions are equivalent to $\boldsymbol{x}, \boldsymbol{w}$

being a saddle-point of the Lagrangian, which is given as $\mathcal{L}(\boldsymbol{x}, \boldsymbol{w}) = f^*(\boldsymbol{w}) - \langle A\boldsymbol{x}, \boldsymbol{w}\rangle - g(\boldsymbol{x})$ if $\boldsymbol{x} \in \mathrm{dom}(g)$ and $\boldsymbol{w} \in \mathrm{dom}(f^*)$, see Appendix A.8 for details.

The Constrained Case. Any constrained convex optimization problem of the form

$$\min_{\boldsymbol{x}\in\mathcal{C}} f(A\boldsymbol{x}) \tag{A.3}$$

for a constraint set $\mathcal{C}$ can be directly written in the form (A) by using the indicator function of the constraint set as the penalization term g. (The indicator function $\boldsymbol{\iota}_{\mathcal{C}}$ of a set $\mathcal{C} \subset \mathbb{R}^n$ is defined as $\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) := 0$ if $\boldsymbol{x} \in \mathcal{C}$ and $\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) := +\infty$ otherwise.)

The Partially Separable Case. A very important special case arises when one part of the objective becomes separable. Formally, this is expressed as $g(\boldsymbol{x}) = \sum_{i=1}^n g_i(x_i)$ for univariate functions $g_i : \mathbb{R} \to \mathbb{R}$ for $i \in [n]$. Nicely in this case, the conjugate of g also separates as $g^*(\boldsymbol{y}) = \sum_i g_i^*(y_i)$. Therefore, the two optimization problems (A) and (B) write as

$$\mathcal{O}_A(\boldsymbol{x}) := f(A\boldsymbol{x}) + \textstyle\sum_i g_i(x_i) \tag{SA}$$
$$\mathcal{O}_B(\boldsymbol{w}) := f^*(\boldsymbol{w}) + \textstyle\sum_i g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w})\ , \tag{SB}$$

where $\boldsymbol{a}_i \in \mathbb{R}^d$ denotes the i-th column of A.

Crucially in this case, the optimality conditions (A.2a) and (A.2b) now become separable, that is

$$-\boldsymbol{a}_i^\top \boldsymbol{w} \in \partial g_i(x_i) \qquad \forall i\ . \tag{A.4a}$$
$$x_i \in \partial g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w}) \qquad \forall i\ . \tag{A.4b}$$

Note that the two other conditions (A.1a) and (A.1b) are unchanged in this case.

A.3 Duality Gap and Certificates

The duality gap for our problem structure provides an optimality certificate for our class of optimization problems. It will be the most important tool for us to provide guaranteed information about the optimal point (as in Section A.3.2), which will then be the foundation for the second step, to perform screening on the optimal point (as we will do in the later Sections A.4 and A.5).

A.3.1 Duality Gap Structure

For the problem structure (A) and (B) as given by Fenchel-Rockafellar duality, the *duality gap* for any pair of primal and dual variables $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{w} \in \mathbb{R}^d$ is defined as

$G(\boldsymbol{w},\boldsymbol{x}) := \mathcal{O}_A(\boldsymbol{x}) + \mathcal{O}_B(\boldsymbol{w})$. Non-negativity of the gap – that is weak duality – is satisfied by all pairs.

Most importantly, the duality gap acts as a certificate of approximation quality — the true optimum values $\mathcal{O}_A(\boldsymbol{x}^*)$ and $-\mathcal{O}_B(\boldsymbol{w}^*)$ (which are both unknown) will always lie within the (known) duality gap.

The Gap Function. For the special case of differentiable function f, we can study a simpler duality gap

$$G(\boldsymbol{x}) := \mathcal{O}_A(\boldsymbol{x}) + \mathcal{O}_B(\boldsymbol{w}(\boldsymbol{x})) \tag{A.5}$$

purely defined as a function of $\boldsymbol{x}$, using the optimality relation (A.1a), i.e. $\boldsymbol{w}(\boldsymbol{x}) := \nabla f(A\boldsymbol{x})$.

The Wolfe-Gap Function. For any constrained optimization problem (A.3) defined over a bounded set $\mathcal{C}$ and $\boldsymbol{x} \in \mathcal{C}$, the Wolfe gap function (also known as Hearn gap or Frank-Wolfe gap) is defined as the difference of f to the minimum of its linearization over the same domain. Formally,

$$GW(\boldsymbol{x}) := \max_{\boldsymbol{y}\in\mathcal{C}} (A\boldsymbol{x} - A\boldsymbol{y})^\top \nabla f(A\boldsymbol{x}). \tag{A.6}$$

It is not hard to see that the convenient Wolfe gap function is a special case of our above defined general duality gap $G(\boldsymbol{x}) := \mathcal{O}_A(\boldsymbol{x}) + \mathcal{O}_B(\boldsymbol{w}(\boldsymbol{x}))$, for g being the indicator function of the constraint set $\mathcal{C}$, and $\boldsymbol{w}(\boldsymbol{x}) := \nabla f(A\boldsymbol{x})$. For more details, see Appendix A.9.1, or also [122, Appendix D].

A.3.2 Obtaining Information about the Optimal Points

As we have mentioned, any type of screening will crucially rely on first deriving safe knowledge about the unknown optimal points of our given optimization problem. Here, we will use the duality gap to obtain such knowledge on the optimal points $\boldsymbol{x}^\star \in \mathbb{R}^n$ and $\boldsymbol{w}^\star \in \mathbb{R}^d$ of the respective optimization problems (A) and (B) respectively. Proofs are provided in Appendix A.9.2.

Our first lemma shows how to bound the distance between any (feasible) current dual iterate and the solution $\boldsymbol{w}^\star$ using standard assumptions on the objective functions.

Lemma A.3.2.1. *Consider the problem (B) with optimal solution $\boldsymbol{w}^\star \in \mathbb{R}^d$. For f being μ-smooth, we have*

$$\|\boldsymbol{w} - \boldsymbol{w}^\star\|^2 \leq \frac{2}{\mu}(f^*(\boldsymbol{w}) - f^*(\boldsymbol{w}^\star)) \tag{A.7}$$

The following corollary will be important to derive screening rules for penalized problems in Section A.5, as well as box-constrained problems (Section A.4.4).

Corollary A.3.2.2. *We consider the problem setup (A) and (B), and assume f is μ-smooth. Then*

$$\|\boldsymbol{w} - \boldsymbol{w}^\star\|^2 \leq \frac{2}{\mu} G(\boldsymbol{x}). \tag{A.8}$$

Here $G(\boldsymbol{x})$ is the duality gap function as defined in equation (A.5).

The following two results hold for general constrained optimization problems of the form (A.3), where g is the indicator function of a constraint set $\mathcal{C} \subset \mathbb{R}^n$ and hence are useful for deriving screening rules for such problems.

Lemma A.3.2.3. *Consider problem (A) and assume that f is μ-strongly convex over a bounded set $\mathcal{C}$. Then it holds that*

$$\|A\boldsymbol{x} - A\boldsymbol{x}^\star\|_2^2 \leq \frac{1}{\mu} GW(\boldsymbol{x}), \tag{A.9}$$

where $\boldsymbol{x}^\star$ is an optimal solution and GW is the Wolfe-Gap function of f over the bounded set $\mathcal{C}$.

Corollary A.3.2.4. *Assuming f is L-smooth as well as μ-strongly convex over a bounded set $\mathcal{C}$, we have*

$$\|\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}^\star)\| \leq \frac{L}{\sqrt{\mu}} \sqrt{GW(\boldsymbol{x})} \tag{A.10}$$

A.4 Screening Rules for Constrained Problems

In the following, we will develop screening rules for constrained optimization problems of the form (A.3), by exploiting the structure of the constraint set for a variety of sparsity-inducing problems. First of all, we give a general lemma which we will be using in rest of the paper to derive screening rules when any of the function in A and B is indicator function.

Lemma A.4.0.1. *For general constrained optimization $\min_{\boldsymbol{x}\in\mathcal{C}} f(A\boldsymbol{x})$, the optimality condition (A.2a) gives rise to the following optimality rule at the optimal point:*

$$(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star = \min_{\boldsymbol{z}\in\mathcal{C}} (A\boldsymbol{z})^\top \boldsymbol{w}^\star \tag{A.11}$$

The above equation (A.11) also suggest that $\boldsymbol{x}^\star = \arg\min_{\boldsymbol{z}\in\mathcal{C}} (A\boldsymbol{z})^\top \boldsymbol{w}^\star$. Lemma A.4.0.1 is very crucial in further deriving screening rules for constrained optimization problem as well as norm penalized problems whose conjugate is indicator function of the dual norm.

A.4.1 Simplex Constrained Problems

Optimization over unit simplex $\triangle := \{\boldsymbol{x} \in \mathbb{R}^n \mid x_i \geq 0, \quad \sum_{i=1}^n x_i = 1\}$ is a important class of constrained problems (A.3), as it includes optimization over any finite polytope.

In this case, the columns of A describe the vertices, and $\boldsymbol{x}$ are barycentric coordinates representing the point $A\boldsymbol{x}$. Formally, $g(\boldsymbol{x})$ is the indicator function of the unit simplex $\mathcal{C} = \triangle$ in this case.

The following two theorems provide screening rules for simplex constrained problems. We provide all proofs in Appendix A.10.1.

Theorem A.4.1.1. *For general simplex constrained optimization* $\min_{\boldsymbol{x}\in\triangle} f(A\boldsymbol{x})$*, the optimality condition* (A.2a) *gives rise to the following screening rule at the optimal point, for any* $i \in [n]$

$$(\boldsymbol{a}_i - A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star > 0 \;\Rightarrow\; x_i^\star = 0\,. \tag{A.12}$$

In the following Theorem A.4.1.2 we now assume smoothness and strong convexity of function f to provide screening rules for simplex problems, in terms of an arbitrary iterate $\boldsymbol{x}$, without knowing $\boldsymbol{x}^\star$.

Theorem A.4.1.2. *Let* f *be* L*-smooth and* μ*-strongly convex over the unit simplex* $\mathcal{C} = \triangle$*. Then for simplex constrained optimization* $\min_{\boldsymbol{x}\in\triangle} f(A\boldsymbol{x})$ *we have the following screening rule, for any* $i \in [n]$

$$(\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) > L\sqrt{\frac{GW(\boldsymbol{x})}{\mu}}\|\boldsymbol{a}_i - A\boldsymbol{x}\| \;\Rightarrow\; x_i^\star = 0\,.$$

Our general screening rules for simplex constrained problems as in Theorem A.4.1.2 allows many practical implications. For example, new screening rules for squared loss SVM and minimum enclosing ball problem come as a direct consequence.

Squared Hinge Loss SVM. The squared hinge-loss SVM problem in its dual form is formulated as

$$\min_{\boldsymbol{x}\in\triangle} \left[\, f(A\boldsymbol{x}) := \tfrac{1}{2}\boldsymbol{x}^\top A^\top A\boldsymbol{x} \,\right] \tag{A.13}$$

over a unit simplex constraint $\boldsymbol{x} \in \triangle \subset \mathbb{R}^n$. Here for given data examples $\bar{\boldsymbol{a}}_1, \ldots, \bar{\boldsymbol{a}}_n \in \mathbb{R}^d$ and corresponding labels $y_i \in \pm 1$, the matrix A collects the columns $\boldsymbol{a}_i = y_i\bar{\boldsymbol{a}}_i$, see e.g. [247]. We obtain the following novel screening rule for square loss SVM:

Corollary A.4.1.3. *For the squared hinge loss SVM* (A.13) *we have the screening rule*

$$\begin{aligned}(\boldsymbol{a}_i - A\boldsymbol{x})^\top A\boldsymbol{x} > \sqrt{\max_i\, (A\boldsymbol{x} - \boldsymbol{a}_i)^\top A\boldsymbol{x}}\,\|\boldsymbol{a}_i - A\boldsymbol{x}\| \\ \Rightarrow\; x_i^\star = 0.\end{aligned} \tag{A.14}$$

Minimum Enclosing Ball. The primal and dual for the minimum enclosing ball problem is given as the following pair of optimization formulations (A.15) and (A.16) respectively.

$$\min_{\boldsymbol{c}\in\mathbb{R}^d, r\in\mathbb{R}} \quad r^2 \quad \text{s.t.} \quad \|\boldsymbol{c}-\boldsymbol{a}_i\|_2^2 \leq r^2 \quad \forall i \in [n] \tag{A.15}$$

$$\min_{\boldsymbol{x}\in\triangle\subset\mathbb{R}^n} \quad \boldsymbol{x}^\top A^\top A\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x}\,, \tag{A.16}$$

where $\boldsymbol{c}$ is a vector whose i^{th} element c_i is $-\boldsymbol{a}_i^\top \boldsymbol{a}_i$, see for example [150] or our Appendix A.10.1. Given a set of n points, $\boldsymbol{a}_1$ to $\boldsymbol{a}_n$ in $\mathbb{R}^d$, the minimum enclosing ball is defined as the smallest ball $B_{\boldsymbol{c},r}$ with center $\boldsymbol{c}$ and radius r, i.e.: $B_{\boldsymbol{c},r} := \{\boldsymbol{x}\in\mathbb{R}^d \mid \|\boldsymbol{c}-\boldsymbol{x}\| \leq r\}$, such that all points $\boldsymbol{a}_i$ lie in its interior. In this set-up, screening means to identify points $\boldsymbol{a}_i$ lying in the interior of the optimal ball $B_{\boldsymbol{c}^\star,r^\star}$. Removing those points from the problem does not change the optimal ball.

Corollary A.4.1.4. *For the minimum enclosing ball problem (A.15) we have the screening rule*

$$(\boldsymbol{e}_i-\boldsymbol{x})^\top(2A^\top A\boldsymbol{x}+\boldsymbol{c}) > 2\sqrt{\tfrac{1}{2}\max_i (\boldsymbol{x}-\boldsymbol{e}_i)^\top(2A^\top A\boldsymbol{x}+\boldsymbol{c})}\,\|\boldsymbol{a}_i - A\boldsymbol{x}\| \Rightarrow\; x_i^\star = 0\,. \tag{A.17}$$

Our result improves upon the known rules by [3, 107] by providing a broader selection criterion (A.17).

A.4.2 L_1-Constrained Problems

L_1-constrained formulations are very widely used in order to induce sparsity in the variables. Here below we provide results for screening on general L_1-constrained problems, that is $\min_{\boldsymbol{x}\in\mathcal{C}} f(A\boldsymbol{x})$ for $\mathcal{C} = L_1 \subset \mathbb{R}^n$ (or a scaled version of the L_1-ball). Proofs are provided in Appendix A.10.2.

Theorem A.4.2.1. *For general L_1-constrained optimization $\min_{\boldsymbol{x}\in L_1} f(A\boldsymbol{x})$, the optimality condition (A.2a) gives rise to the following screening rule at the optimal point, for any $i \in [n]$*

$$\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| + (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star < 0 \;\Rightarrow\; x_i^\star = 0\,. \tag{A.18}$$

Using only a current iterate $\boldsymbol{x}$ instead of an optimal point, we obtain screening for general smooth and strongly convex function f:

Theorem A.4.2.2. *Let f be L-smooth and μ-strongly convex over the L_1-ball. Then for L_1-constrained optimization $\min_{\boldsymbol{x}\in L_1} f(A\boldsymbol{x})$ we have the following screening rule, for any*

$i \in [n]$

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) + L(\|\boldsymbol{a}_i\|_2 + \|A\boldsymbol{x}\|_2)\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} < 0$$
$$\Rightarrow \; \boldsymbol{x}_i^\star = 0 \tag{A.19}$$

A.4.3 Elastic Net Constrained Problems

Elastic net regularization as an alternative to L_1 is often used in practice, and can outperform the Lasso, while still enjoying a similar sparsity of representation [276]. The elastic net is given by the expression

$$\alpha\|\boldsymbol{x}\|_1 + (1-\alpha)\tfrac{1}{2}\|\boldsymbol{x}\|_2^2 \,.$$

Here below we provide novel result for screening on general elastic net constrained problems, that is $\min_{\boldsymbol{x}\in\mathcal{C}} f(A\boldsymbol{x})$ for $\mathcal{C}$ being the elastic net constraint, or a scaled version of it. Proofs are provided in Appendix A.10.3.

Theorem A.4.3.1. *For general elastic net constrained optimization* $\min_{\boldsymbol{x}\in L_E} f(A\boldsymbol{x})$ *where* $L_E := \{\boldsymbol{x} \in \mathbb{R}^n \mid \alpha\|\boldsymbol{x}\|_1 + \frac{(1-\alpha)}{2}\|\boldsymbol{x}\|_2^2 \leq 1\}$*, the optimality condition* (A.2a) *gives rise to the following screening rule at the optimal point, for any* $i \in [n]$

$$|\boldsymbol{a}_i^\top \boldsymbol{w}^\star| + (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star \left[\frac{\alpha}{1 + \frac{(1-\alpha)}{2}\|\boldsymbol{x}^\star\|_2^2}\right] < 0 \Rightarrow x_i^\star = 0$$

Using only a current iterate $\boldsymbol{x}$ instead of an optimal point, we obtain screening for general smooth and strongly convex function f:

Theorem A.4.3.2. *Let* f *be* L*-smooth and* μ*-strongly convex over the elastic net norm ball. Then for elastic net constrained optimization* $\min_{\boldsymbol{x}\in L_E} f(A\boldsymbol{x})$ *we have the following screening rule, for any* $i \in [n]$

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x})\left[\frac{2\alpha}{3-\alpha}\right]$$
$$+ L(\|\boldsymbol{a}_i\|_2 + \|A\boldsymbol{x}\|_2\left[\frac{2\alpha}{3-\alpha}\right])\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} < 0$$
$$\Rightarrow \; \boldsymbol{x}_i^\star = 0 \tag{A.20}$$

Note that both above results also recover the L_1 constrained case as a special case, when $\alpha \to 1$.

A.4.4 Screening for Box Constrained Problems

Box-constrained problems are important in several machine learning applications, including SVMs. After variable rescaling, w.l.o.g. we can assume the constraint set $\mathcal{C} = \square := \{\boldsymbol{x} \in \mathbb{R}^n \mid 0 \leq x_i \leq 1\}$. We derive screening rules for predicting both if a variable will take the upper or lower constraint.

Theorem A.4.4.1. *Let f be L-smooth. Then for box-constrained optimization $\min_{\boldsymbol{x}\in\square} f(A\boldsymbol{x})$, we obtain the following screening rules, for any $i \in [n]$*

$$\begin{aligned}
&\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} > 0 \;\Rightarrow\; x_i^\star = 0\,, \;\; \textit{and} \\
&\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) + \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} < 0 \;\Rightarrow\; x_i^\star = 1\,.
\end{aligned}$$

Box constrained opptimization problems arise very often in machine learning probelm. Hinge loss SVM happens to one of many special cases of box-constrained optimizaion problem.

Hinge Loss SVM. The dual of the classical support vector machine with hinge loss, when not using a bias value, is a box-constrained problem. As a direct consequence of Theorem A.4.4.1 we therefore obtain screening rules for SVM with hinge loss and no bias. The primal formulation of the SVM in this setting, for a regularization parameter $C > 0$, is

$$\begin{aligned}
&\min_{\boldsymbol{w}\in\mathbb{R}^d, \boldsymbol{\varepsilon}\in\mathbb{R}^n} \tfrac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\mathbf{1}^\top \boldsymbol{\varepsilon} \\
&\text{s.t.} \quad \boldsymbol{w}^\top \boldsymbol{a}_i \geq 1 - \varepsilon_i \;\; \forall i \in [n] \\
&\varepsilon_i \geq 0 \quad \forall i \in [n]
\end{aligned} \tag{A.21}$$

Corollary A.4.4.2. *For SVM with hinge loss and no bias as given in (A.21), we have the screening rules*

$$\begin{aligned}
&\boldsymbol{a}_i^\top A\boldsymbol{x} - \|\boldsymbol{a}_i\|_2 \sqrt{2G(\boldsymbol{x})} > 0 \;\Rightarrow\; x_i^\star = 0\,, \;\; \textit{and} \\
&\boldsymbol{a}_i^\top A\boldsymbol{x} + \|\boldsymbol{a}_i\|_2 \sqrt{2G(\boldsymbol{x})} < 0 \;\Rightarrow\; x_i^\star = C\,.
\end{aligned}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is any feasible dual point.

We get similar screening rules for hinge loss SVM as in [275] as well as in [220]. The closest known result to our Corollary A.4.4.2 for screening in hinge loss SVM is given in [275] and [220]. The work of [275] also covers the kernelized SVM case, and improves the threshold given in our Corollary A.4.4.2 by a constant of $\sqrt{2}$. In Appendix A.10.4, we show that our more general approach here can also be adjusted to gain this constant factor.

A.5 Screening for Penalized Problems

In this section we will develop screening methods for general penalized convex optimization problems of the form (A) and (B). The cornerstone application are L_1 regularized problems, for which we now develop screening rules with general cost function f. We show in Appendix A.11.1 that our method can reproduce the screening rules of [161] as special cases, whereas their method does not directly extend to general f. Beyond L_1 problems, we also describe new screening rules for elastic net regularized problems, as well as the important case of structured norm regularized optimization.

A.5.1 L_1-Penalized Problems

The next theorem describes a screening rule for general L_1-penalized problems, under a smoothness assumption on function f. Proofs for are given in Appendix A.11.1.

Theorem A.5.1.1. *Consider an L_1-regularized optimization problem of the form*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; f(A\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1 \tag{A.22}$$

If f is L-smooth, then the following screening rule holds for all $i \in [n]$:

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| < \lambda - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$$

By careful observation of the expression in Theorem A.5.1.1, it is easy to find a connection between our screening rule and the geometric sphere test method based screening [264]. The general idea behind the sphere test is to consider the maximum value of the objective function in a spherical region which contains the optimal dual variable. We discuss this connection in more detail in section A.5.4.

Here, we also discuss the special cases of squared loss regression and logistic loss regression with L_1 penalization. These results are presented in Corollaries A.5.1.2 and A.5.1.3 as direct consequences of Theorem A.5.1.1. Both of the corollaries can also be derived from the framework discussed in the paper [161] and produce similar screening rules. Proof of both of the corollaries are discussed in Appendix A.11.1.

Corollary A.5.1.2. *Consider an optimization problem of the form:*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; \tfrac{1}{2}\|A\boldsymbol{x} - b\|_2^2 + \lambda \|\boldsymbol{x}\|_1$$

Then the screening rule is given by:
$\left|\boldsymbol{a}_i^\top (A\boldsymbol{x} - \boldsymbol{b})\right| < \lambda - \|\boldsymbol{a}_i\|_2 \sqrt{2G(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0.$

In the Appendix A.11.1, we also discuss how to improve the screening threshold for squared loss penalized regression by a constant factor.

Corollary A.5.1.3. *The optimization problem for logistic regression with L_1 regularizer can be written in the form of:*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \sum_{i=1}^{n} \log(\exp([A\boldsymbol{x}]_i)+1) + \lambda\|\boldsymbol{x}\|_1 \tag{A.23}$$

And screening rule for above problem can be written as :

$$\left|\boldsymbol{a}_i^\top \left(\frac{\exp(A\boldsymbol{x})}{\exp(A\boldsymbol{x})+1}\right)\right| < \lambda - \|\boldsymbol{a}_i\|_2\sqrt{2G(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$$

where $\left(\frac{\exp(A\boldsymbol{x})}{\exp(A\boldsymbol{x})+1}\right)$ *is element wise vector whose* i_{th} *element is* $\left(\frac{\exp([A\boldsymbol{x}]_i)}{\exp([A\boldsymbol{x}]_i)+1}\right)$

A.5.2 Elastic-Net Penalized Problems

In the next corollary, we present a novel screening rule for the elastic net squared loss regression problem.

Corollary A.5.2.1. *Consider the elastic net regression formulation*

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \tfrac{1}{2}\|A\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \lambda_2\|\boldsymbol{x}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 \tag{A.24}$$

The following screening rule holds for all $i \in [n]$:

$$\left|(\boldsymbol{a}_i^\top A + 2\lambda_2\boldsymbol{e}_i^\top)\boldsymbol{x} - \boldsymbol{a}_i^\top\boldsymbol{b}\right| < \lambda_1 - \sqrt{2(\boldsymbol{a}_i^\top\boldsymbol{a}_i + 2\lambda_2)G(\boldsymbol{x})} \;\Rightarrow\; x_i^\star = 0.$$

We also recover existing screening rules for elastic net regularized problem with more general objective f using our framework,

Theorem A.5.2.2. *If we consider the general elastic net formulation of the form*

$$\min_{\boldsymbol{x}} f(A\boldsymbol{x}) + (1-\alpha)\tfrac{1}{2}\|\boldsymbol{x}\|_2^2 + \alpha\|\boldsymbol{x}\|_1 \tag{A.25}$$

If f is L-smooth, then the following screening rule holds for all $i \in [n]$:

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| < \alpha - \|\boldsymbol{a}_i\|_2\sqrt{2LG(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0\,.$$

This rule has been derived earlier in [220], but can also be seen as a special case of our framework, see Appendix A.11.1. In the proof, we derive screening rules from both the formulation (B.17) and (SB) using optimality condition (A.4a) and (A.4b) which is novel as well as help us to understand the property useful in deriving screening rules for elastic net penalized problems.

A.5.3 Structured Norm Penalized Problems

Here in this section we present screening rules for non-overlapping group norm regularized problems. Group-norm regularization is widely used to induce sparsity in terms of groups of variables of the the solution of the optimization problem. The most prominent example is the group lasso (ℓ_2/ℓ_1-regularization). Here in this section we mostly discuss screening for general objectives with an ℓ_2/ℓ_1-regularization. Proofs are provided in Appendix A.11.2.

Group Norm - ℓ_2/ℓ_1 Regularization. In the following, we use the notation $\{\boldsymbol{x}_1 \cdots \boldsymbol{x}_G\}$ to express a vector $\boldsymbol{x}$ as a partition of non-overlapping groups $g \in \mathcal{G}$ of variables, such that $\boldsymbol{x}^\top = \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top \cdots \boldsymbol{x}_G^\top\right]$. Correspondingly, the matrix A can be denoted as the concatenation of the respective column groups $A = [A_1\, A_2 \cdots A_G]$, and $\sum_{g\in\mathcal{G}} |g| = n$.

Theorem A.5.3.1. *For ℓ_2/ℓ_1-regularized optimization problem of the form*

$$\min_{\boldsymbol{x}} f(A\boldsymbol{x}) + \sum_{g=1}^{G} \sqrt{\rho_g}\|\boldsymbol{x}_g\|_2$$

Assuming f is L-smooth, then the following (group-level) screening rule holds for all groups g:

$$\|A_g^\top \nabla f(A\boldsymbol{x})\|_2 + \sqrt{2L}\left\|A_g\right\|_{Fro} < \sqrt{\rho_g} \;\Rightarrow\; \boldsymbol{x}_g^\star = \boldsymbol{0} \in \mathbb{R}^{|g|}\,.$$

Corollary A.5.3.2. ***Group Lasso Regression with Squared Loss*** *- For the group lasso formulation*

$$\min_{\boldsymbol{x}} \tfrac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{\rho_g}\|\boldsymbol{x}_g\|_2$$

we have the following screening rule for all groups g:

$$\|A_g^\top (A\boldsymbol{x} - \boldsymbol{b})\|_2 + \sqrt{2G(\boldsymbol{x})}\left\|A_g\right\|_{Fro} < \lambda\sqrt{\rho_g} \;\Rightarrow\; \boldsymbol{x}_g^\star = \boldsymbol{0}\,.$$

Group lasso regression is widely used in applications as a working example case of structured norm penalization. For the squared-loss special case, group lasso screening rules were recently developed by [162]. Similarly, [130] is also restricted to least-squares f objective.

A.5.4 Connection with Sphere Test Method

The general idea behind the sphere test method [264] is to consider the maximum value of desired function in a spherical region which contains the optimal dual variable. In

context of our general framework (A) and (B), we obtain this case when considering an ℓ_1 penalty or ℓ_2/ℓ_1 penalty. That means g is a norm and hence from Lemma A.9.2.1, g^* becomes the indicator function of the dual norm ball of $A^\top \boldsymbol{w}$. The dual norm function for ℓ_1 norm is of the form $\max_i |\boldsymbol{a}_i^\top \boldsymbol{w}|$ and for ℓ_2/ℓ_1 norm, it is $\max_g \|A_g^\top \boldsymbol{w}\|$. Hence, we try to find maximum value of the function of the forms $\max_{\boldsymbol{\theta}\in\mathcal{S}(\boldsymbol{q},r)} \boldsymbol{a}_i^\top \boldsymbol{\theta}$ where $\mathcal{S}(\boldsymbol{q},r) = \{z : \|\boldsymbol{z}-\boldsymbol{q}\|_2 \leq r\}$ the ball $\mathcal{S}$ also contains the optimal dual point $\boldsymbol{w}^\star$. If the maximum value of $\boldsymbol{a}_i^\top \boldsymbol{\theta}$ is less than some particular value for all the $\boldsymbol{\theta}$ in the ball hence $\boldsymbol{a}_i^\top \boldsymbol{w}$ will also be less than that particular value and that is the main reason we try to find maximum of $\boldsymbol{a}_i^\top \boldsymbol{\theta}$ over the ball $\mathcal{S}$.

$$\begin{aligned}\max_{\boldsymbol{\theta}\in\mathcal{S}(\boldsymbol{q},r)} \boldsymbol{a}_i^\top \boldsymbol{\theta} &= \boldsymbol{a}_i^\top(\boldsymbol{\theta}-\boldsymbol{q}+\boldsymbol{q}) = \boldsymbol{a}_i^\top(\boldsymbol{\theta}-\boldsymbol{q}) + \boldsymbol{a}_i^\top \boldsymbol{q} \\ &\leq \|\boldsymbol{a}_i\|_2 \|\boldsymbol{\theta}-\boldsymbol{q}\| + \boldsymbol{a}_i^\top \boldsymbol{q} \leq r\|\boldsymbol{a}_i\|_2 + \boldsymbol{a}_i^\top \boldsymbol{q}\end{aligned}$$

Similar arguments can be given in the ℓ_2/ℓ_1-norm case. A variety of existing screening test for lasso and group lasso are of this flavor of sphere tests. The difference between these approaches mainly lie in the way of choosing the center and bounding the radius of the sphere, such that the optimal dual variables lie inside the sphere. Our method can be seen as a general framework for such a sphere test based screening with dynamic screening rules. Our method can be interpreted as a sphere test with the current iterate of the dual variable $\boldsymbol{w}$ as a center of the ball, and we obtain the bound on the radius in terms of duality gap function.

A.6 Illustrative Experiments

While the contribution of our paper is on the theoretical generality and the collection of new screening applications, we will still briefly illustrate the performance of some of the proposed screening algorithms, for the classical examples of simplex constrained and L_1-constrained problems. We compare the fraction of active variables and the Wolfe-Gap function as optimization algorithm progress.

We consider the optimization problem of the form $\min_{\boldsymbol{x}\in\mathcal{B}_{L_1}} \|A\boldsymbol{x}-\boldsymbol{b}\|_2^2$. $\mathcal{B}_{L_1}$ is a scaled L_1-ball with radius 35. $A \in \mathbb{R}^{3000\times 600}$ is a random Gaussian matrix and a noisy measurement $\boldsymbol{b} = A\boldsymbol{x}^\star$ where $\boldsymbol{x}^\star$ is a sparse vector of $+1$ and -1 with only 70 non zeros entries. We solve the above optimization problem using the Frank-Wolfe algorithm (pair-wise variant, see [119]). Before putting this optimization problem into the solver we convert this problem into the barycentric representation which is $\min_{\boldsymbol{x}_\triangle\in\triangle} \|A_\triangle \boldsymbol{x}_\triangle - \boldsymbol{b}\|_2^2$. The relation between the transformed variable and original variable can be given by $A_\triangle = [A\,|\,-A]$ and $\boldsymbol{x} = [I_n\,|\,-I_n]\boldsymbol{x}_\triangle$. For more details see [97].

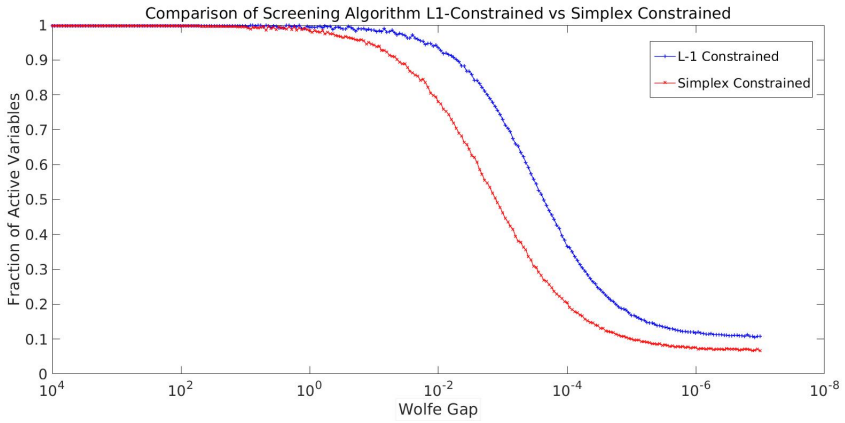


Figure A.1: Simplex- vs L_1-constrained Screening

Dataset/ No. of Samples	**No Screening (Simplex)**	**Screening (Simplex)**
Synth1 5000	13.1 sec	11.7sec
Synth2 10000	28.3 sec	23.1 sec
RCV1 20242	18.6 min	13.5 min
news20B 19996	33.4 min	25.2 min

Table A.1: Simplex-constrained screening, clock time

Dataset/ No. of Samples	**No Screening (ℓ_1-constr.)**	**Screening (ℓ_1-constr.)**
Synth1 5000	13.1	12.2 sec
Synth2 10000	28.3 sec	24.7 sec
RCV1 20242	18.6 min	14.9 min
news20B 19996	33.4 min	27.1 min

Table A.2: L_1-constrained screening, clock time

Now we apply our Theorems A.4.2.2 and A.4.1.2 on variable of $\boldsymbol{x}$ and $\boldsymbol{x}_\triangle$ respectively to screen, in order to compare the two alternative screening approaches on the same problem. Note that the Wolfe gap is identical in both parameterizations, for any $\boldsymbol{x}$. One important point to note here is that dimension of $\boldsymbol{x}_\triangle$ is the double of the dimension of $\boldsymbol{x}$, and any L_1-coordinate value x_i is zero if and only if both "duplicate" variables $x_{\triangle,i}$ and $x_{\triangle,n+i}$ are zero, where n is the dimensionality of $\boldsymbol{x}$.

Therefore, the simplex variant (with more variables) performs a more fine-grained variant of screening, where we can screen each of the sign patterns separately for each variable. In Fig A.1, the blue curve illustrates the screening efficiency for the L_1-constrained

screening case, while the red curve illustrate simplex constrained screening. Our theorems A.4.2.2 and A.4.1.2 are well in line with the phenomena in Fig A.1. For the L_1-constrained case, the screening starts relatively at later stage than simplex case due to the fact that in Equation (A.19), two out of three terms are absolute values of some quantity and hence it is very tough to compensate both of them by the third quantity, in order for the entire sum to become negative. Hence in the beginning this rule can often be ineffective. As algorithm progresses, the duality gap becomes smaller and screening starts but at the same time the gradient (and therefore gap) also starts to decay which brings the trade-off shown in the plot. For both variants, screening becomes slow towards the end.

We also report the time taken to reach a duality gap of 10^{-7} with both the approaches mentioned above (simplex constrained and L_1-constrained) on for different datasets. The first two datasets (*Synth1* and *Synth2*) are generated under the same setting described earlier but *Synth1* with 5000 samples and *Synth2* with 10000 samples. *RCV1* is a real world dataset having $20,242$ samples and $47,236$ data dimensions. *news20Binary* is also a real world dataset having $19,996$ entries and $1,355,191$ dimensions. Below in Tables A.1 and A.2, we describe the running time of the optimization methods to reach a duality gap threshold of 10^{-7} with or without screening. On *RCV1* dataset we try the feature learning with L_1-norm ball constraint of 200 and on *news20Binary* we use L_1-norm ball constraint of 35. In the case of *RCV1* and *news20Binary*, A is the data matrix and $\boldsymbol{b}$ is the label of each instance in the dataset. From Tables A.1 and A.2 it is also evident that simplex screening rule is more tighter than the L_1-constrained screening rule.

A.7 Discussion

We have presented a unified way to derive screening rules for general constrained and penalized optimization problems. For both cases, our framework crucially utilizes the structure of piece-wise linearity of the problem at hand. For the constrained case, we showed that screening rules follow from the piece-wise linearity of the boundary of the constraint set.

The crucial property is that at non-differentiable boundary points, the normal cone – i.e. the sub-differential of the indicator function of the constraint set – becomes a relatively large set. Under moderate assumptions on the objective function, we are able to guarantee that also the gradient of an optimal point must lie in this same cone region, leading to screening.

On the other hand for penalized optimization problems, we are able to derive screening rules from either piece-wise linearity of the penalty function, or as well from exploiting piece-wise linearity of the constraint set arising from the dual (conjugate) of the penalty function.

Proofs for Main Results

A.8 Primal Dual Structure (Proofs for Section A.2)

The relation of our primal and dual problems (A) and (B) is standard in convex analysis, and is a special case of the concept of Fenchel Duality. Using the combination with the linear map A as in our case, the relationship is called *Fenchel-Rockafellar Duality*, see e.g. [29, Theorem 4.4.2] or [21, Proposition 15.18]. For completeness, we here illustrate this correspondence with a self-contained derivation of the duality.

Proof. Starting with the original formulation (A), we introduce a helper variable vector $\boldsymbol{v} \in \mathbb{R}^d$ representing $\boldsymbol{v} = A\boldsymbol{\alpha}$. Then optimization problem (A) becomes:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad f(\boldsymbol{v}) + g(\boldsymbol{\alpha}) \quad \text{such that } \boldsymbol{v} = A\boldsymbol{\alpha}\,. \tag{A.26}$$

Introducing Lagrange multipliers $\boldsymbol{w} \in \mathbb{R}^d$, the Lagrangian is given by:

$$L(\boldsymbol{\alpha}, \boldsymbol{v}; \boldsymbol{w}) := f(\boldsymbol{v}) + g(\boldsymbol{\alpha}) + \boldsymbol{w}^\top (A\boldsymbol{\alpha} - \boldsymbol{v})\,.$$

The dual problem of (A) follows by taking the infimum with respect to both $\boldsymbol{\alpha}$ and $\boldsymbol{v}$:

$$\begin{aligned}
\inf_{\boldsymbol{\alpha},\boldsymbol{v}} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{v}) &= \inf_{\boldsymbol{v}} \left\{ f(\boldsymbol{v}) - \boldsymbol{w}^\top \boldsymbol{v} \right\} + \inf_{\boldsymbol{\alpha}} \left\{ g(\boldsymbol{\alpha}) + \boldsymbol{w}^\top A\boldsymbol{\alpha} \right\} \\
&= -\sup_{\boldsymbol{v}} \left\{ \boldsymbol{w}^\top \boldsymbol{v} - f(\boldsymbol{v}) \right\} - \sup_{\boldsymbol{\alpha}} \left\{ (-\boldsymbol{w}^\top A)\boldsymbol{\alpha} - g(\boldsymbol{\alpha}) \right\} && \text{(A.27)} \\
&= -f^*(\boldsymbol{w}) - g^*(-A^\top \boldsymbol{w})\,. && \text{(A.28)}
\end{aligned}$$

We change signs and turn the maximization of the dual problem (A.28) into a minimization and thus we arrive at the dual formulation (B) as claimed:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad \Big[\; \mathcal{O}_B(\boldsymbol{w}) := f^*(\boldsymbol{w}) + g^*(-A^\top \boldsymbol{w}) \;\Big]\,.$$

The Partially Separable Case. For $g(\boldsymbol{x})$ is separable, i.e. $g(\boldsymbol{x}) = \sum_{i=1}^n g_i(x_i)$ for univariate functions $g_i : \mathbb{R} \to \mathbb{R}$ for $i \in [n]$, the primal-dual structure remains the separable. In this case, the conjugate of g also separates as $g^*(\boldsymbol{y}) = \sum_i g_i^*(y_i)$. Therefore, in terms of the the primal-dual structure (A) and (B) we obtain the separable special case (B.17) and (SB). □

Optimality Conditions. The first-order optimality conditions follow from the standard definition of the conjugate functions in the Fenchel dual problem, see also e.g. [21, 29].

Proof. The first-order optimality conditions for our pair of vectors $\boldsymbol{w} \in \mathbb{R}^d, \boldsymbol{x} \in \mathbb{R}^n$ in problems (A) and (B) are given by equations (A.1a), (A.2a), (A.1b) and (A.2b). The proof directly comes from equation (A.27) by separately writing optimizing conditions for two expressions $\boldsymbol{w}^\top \boldsymbol{v} - f(\boldsymbol{v})$ and $(-\boldsymbol{w}^\top A)\boldsymbol{\alpha} - g(\boldsymbol{\alpha})$ in equation (A.27).

Crucially in the partially separable case, the optimality conditions (A.2a) and (A.2b) become separable. Comparing the expressions (B.17) and (A), we see that $g(\boldsymbol{x}) = \sum_i g_i(x_i)$ and hence

$$g^*(\boldsymbol{x}) = \sum_i g_i^*(x_i)$$

Hence by applying (A.2a) and (A.2b) we obtain the separable optimality conditions (A.4a) and (A.4b). □

A.9 Duality Gap and Objective Function Properties

A.9.1 Wolfe Gap as a Special Case of Duality Gap

Proof. To see this as a special case of general duality gap of the problem formulation, we consider the constraint as indicator function of set $\mathcal{C}$ such that $g(\boldsymbol{x}) = \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x})$. Now from the definition of the Wolfe gap function

$$GW(\boldsymbol{x}) := \max_{\boldsymbol{y} \in \mathcal{C}} (A\boldsymbol{x} - A\boldsymbol{y})^\top \partial f(A\boldsymbol{x})$$

Here $\partial f(A\boldsymbol{x})$ is an arbitrary subgradient of f at the candidate position $\boldsymbol{x}$, and $\boldsymbol{\iota}_{\mathcal{C}}^*(\boldsymbol{y}) := \sup_{\boldsymbol{s} \in \mathcal{C}} \langle \boldsymbol{s}, \boldsymbol{y} \rangle$ is the support function of $\mathcal{C}$. Now writing the general duality gap $G(\boldsymbol{x})$ as

$$\begin{aligned} G(\boldsymbol{x}) &:= \mathcal{O}_A(\boldsymbol{x}) + \mathcal{O}_B(\boldsymbol{w}(\boldsymbol{x})) \\ &:= f(A\boldsymbol{x}) + \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) + f^*(\boldsymbol{w}(\boldsymbol{x})) + \boldsymbol{\iota}_{\mathcal{C}}^*(-(A^\top \boldsymbol{w}(\boldsymbol{x}))) \end{aligned}$$

the last term disappears since we assumed $\boldsymbol{x} \in \mathcal{C}$. Using the definition of the Fenchel conjugate, one has the Fenchel-Young inequality, i.e.

$$f^*(\boldsymbol{w}) := \max_{\boldsymbol{u} \in \mathbb{R}^d} \boldsymbol{w}^\top \boldsymbol{u} - f(\boldsymbol{u}) \;\Rightarrow\; f^*(\boldsymbol{w}) + f(\boldsymbol{u}) \geq \boldsymbol{w}^\top \boldsymbol{u}$$

The above holds with equality if $\boldsymbol{w}$ is chosen as a subgradient of f at $\boldsymbol{u} = A\boldsymbol{x}$. Therefore, using our first-order optimality mapping $\boldsymbol{w}(\boldsymbol{x}) := \partial f(A\boldsymbol{x})$, we have

$$G(\boldsymbol{x}) = (A\boldsymbol{x})^\top \partial f(A\boldsymbol{x}) + \boldsymbol{\iota}_{\mathcal{C}}^*(-(A^\top \boldsymbol{w}(\boldsymbol{x}))) = GW(\boldsymbol{x})$$

This derivation is adapted from [122, Appendix D]. □

A.9.2 Obtaining Information about the Optimal Points

Lemma A.9.2.1 (Conjugates of Indicator Functions and Norms). *i) The conjugate of the indicator function $\iota_{\mathcal{C}}$ of a set $\mathcal{C} \subset \mathbb{R}^n$ (not necessarily convex) is the support function of the set $\mathcal{C}$, that is* $\iota^*_{\mathcal{C}}(\boldsymbol{x}) = \sup_{\boldsymbol{s}\in\mathcal{C}} \langle \boldsymbol{s}, \boldsymbol{x}\rangle$

ii) The conjugate of a norm is the indicator function of the unit ball of the dual norm.

Proof. [31, Example 3.24 and 3.26] □

Lemma A.9.2.2. *Assume that f is a closed and convex function then f^* is μ-strongly convex with respect to a norm $\|\cdot\|$ if and only if f is $1/\mu$-Lipschitz gradient with respect to dual norm $\|\cdot\|_*$.*

Proof. [106, Theorem 3] □

Proof of Lemma A.3.2.1. From the definition of μ-strongly convex function, we know that

$$\begin{aligned} f^*(\boldsymbol{w}) &\geq f^*(\boldsymbol{w}^\star) + (\boldsymbol{w}-\boldsymbol{w}^\star)^\top \nabla f^*(\boldsymbol{w}^\star) + \frac{\mu}{2}\|\boldsymbol{w}-\boldsymbol{w}^\star\|_2^2 \\ &\geq f^*(\boldsymbol{w}^\star) + \frac{\mu}{2}\|\boldsymbol{w}-\boldsymbol{w}^\star\|_2^2 \end{aligned}$$

The first inequality follows directly by using the first order optimality condition for $\boldsymbol{w}^\star$ being optimal. For any optimal point $\boldsymbol{w}^\star$ and another feasible point $\boldsymbol{w}$,

$$(\boldsymbol{w}-\boldsymbol{w}^\star)^\top \nabla f^*(\boldsymbol{w}^\star) \geq 0.$$

Hence, $\|\boldsymbol{w}^\star - \boldsymbol{w}\|_2^2 \leq \frac{2}{\mu}(f^*(\boldsymbol{w}) - f^*(\boldsymbol{w}^\star))$ □

Proof of Corollary A.3.2.2. This statement directly comes from (A.3.2.1) and the definition of the duality gap. By definition we know that the true optimum values $\mathcal{O}_A(\boldsymbol{x}^*)$ and $-\mathcal{O}_B(\boldsymbol{w}^*)$ respectively for primal (A) and dual formulation (B) will always lie within the duality gap which implies

$$G(\boldsymbol{x}) \geq \mathcal{O}_B(\boldsymbol{w}) - \mathcal{O}_B(\boldsymbol{w}^\star)$$

By equation (B), we know that $\mathcal{O}_B(\boldsymbol{w}) = f^*(\boldsymbol{w}) + g^*(-A^\top \boldsymbol{w}^\star)$
Now since f^* is μ-strongly convex function and g^* is convex hence,

$$f^*(\boldsymbol{w}) \geq f^*(\boldsymbol{w}^\star) + \nabla f^*(\boldsymbol{w}^\star)^\top(\boldsymbol{w}-\boldsymbol{w}^\star) + \frac{\mu}{2}\|\boldsymbol{w}-\boldsymbol{w}^\star\|_2^2 \tag{A.29}$$

$$g^*(-A^\top \boldsymbol{w}) \geq g^*(-A^\top \boldsymbol{w}^\star) + \nabla g^*(-A^\top \boldsymbol{w}^\star)^\top(-A^\top \boldsymbol{w} + A^\top \boldsymbol{w}^\star) \tag{A.30}$$

Hence by adding equation (A.29) and (A.30), we get

$$\mathcal{O}_B(\boldsymbol{w}) \geq \mathcal{O}_B(\boldsymbol{w}^\star) + (\nabla f^*(\boldsymbol{w}^\star) - A\nabla g^*(-A^\top \boldsymbol{w}^\star))^\top(\boldsymbol{w}-\boldsymbol{w}^\star) + \frac{\mu}{2}\|\boldsymbol{w}-\boldsymbol{w}^\star\|_2^2$$

$$\Rightarrow \mathcal{O}_B(\boldsymbol{w}) \geq \mathcal{O}_B(\boldsymbol{w}^\star) + \nabla \mathcal{O}_B(\boldsymbol{w}^\star)^\top (\boldsymbol{w} - \boldsymbol{w}^\star) + \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{w}^\star\|_2^2$$

At optimal point $\boldsymbol{w}^\star$, $\nabla \mathcal{O}_B(\boldsymbol{w}^\star)^\top (\boldsymbol{w} - \boldsymbol{w}^\star) \geq 0$.
Hence,

$$G(\boldsymbol{x}) \geq \mathcal{O}_B(\boldsymbol{w}) - \mathcal{O}_B(\boldsymbol{w}^\star) \geq \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{w}^\star\|_2^2$$

□

Proof of Lemma A.3.2.3. From the definition of μ-strong convexity of f and using optimality condition,

$$\begin{aligned} \mu\|A\boldsymbol{x} - A\boldsymbol{x}^\star\|^2 &\leq (A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top (\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}^\star)) && \text{(A.31)} \\ &\leq (A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}) && \text{(A.32)} \\ &\leq GW(\boldsymbol{x}) && \text{(A.33)} \end{aligned}$$

Equation (A.31) comes from the definition of μ-strong convexity.
Equation (A.32) is first order optimality condition for $\boldsymbol{x}^\star$ being optimal which implies

$$(A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) \geq 0$$

The inequality (A.33) follows by the definition of the gap function given in (A.6). □

Proof of Corollary A.3.2.4. This comes by definition of L-smooth functions and Lemma A.3.2.3. From the definition,

$$\begin{aligned} \|\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}^\star)\| &\leq L\|A\boldsymbol{x} - A\boldsymbol{x}^\star\| \\ &\leq \frac{L}{\sqrt{\mu}}\sqrt{GW(\boldsymbol{x})} \end{aligned}$$

Second inequality directly comes from Lemma A.3.2.3. □

A.10 Screening on Constrained Problems

Lemma A.10.0.1. *Let $\mathcal{C}$ be a convex set, and $\boldsymbol{\iota}_{\mathcal{C}}$ be its indicator function, then*

1. *For $\boldsymbol{x} \notin \mathcal{C}$, $\partial \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) = \emptyset$*
2. *For $\boldsymbol{x} \in \mathcal{C}$, we have that $\boldsymbol{w} \in \partial \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x})$ if $\boldsymbol{w}^\top (\boldsymbol{z} - \boldsymbol{x}) \leq 0 \quad \forall \boldsymbol{z} \in \mathcal{C}$*

Proof. Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a closed convex set. Then subgradient of indicator function $\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x})$ at $\boldsymbol{x}$ will be vectors $\boldsymbol{u}$ which satisfy

$$\begin{aligned} &\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) \geq \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) + \boldsymbol{u}^\top (\boldsymbol{z} - \boldsymbol{x}) \quad \forall \boldsymbol{z} \in \text{dom}(\boldsymbol{\iota}_{\mathcal{C}}) \\ &\Rightarrow \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) \geq \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) + \boldsymbol{u}^\top (\boldsymbol{z} - \boldsymbol{x}) \quad \forall \boldsymbol{z} \in \mathbb{R}^n \end{aligned} \tag{A.34}$$

If $\text{int}(\mathcal{C})$ represents the interior of the set $\mathcal{C}$ such that it contains n-dimensional ball of radius $r > 0$, and $Bd(\mathcal{C})$ represents boundary of the set $\mathcal{C}$. Now we have to assume various cases for proving Lemma A.10.0.1.

Case 1 We evaluate Equation (A.34) when $\boldsymbol{x} \in \text{int}(\mathcal{C})$. Equation (A.34) becomes

$$\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) \geq \boldsymbol{u}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \quad \forall \boldsymbol{z} \in \mathbb{R}^n$$

Now since the above equation is satisfied for all $\boldsymbol{z} \in \mathbb{R}^n$, we assume $\boldsymbol{z} \in \text{int}(\mathcal{C})$ such that $(\boldsymbol{z}-\boldsymbol{x})$ can be anywhere in the ball. Hence $\boldsymbol{u}$ needs to be 0 in this case.

Case 2 In this case we assume $\boldsymbol{x} \in Bd(\mathcal{C})$. That gives

$$\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) \geq \boldsymbol{u}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \quad \forall \boldsymbol{z} \in \mathbb{R}^n$$

If we take $\boldsymbol{z} \in \mathcal{C}$ then $\boldsymbol{u}$ satisfies $\boldsymbol{u}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \leq 0 \quad \forall \boldsymbol{z} \in \mathcal{C}$
If $\boldsymbol{z} \notin \mathcal{C}$ then $\boldsymbol{u}$ can take all the value. Hence taking intersection, $\boldsymbol{u}$ satisfies

$$\boldsymbol{u}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \leq 0 \quad \forall \boldsymbol{z} \in \mathcal{C}$$

Case 3 When we assume $\boldsymbol{x} \notin \mathcal{C}$, we get

$$\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) \geq +\infty + \boldsymbol{u}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \quad \forall \boldsymbol{z} \in \mathbb{R}^n$$

If we again take $\boldsymbol{z} \in \mathcal{C}$ then no finite $\boldsymbol{u}$ can satisfy the equation $\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) \geq +\infty + \boldsymbol{u}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \quad \forall \boldsymbol{z} \in \mathcal{C}$ because $\boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) = 0$ if $\boldsymbol{z} \in \mathcal{C}$.
And if $\boldsymbol{z} \notin \mathcal{C} \Rightarrow \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{z}) = +\infty$ then again nothing can be said about the vector $\boldsymbol{u}$.
Hence by convention it is assumed that $\boldsymbol{x} \notin \mathcal{C} \Rightarrow \boldsymbol{u} \in \emptyset$

By the above arguments we conclude that,

1. For $\boldsymbol{x} \notin \mathcal{C}$, $\partial \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x}) = \emptyset$
2. For $\boldsymbol{x} \in \mathcal{C}$, we have that $\boldsymbol{w} \in \partial \boldsymbol{\iota}_{\mathcal{C}}(\boldsymbol{x})$ if $\boldsymbol{w}^{\top}(\boldsymbol{z}-\boldsymbol{x}) \leq 0 \quad \forall \boldsymbol{z} \in \mathcal{C}$

Hence the claim made in Lemma A.10.0.1 is proved. □

Proof of Lemma A.4.0.1. From Lemma A.10.0.1, we know the expression for subgradient of the indication function $\boldsymbol{\iota}_{\mathcal{C}}$

$$\begin{aligned} \partial g(\boldsymbol{x}^{\star}) &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \mathcal{C} \;\; \boldsymbol{s}^{\top}(\boldsymbol{z}-\boldsymbol{x}^{\star}) \leq 0 \quad \right\} \\ &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \mathcal{C} \;\; \boldsymbol{s}^{\top}\boldsymbol{z} \leq \boldsymbol{s}^{\top}\boldsymbol{x}^{\star} \quad \right\} \end{aligned} \tag{A.35}$$

Now, by the optimality condition (A.2a), $-A^{\top}\boldsymbol{w}^{\star} \in \partial g(\boldsymbol{x}^{\star})$ and since this holds, hence $-A^{\top}\boldsymbol{w}^{\star}$ should satisfy the required constrained which is needed to be in the set of

subgradients of $\partial g(\boldsymbol{x}^\star)$ according to conditions in equation (A.40). Hence,

$$
\begin{aligned}
&(-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \leq (-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \quad \forall \boldsymbol{z} \in \mathcal{C} && \text{(A.36)}\\
\Rightarrow\ &(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad \forall \boldsymbol{z}\mathcal{C} && \text{(A.37)}\\
\Rightarrow\ &(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq \min_{z} (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad s.t \quad \boldsymbol{z} \in \mathcal{C} && \text{(A.38)}\\
\Rightarrow\ &(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star \leq \min_{\boldsymbol{z}\in\mathcal{C}} (A\boldsymbol{z})^\top \boldsymbol{w}^\star \quad s.t \quad \boldsymbol{z} \in \mathcal{C} && \text{(A.39)}
\end{aligned}
$$

Since $\boldsymbol{x}^\star$ is a feasible point hence $(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star = \min_{\boldsymbol{z}\in\mathcal{C}} (A\boldsymbol{z})^\top \boldsymbol{w}^\star \quad s.t \quad \boldsymbol{x}^\star, \boldsymbol{z} \in \mathcal{C}.$ □

A.10.1 Screening on Simplex Constrained Problems (Section A.4.1)

General Simplex Constrained Screening

Proof of Theorem A.4.1.1. In the simplex case, we have $g(\boldsymbol{x}) = \boldsymbol{\iota}_{\triangle}(\boldsymbol{x})$ and by Lemma A.10.0.1

$$
\begin{aligned}
\partial g(\boldsymbol{x}^\star) &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \triangle \ \ \boldsymbol{s}^\top (\boldsymbol{z} - \boldsymbol{x}^\star) \leq 0 \quad \right\} \\
&= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \triangle \ \ \boldsymbol{s}^\top \boldsymbol{z} \leq \boldsymbol{s}^\top \boldsymbol{x}^\star \quad \right\} && \text{(A.40)}
\end{aligned}
$$

Now, by the optimality condition (A.2a), $-A^\top \boldsymbol{w}^\star \in \partial g(\boldsymbol{x}^\star)$ and since this holds, hence $-A^\top \boldsymbol{w}^\star$ should satisfy the required constrained which is needed to be in the set of subgradients of $\partial g(\boldsymbol{x}^\star)$ according to conditions in equation (A.40). Hence,

$$
\begin{aligned}
&(-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \leq (-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \quad \forall \boldsymbol{z} \in \triangle && \text{(A.41)}\\
\Rightarrow\ &(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad \forall \boldsymbol{z} \in \triangle && \text{(A.42)}\\
\Rightarrow\ &(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq \min_{z} (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad s.t \quad \boldsymbol{z} \in \triangle && \text{(A.43)}\\
\Rightarrow\ &(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq \min_{i} \boldsymbol{a}_i^\top \boldsymbol{w}^\star && \text{(A.44)}\\
\Rightarrow\ &(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star \leq \min_{i} \boldsymbol{a}_i^\top \boldsymbol{w}^\star && \text{(A.45)}\\
\Rightarrow\ &(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star = \min_{i} \boldsymbol{a}_i^\top \boldsymbol{w}^\star && \text{(A.46)}
\end{aligned}
$$

Equation (A.44) is due to the fact that $\boldsymbol{z}$ lie in the simplex, hence minimum value of $(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z}$ is $\min_i \boldsymbol{a}_i^\top \boldsymbol{w}^\star$ and equation (A.46) also comes from the same fact that $\boldsymbol{x}^\star$ lie in the simplex and hence $(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star$ can not be smaller than $\min_i \boldsymbol{a}_i^\top \boldsymbol{w}^\star$. That implies these two quantities need to be equal and all the i's where this equality doesn't hold refers to $x_i^\star = 0$ for all such i's.

$$
\boldsymbol{a}_i^\top \boldsymbol{w}^\star > \ (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \ \Rightarrow \ x_i = 0
$$

$$(\boldsymbol{a}_i - A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star > 0 \;\Rightarrow\; x_i = 0$$

□

Proof of Theorem A.4.1.2. From the optimality condition (A.1a), we have $\boldsymbol{w}^\star = \nabla f(A\boldsymbol{x}^\star)$ since f is differentiable. Hence,

$$\begin{aligned}
(\boldsymbol{a}_i - A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star &= (\boldsymbol{a}_i - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.47)}\\
&= (\boldsymbol{a}_i - A\boldsymbol{x}^\star + A\boldsymbol{x} - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.48)}\\
&= (\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) + (A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.49)}\\
&\geq (\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) \quad \{\text{From the optimality of } f(A\boldsymbol{x})\} && \text{(A.50)}\\
&= (\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) - (\boldsymbol{a}_i - A\boldsymbol{x})^\top (\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}^\star)) && \text{(A.51)}\\
&\geq (\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) - \|\boldsymbol{a}_i - A\boldsymbol{x}\| \|\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}^\star)\| && \text{(A.52)}\\
&\geq (\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) - L\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} \|\boldsymbol{a}_i - A\boldsymbol{x}\| && \text{(A.53)}
\end{aligned}$$

Eq. (A.50) comes from the fact that at the optimal point $\boldsymbol{x}^\star$, the inequality $(A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) \geq 0$ holds $\forall\, \boldsymbol{x}$. Equation (A.53) comes from Corollary A.3.2.4 for smooth function f over a constrained set $\mathcal{C}$.
Hence from Theorem A.4.1.1, we obtain the screening rule

$$(\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) > L\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} \|\boldsymbol{a}_i - A\boldsymbol{x}\| \;\Rightarrow\; x_i^\star = 0$$

□

Screening for Squared Hinge Loss SVM.

Proof of Corollary A.4.1.3. Theorem A.4.1.2 is directly applicable to problems of the form (A.13). The objective function $f(\boldsymbol{y}) = f(A\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A^\top A\boldsymbol{x}$ is strongly convex with parameter $\mu = 1$. Also the derivative ∇f is Lipschitz-continuous with parameter $L = 1$. To obtain an upper bound on the distance between any approximate solution and the optimal solution $\|A\boldsymbol{x} - A\boldsymbol{x}^\star\|$, we employ Lemma A.3.2.3. Since the constrained of the optimization problem is unit simplex and hence the value of Wolfe gap function $GW(\boldsymbol{x}) := \max_{\boldsymbol{y}\in\mathcal{C}} (A\boldsymbol{x} - A\boldsymbol{y})^\top \nabla f(A\boldsymbol{x})$ as defined in Section A.3 will be attained on one of the vertices. So, $GW(\boldsymbol{x}) = \max_{i\in 1\ldots m} (A\boldsymbol{x} - \boldsymbol{a}_i)^\top A\boldsymbol{x}$. Finally, Theorem A.4.1.2 gives us the screening rule for squared hinge loss SVM:

$$(\boldsymbol{a}_i - A\boldsymbol{x})^\top A\boldsymbol{x} > \sqrt{\max_{i\in 1\ldots m} (A\boldsymbol{x} - \boldsymbol{a}_i)^\top A\boldsymbol{x}} \|\boldsymbol{a}_i - A\boldsymbol{x}\| \;\Rightarrow\; x_i^\star = 0 \qquad \text{(A.54)}$$

□

Screening on Minimum Enclosing Ball.

Proof of Corollary A.4.1.4. The minimum enclosing ball problem can be formulated as an optimization problem of the form given in Equation (A.15):

$$\min_{\boldsymbol{c},r} \quad r^2 \quad \text{s.t.} \|\boldsymbol{c}-\boldsymbol{a}_i\|_2^2 \leq r^2 \quad \forall i \in [n]$$

As we have seen, the dual formulation can be written in the form of Equation (A.16) as given in [150, Chapter 8.7]:

$$\min_{\boldsymbol{x}} \ \boldsymbol{x}^\top A^\top A\boldsymbol{x} - \sum_{j=1}^{p} \boldsymbol{a}_j^\top \boldsymbol{a}_j x_j \quad \text{s.t. } \boldsymbol{x} \in \triangle$$

Now the function $\boldsymbol{x}^\top A^\top A\boldsymbol{x} - \sum_{j=1}^{p} \boldsymbol{a}_j^\top \boldsymbol{a}_j x_j$ is strongly convex in $A\boldsymbol{x}$ with parameter $\mu = 2$. Since the constrained of the optimization problem is unit simplex and hence the value of the Wolfe gap function $GW(\boldsymbol{x}) := \max_{\boldsymbol{y}\in\mathcal{C}} (A\boldsymbol{x}-A\boldsymbol{y})^\top \nabla f(A\boldsymbol{x})$ as defined in Section A.3 will be attained at one of the vertices of unit simplex. Hence Corollary A.3.2.4 gives $GW(\boldsymbol{x}) = \sqrt{\frac{1}{2}\max_i (\boldsymbol{x}-\boldsymbol{e}_i)^\top (2A^\top A\boldsymbol{x}+\boldsymbol{c}')}$. Now applying the findings of Theorem A.4.1.2, we get a sufficient condition for $\boldsymbol{a}_i$ to be non-influential, i.e. $\boldsymbol{a}_i$ lies in the interior of the MEB. But before that we will simplify the left hand side of the theorem A.4.1.2 a bit. $(\boldsymbol{a}_i - A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x})$ can we written as $(\boldsymbol{e}_i-\boldsymbol{x})^\top A^\top \nabla f(A\boldsymbol{x})$. Hence we get our result claimed in Corollary A.4.1.4.

$$(\boldsymbol{e}_i-\boldsymbol{x})^\top (2A^\top A\boldsymbol{x}+\boldsymbol{c}') > 2\sqrt{\tfrac{1}{2}\max_j (\boldsymbol{x}-\boldsymbol{e}_i)^\top (2A^\top A\boldsymbol{x}+\boldsymbol{c}')}\|\boldsymbol{a}_i - A\boldsymbol{x}\| \ \Rightarrow \ x_i^\star = 0 \quad \text{(A.55)}$$

That means $\boldsymbol{a}_i$ is non influential. □

A.10.2 Screening on L_1-ball Constrained Problems

Proof of Theorem A.4.2.1. In the constrained Lasso case, we have $g(\boldsymbol{x}) = \boldsymbol{\iota}_{\mathcal{B}_{L_1}}(\boldsymbol{x})$ and by Lemma A.10.0.1

$$\begin{aligned}\partial g(\boldsymbol{x}^\star) &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \mathcal{B}_{L_1} \ \ \boldsymbol{s}^\top(\boldsymbol{z}-\boldsymbol{x}^\star) \leq 0 \quad \right\} \\ &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \mathcal{B}_{L_1} \ \ \boldsymbol{s}^\top \boldsymbol{z} \leq \boldsymbol{s}^\top \boldsymbol{x}^\star \quad \right\} \end{aligned} \quad \text{(A.56)}$$

Now, by the optimality condition (A.2a), $-A^\top \boldsymbol{w}^\star \in \partial g(\boldsymbol{x}^\star)$ and since this holds, hence $-A^\top \boldsymbol{w}^\star$ should satisfy the required constrained which is needed to be in the set of

subgradients of $\partial g(\boldsymbol{x}^\star)$ according to conditions in equation (A.71). Hence,

$$
\begin{aligned}
& (-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \leq (-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \quad \forall \boldsymbol{z} \in \mathcal{B}_{L_1} && \text{(A.57)}\\
\Rightarrow\ & (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad \forall \boldsymbol{z} \in \mathcal{B}_{L_1} && \text{(A.58)}\\
\Rightarrow\ & (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq \min_{z} (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad s.t \quad \boldsymbol{z} \in \mathcal{B}_{L_1} && \text{(A.59)}\\
\Rightarrow\ & (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq -\max_i \left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right| && \text{(A.60)}\\
\Rightarrow\ & (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star \leq -\max_i \left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right| && \text{(A.61)}\\
\Rightarrow\ & (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star = -\max_i \left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right| && \text{(A.62)}
\end{aligned}
$$

Equation (A.60) is due to the fact that $\boldsymbol{z}$ lie in the L_1-ball and hence minimum value of $(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z}$ is $-\max_i \left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right|$ and Equation (A.62) also comes from the same fact that $\boldsymbol{x}^\star$ lie in the L_1-ball and hence $(A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star$ can not be smaller than $-\max_i \left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right|$. That implies these two quantities need to be equal and all the i's where this equality doesn't hold refers to $x_i^\star = 0$ for all such i's. Hence whenever these two quantities are not equal this holds:

$$
\begin{aligned}
& -\left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right| > (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star \ \Rightarrow\ x_i^\star = 0\\
\Rightarrow\ & \left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right| + (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star < 0 \ \Rightarrow\ x_i^\star = 0
\end{aligned}
$$

□

Proof of Theorem A.4.2.2. Using optimality condition (A.1a), we know that $\boldsymbol{w}^\star \in \partial f(A\boldsymbol{x})$

$$
\begin{aligned}
\left| \boldsymbol{a}_i^\top \boldsymbol{w}^\star \right| + (A\boldsymbol{x}^\star)^\top \boldsymbol{w}^\star &= \left| \boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}^\star) \right| + (A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.63)}\\
&= \left| \boldsymbol{a}_i^\top \left(\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}) + \nabla f(A\boldsymbol{x}^\star)\right) \right| + (A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.64)}\\
&\leq \left| \boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) \right| + \left| \boldsymbol{a}_i^\top \left(\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x})\right) \right|\\
&\qquad + (A\boldsymbol{x}^\star - A\boldsymbol{x} + A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.65)}\\
&= \left| \boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) \right| + \left| \boldsymbol{a}_i^\top \left(\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x})\right) \right|\\
&\qquad + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) - (A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.66)}\\
&\leq \left| \boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) \right| + \left| \boldsymbol{a}_i^\top \left(\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x})\right) \right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) && \text{(A.67)}\\
&\leq \left| \boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}) \right| + \left| \boldsymbol{a}_i^\top \left(\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x})\right) \right|
\end{aligned}
$$

$$+ (A\boldsymbol{x})^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}) + \nabla f(A\boldsymbol{x})) \tag{A.68}$$

$$\leq \left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + \left|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}))\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) + (A\boldsymbol{x})^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x})) \tag{A.69}$$

$$\leq \left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) + L(\|\boldsymbol{a}_i\| + \|A\boldsymbol{x}\|)\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} \tag{A.70}$$

Eq. (A.66) comes from the fact that at the optimal point $\boldsymbol{x}^\star$, the inequality $(A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) \geq 0$ holds $\forall \boldsymbol{x}$. Hence using Theorem A.4.2.1, Lemma A.3.2.3 and Corollary A.3.2.4, we get the screening rule for L_1 constrained as whenever,
$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}) + L(\|\boldsymbol{a}_i\| + \|A\boldsymbol{x}\|)\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} < 0 \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$ □

A.10.3 Screening on Elastic Net Constrained Problems

Proof of Theorem A.4.3.1. Formulation :

$$\begin{aligned} &\min_{\boldsymbol{x}} f(A\boldsymbol{x}) \\ s.t\;\; &\alpha\|\boldsymbol{x}\|_1 + \frac{(1-\alpha)}{2}\|\boldsymbol{x}\|_2^2 \leq 1 \\ &\Rightarrow \alpha \sum_{i=1}^{n} |\boldsymbol{x}_i| + \frac{(1-\alpha)}{2}\sum_{i=1}^{n} \boldsymbol{x}_i^2 \leq 1 \end{aligned}$$

In the elastic net constrained case, we have $g(\boldsymbol{x}) = \boldsymbol{\iota}_{\mathcal{B}_{L_E}}(\boldsymbol{x})$ where $\boldsymbol{\iota}_{\mathcal{B}_{L_E}}$ is elastic net norm ball. That implies

$$\boldsymbol{x} \in \boldsymbol{\iota}_{\mathcal{B}_{L_E}} \quad : \quad \alpha\|\boldsymbol{x}\|_1 + (1-\alpha)\|\boldsymbol{x}\|_2^2 \leq 1$$

From the subgradient of indicator function and optimality condition for A and B framework

$$\begin{aligned} \partial g(\boldsymbol{x}^\star) &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \mathcal{B}_{L_1} \;\; \boldsymbol{s}^\top(\boldsymbol{z} - \boldsymbol{x}^\star) \leq 0 \quad \right\} \\ &= \left\{ \boldsymbol{s} \mid \forall \boldsymbol{z} \in \mathcal{B}_{L_1} \;\; \boldsymbol{s}^\top \boldsymbol{z} \leq \boldsymbol{s}^\top \boldsymbol{x}^\star \quad \right\} \end{aligned} \tag{A.71}$$

Now, by the optimality condition (A.2a), $-A^\top \boldsymbol{w}^\star \in \partial g(\boldsymbol{x}^\star)$ and since this holds, hence $-A^\top \boldsymbol{w}^\star$ should satisfy the required constrained which is needed to be in the set of subgradients of $\partial g(\boldsymbol{x}^\star)$ according to conditions in equation (A.71). Hence,

$$(-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \leq (-A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \quad \forall \boldsymbol{z} \in \mathcal{B}_{L_E} \tag{A.72}$$

$$\Rightarrow (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad \forall \boldsymbol{z} \in \mathcal{B}_{L_E} \tag{A.73}$$

$$\Rightarrow (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \leq \min_{z} (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad s.t \quad \boldsymbol{z} \in \mathcal{B}_{L_E} \tag{A.74}$$

Since $\boldsymbol{x}^\star$ is a feasible point hence $(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star = \min_{z} (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \quad s.t \quad \boldsymbol{x}^\star, \boldsymbol{z} \in \mathcal{B}_{L_E}$. At the point where above equaliy hold $\boldsymbol{x}^\star$ would be same as optimal $\boldsymbol{z}$. Hence the problem reduces to,

$$\begin{aligned} & \min \ (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \\ s.t \ \ & \alpha \|\boldsymbol{z}\|_1 + \frac{(1-\alpha)}{2} \|\boldsymbol{z}\|_2^2 \leq 1 \\ & \Rightarrow \alpha \sum_{i=1}^{n} |z_i| + \frac{(1-\alpha)}{2} \sum_{i=1}^{n} z_i^2 \leq 1 \end{aligned}$$

Without the loss of generality let us assume that for $i \in \{1 \ldots m\}, \ z_i \geq 0$ and $i \in \{m+1 \ldots n\}, \ z_i \leq 0$. Hence the optimization problem can be written as :

$$\begin{aligned} & \min \ (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} \\ s.t \ \ & \alpha \Big(\sum_{i=1}^{m} z_i - \sum_{i=m+1}^{n} z_i \Big) + \frac{(1-\alpha)}{2} \sum_{i=1}^{n} z_i^2 \leq 1 \\ & -z_i \leq 0 \ \ for \ \ i \in \{1 \ldots m\} \\ & z_i \leq 0 \ \ for \ \ i \in \{m+1 \ldots n\} \end{aligned} \tag{A.75}$$

Writing lagrangian for optimization problem (A.75)

$$\mathcal{L}(\boldsymbol{z}, \lambda, u) = (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} - \sum_{i=1}^{m} \lambda_i z_i + \sum_{i=m+1}^{n} \lambda_i z_i + u \Big(\alpha \Big(\sum_{i=1}^{m} z_i - \sum_{i=m+1}^{n} z_i \Big) + \frac{(1-\alpha)}{2} \sum_{i=1}^{n} z_i^2 - 1 \Big)$$

Also optimization conditions are $\lambda_i \geq 0$, $\lambda_i z_i = 0$ and $\alpha \big(\sum_{i=1}^{m} z_i - \sum_{i=m+1}^{n} z_i \big) + (1-\alpha) \sum_{i=1}^{n} z_i^2 = 1$. Also we conclude from above that if $\lambda_i > 0 \Rightarrow z_i = 0$. From first order optimality condition,
For $i \in \{1 \ldots m\}$

$$\boldsymbol{a}_i^\top \boldsymbol{w}^\star - \lambda_i = -u \big(\alpha + (1-\alpha) |z_i| \big) \tag{A.76}$$

For $i \in \{m+1 \ldots n\}$

$$\boldsymbol{a}_i^\top \boldsymbol{w}^\star + \lambda_i = -u \big(\alpha + (1-\alpha) |z_i| \big) \tag{A.77}$$

Now in equations (A.76) and (A.77) we multiply by z_i and add them. We get:

$$(A^\top \boldsymbol{w}^\star)^\top \boldsymbol{z} + u[1 + \frac{(1-\alpha)}{2} \|z\|_2^2] = 0 \tag{A.78}$$

From equations (A.76), (A.77), (A.78) and optimality conditions discussed above we get:

$$|\boldsymbol{a}_i^\top \boldsymbol{w}^\star| + (A^\top \boldsymbol{w}^\star)^\top z \Big[\frac{\alpha + (1-\alpha)|z_i|}{1 + \frac{(1-\alpha)}{2}\|z\|_2^2}\Big] < 0 \Rightarrow z_i = 0$$

As discussed above $\boldsymbol{x}^\star$ share same solution as optimal z. Hence

$$|\boldsymbol{a}_i^\top \boldsymbol{w}^\star| + (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \Big[\frac{\alpha}{1 + \frac{(1-\alpha)}{2}\|\boldsymbol{x}^\star\|_2^2}\Big] < 0 \Rightarrow x_i^\star = 0$$

□

Proof of Theorem A.4.3.2. Using optimality condition (A.1a), we know that $\boldsymbol{w}^\star \in \partial f(A\boldsymbol{x})$

$$|\boldsymbol{a}_i^\top \boldsymbol{w}^\star| + (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \Big[\frac{\alpha}{1 + \frac{(1-\alpha)}{2}\|\boldsymbol{x}^\star\|_2^2}\Big] \leq |\boldsymbol{a}_i^\top \boldsymbol{w}^\star| + (A^\top \boldsymbol{w}^\star)^\top \boldsymbol{x}^\star \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.79}$$

$$= \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x}^\star)\Big| + (A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.80}$$

$$= \Big|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}) - \nabla f(A\boldsymbol{x}) + \nabla f(A\boldsymbol{x}^\star))\Big| + (A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.81}$$

$$\leq \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big| + \Big|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}))\Big| + (A\boldsymbol{x}^\star - A\boldsymbol{x} + A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.82}$$

$$= \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big| + \Big|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}))\Big| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) \Big[\frac{2\alpha}{3-\alpha}\Big] - (A\boldsymbol{x} - A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star) \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.83}$$

$$\leq \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big| + \Big|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}))\Big| + (A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x}^\star) \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.84}$$

$$\leq \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big| + \Big|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}))\Big| + (A\boldsymbol{x})^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}) + \nabla f(A\boldsymbol{x})) \Big[\frac{2\alpha}{3-\alpha}\Big] \tag{A.85}$$

$$\leq \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big| + \Big|\boldsymbol{a}_i^\top (\nabla f(A\boldsymbol{x}^\star) - \nabla f(A\boldsymbol{x}))\Big|$$

$$+(A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x})\big[\frac{2\alpha}{3-\alpha}\big]+(A\boldsymbol{x})^\top(\nabla f(A\boldsymbol{x}^\star)-\nabla f(A\boldsymbol{x}))\big[\frac{2\alpha}{3-\alpha}\big] \tag{A.86}$$

$$\leq \Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big|+(A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x})\big[\frac{2\alpha}{3-\alpha}\big] + L(\|\boldsymbol{a}_i\|+\|A\boldsymbol{x}\|\big[\frac{2\alpha}{3-\alpha}\big])\sqrt{\frac{GW(\boldsymbol{x})}{\mu}} \tag{A.87}$$

Eq. (A.83) comes from the fact that at the optimal point $\boldsymbol{x}^\star$, the inequality $(A\boldsymbol{x}-A\boldsymbol{x}^\star)^\top \nabla f(A\boldsymbol{x}^\star)\geq 0$ holds $\forall \boldsymbol{x}$. Hence using Theorem A.4.2.1, Lemma A.3.2.3 and Corollary A.3.2.4, we get the screening rule for L_1 constrained as whenever,

$$\Big|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\Big|+(A\boldsymbol{x})^\top \nabla f(A\boldsymbol{x})\big[\frac{2\alpha}{3-\alpha}\big]+L(\|\boldsymbol{a}_i\|_2+\|A\boldsymbol{x}\|_2\big[\frac{2\alpha}{3-\alpha}\big])\sqrt{\frac{GW(\boldsymbol{x})}{\mu}}<0 \Rightarrow \boldsymbol{x}_i^\star=0$$

□

A.10.4 Screening for Box Constrained Problems

Screening for General Box Constrained Problems (Section A.4.4)

Proof of Theorem A.4.4.1. The box-constrained case can be seen in the form of the partially separable optimization problem pair (B.17) and (SB). According to optimality condition (A.4a) for this case, we have

$$-\boldsymbol{a}_i^\top \boldsymbol{w}^\star \in \partial g_i(x_i^\star) \qquad \forall i \tag{A.88}$$

Now from the definition of subgradient for an indicator function as given in Lemma A.10.0.1. Also since x_i is a number now, we will get rid of the transpose here.

$$\begin{aligned}\partial g(x_i^\star) &= \{s \mid 0\leq z\leq C,\ \ s(z-x_i^\star)\leq 0 \quad\}\\ &= \{s \mid 0\leq z\leq C,\ \ sz\leq sx_i^\star \quad\}\end{aligned} \tag{A.89}$$

Now, by the optimality condition (A.4a), $-\boldsymbol{a}_i^\top \boldsymbol{w}^\star \in \partial g(x_i^\star)$ and since this holds, hence $-\boldsymbol{a}_i^\top \boldsymbol{w}^\star$ should satisfy the required constrained which is needed to be in the set of subgradients of $\partial g(x_i^\star)$ according to conditions in Equation (A.89). Hence,

$$\begin{aligned}&(-\boldsymbol{a}_i^\top \boldsymbol{w}^\star)z\leq(-\boldsymbol{a}_i^\top \boldsymbol{w}^\star)x_i^\star \quad \forall z \ \text{ s.t } \ 0\leq z\leq C,\\ \Rightarrow &\min_z(\boldsymbol{a}_i^\top \boldsymbol{w}^\star)z\geq(\boldsymbol{a}_i^\top \boldsymbol{w}^\star)x_i^\star \quad s.t \ \ 0\leq z\leq C\end{aligned} \tag{A.90}$$

Now (A.90) can be manipulated in two ways

Case 1

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{w}^\star > 0 \Rightarrow \min_z (\boldsymbol{a}_i^\top \boldsymbol{w}^\star)^\top z \geq (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star \quad s.t \quad 0 \leq z \leq C \\ \Rightarrow 0 \geq (\boldsymbol{a}_i^\top \boldsymbol{w}^\star)^\top x_i^\star \end{aligned}$$

But since $\boldsymbol{a}_i^\top \boldsymbol{w}^\star > 0$ and also $x_i^\star \geq 0$ hence $(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star \not< 0$. This implies $(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star = 0$ and hence if $\boldsymbol{a}_i^\top \boldsymbol{w}^\star > 0 \;\Rightarrow\; x_i^\star = 0$

Case 2

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{w}^\star < 0 \Rightarrow \min_z (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) z \geq (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star \quad s.t \quad 0 \leq z \leq C \\ \Rightarrow (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) C \geq (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star \end{aligned}$$

But since $\boldsymbol{a}_i^\top \boldsymbol{w}^\star < 0$ and also $x_i^\star \leq C$ hence $(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star \not< (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) C$. This implies $(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) x_i^\star = (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) C$ and hence if $\boldsymbol{a}_i^\top \boldsymbol{w}^\star < 0 \;\Rightarrow\; x_i^\star = C$

Final optimality arguments can be given as

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{w}^\star > 0 \;\Rightarrow\; x_i^\star = 0 \\ \boldsymbol{a}_i^\top \boldsymbol{w}^\star < 0 \;\Rightarrow\; x_i^\star = C \end{aligned} \tag{A.91}$$

Now

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{w}^\star = \boldsymbol{a}_i^\top (\boldsymbol{w}^\star + \boldsymbol{w} - \boldsymbol{w}) = \boldsymbol{a}_i^\top \boldsymbol{w} + \boldsymbol{a}_i^\top (\boldsymbol{w}^\star - \boldsymbol{w}) \\ \boldsymbol{a}_i^\top \boldsymbol{w} - \|\boldsymbol{a}_i\|_2 \|\boldsymbol{w} - \boldsymbol{w}^\star\|_2 \leq \boldsymbol{a}_i^\top \boldsymbol{w}^\star \leq \boldsymbol{a}_i^\top \boldsymbol{w} + \|\boldsymbol{a}_i\|_2 \|\boldsymbol{w} - \boldsymbol{w}^\star\|_2 \end{aligned} \tag{A.92}$$

Since f is L-Lipschitz gradient hence f^* is $1/L$-strongly convex, hence using Lemmas A.3.2.1 and A.9.2.2, Equation (A.91) becomes

$$\boldsymbol{a}_i^\top \boldsymbol{w} - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} \leq \boldsymbol{a}_i^\top \boldsymbol{w}^\star \leq \boldsymbol{a}_i^\top \boldsymbol{w} + \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} \tag{A.93}$$

Hence using equation (A.93) and earlier arguments we get,

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{w}^\star > 0 \;\Rightarrow\; x_i^\star = 0 \\ \Rightarrow \boldsymbol{a}_i^\top \boldsymbol{w} - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} > 0 \;\Rightarrow\; x_i^\star = 0 \end{aligned}$$

And if

$$\begin{aligned} \boldsymbol{a}_i^\top \boldsymbol{w}^\star < 0 \;\Rightarrow\; x_i^\star = C \\ \Rightarrow \boldsymbol{a}_i^\top \boldsymbol{w} + \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} > 0 \;\Rightarrow\; x_i^\star = C \end{aligned}$$

□

Screening on SVM with hinge loss and no bias

Proof of Corollary A.4.4.2. Here the primal problem is given by:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\varepsilon}} \quad & \tfrac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\mathbf{1}^\top \boldsymbol{\varepsilon} \\ \text{s.t.} \quad & \boldsymbol{w}^\top \boldsymbol{a}_i \geq 1 - \varepsilon_i \quad \forall i \in \{1:p\} \\ & \varepsilon_i \geq 0 \qquad\qquad \forall i \in \{1:p\} \end{aligned} \tag{A.94}$$

A dual formulation of the problem can be written as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & -\boldsymbol{x}^\top \mathbf{1} + \tfrac{1}{2}\boldsymbol{x}^\top A^\top A\boldsymbol{x} \\ \text{s.t.} \quad & 0 \leq \boldsymbol{x} \leq C\mathbf{1} \end{aligned} \tag{A.95}$$

Theorem A.4.4.1 is applied on the dual formulation. The objective function $\frac{1}{2}\boldsymbol{x}^\top A^\top A\boldsymbol{x} - \boldsymbol{x}^\top \mathbf{1}$ is strongly convex with parameter 1 and its derivative Lipschitz continuous with parameter 1. The duality gap between primal and dual feasible points $G(\boldsymbol{w},\boldsymbol{\varepsilon},\boldsymbol{x})$ is now used as suboptimality certificate which can play the role of the upper bound $\|\boldsymbol{w}-\boldsymbol{w}^\star\|$ using Lemma A.3.2.2. For a given $\boldsymbol{x}$ a primal feasible point can be obtained by setting $\boldsymbol{w} = A\boldsymbol{x}$ and $\boldsymbol{\varepsilon}$ minimal such that the first constraint of the primal problem is satisfied. Using the obtained point for the duality gap, it only depends on the point $\boldsymbol{x}$. All together this gives the screening rule:

$$\boldsymbol{a}_i^\top A\boldsymbol{x} + 1 > \|\boldsymbol{a}_i\|\sqrt{2G(\boldsymbol{x})} \quad \Rightarrow \quad x_i^\star = 0 \tag{A.96}$$

$$\boldsymbol{a}_i^\top A\boldsymbol{x} + 1 < -\|\boldsymbol{a}_i\|\sqrt{2G(\boldsymbol{x})} \quad \Rightarrow \quad x_i^\star = C \tag{A.97}$$

□

Note - Since the primal and dual of hinge loss SVM have very nice structure with smooth quadratic function with an addition to piece-wise linear convex function, hence it is not hard to show that both primal and dual function is 1 strongly convex as shown in [275]. For more detailed proof, we recommend to go through [275]. Now for an instance, if we write duality gap function as a function of $\boldsymbol{w}$ then

$$G(\boldsymbol{w}) \geq G(\boldsymbol{w}^\star) + \nabla G(\boldsymbol{w}^\star)^\top (\boldsymbol{w}-\boldsymbol{w}^\star) + \|\boldsymbol{w}-\boldsymbol{w}^\star\|_2^2$$

Since strong duality hold in SVM case, hence at optimal point $\boldsymbol{w}^\star$, $G(\boldsymbol{w}^\star) = 0$. Finally we get,

$$G(\boldsymbol{w}) \geq \|\boldsymbol{w}-\boldsymbol{w}^\star\|_2^2$$

Hence the screening rule comes out as given in [275]:

$$\boldsymbol{a}_i^\top A\boldsymbol{x} + 1 > \|\boldsymbol{a}_i\|\sqrt{G(\boldsymbol{x})} \quad \Rightarrow \quad x_i^\star = 0 \tag{A.98}$$

$$\boldsymbol{a}_i^\top A\boldsymbol{x} + 1 < -\|\boldsymbol{a}_i\|\sqrt{G(\boldsymbol{x})} \quad \Rightarrow \quad x_i^\star = C \tag{A.99}$$

A.11 Screening on Penalized Problems

A.11.1 Screening L_1-regularized Problems

Lemma A.11.1.1. *Considering general L_1-regularized optimization problems*

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} \quad f(A\boldsymbol{x}) + \lambda \|\boldsymbol{x}\|_1 \tag{A.100}$$

At optimum points $\boldsymbol{x}^\star$ and dual optimal point $\boldsymbol{w}^\star$, the following rule is satisfied for the above problem formulation (A.100) :

$$\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| < \lambda \;\Rightarrow\; x_i^\star = 0$$

Proof. Since the optimization problem (A.100) comes under the partially separable framework and we can use the first order optimality condition (A.4a) as well as (A.4b) to derive screening rules for the problem. Also we know that, the conjugate of the norm function is the indicator function of its dual norm ball. By the optimality condition (A.4b), we know that

$$x_i \in \partial g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w})$$

here g_i^* is the indicator function written as $\boldsymbol{\iota}_{L_\infty}(-\boldsymbol{a}_i^\top \boldsymbol{w})$. Hence for the indicator function g^* by Lemma A.10.0.1

$$\begin{aligned}\partial g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w}^\star) &= \left\{ s|\ \ \forall \boldsymbol{z}\ s.t\ \ \left|\frac{\boldsymbol{a}_i^\top \boldsymbol{z}}{\lambda}\right| \le 1\,;\ s(-\boldsymbol{a}_i^\top \boldsymbol{z} + \boldsymbol{a}_i^\top \boldsymbol{w}^\star) \le 0 \right\} \\ &= \left\{ s|\ \ \forall \boldsymbol{z}\ s.t\ \ \left|\boldsymbol{a}_i^\top \boldsymbol{z}\right| \le \lambda\,;\ s(\boldsymbol{a}_i^\top \boldsymbol{z}) \ge s(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \right\}\end{aligned}$$

Since the optimality condition (A.4b) holds hence $-x_i^\star$ should satisfy the required constrained which is needed to be in the set of subgradients of $\partial g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w}^\star)$ according to conditions given above. That is

$$-x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{z}) \le -x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \quad \forall \boldsymbol{z}\ s.t\ \ \left|\boldsymbol{a}_i^\top \boldsymbol{z}\right| \le \lambda \tag{A.101}$$

$$x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{z}) \ge x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \quad \forall \boldsymbol{z}\ s.t\ \ \left|\boldsymbol{a}_i^\top \boldsymbol{z}\right| \le \lambda \tag{A.102}$$

$$\Rightarrow x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \le \min_z(x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{z})) \quad s.t \quad \left|\boldsymbol{a}_i^\top \boldsymbol{z}\right| \le \lambda \tag{A.103}$$

Case 1: $x_i^\star > 0$.

$$x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \le \min_z(x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{z})) \quad s.t \quad \left|\boldsymbol{a}_i^\top \boldsymbol{z}\right| \le \lambda$$

$$\begin{aligned}
&\Rightarrow\ x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \leq -\lambda x_i^\star \\
&\Rightarrow (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \leq -\lambda \\
&\Rightarrow (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) = -\lambda
\end{aligned} \tag{A.104}$$

Equation (A.104) comes from the fact that $\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| \leq \lambda$

Case 2: $x_i^\star < 0$.

$$\begin{aligned}
&x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \leq \min_{z}(x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{z})) \quad s.t \quad \left|\boldsymbol{a}_i^\top \boldsymbol{z}\right| \leq \lambda \\
&\Rightarrow\ x_i^\star(\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \leq \lambda x_i^\star \\
&\Rightarrow (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) \geq \lambda \\
&\Rightarrow (\boldsymbol{a}_i^\top \boldsymbol{w}^\star) = \lambda
\end{aligned} \tag{A.105}$$

Equation (A.104) comes from the fact that $\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| \leq \lambda$

Case 3: $x_i^\star = 0$.
Since if we assume f as a continuous smooth function then $\boldsymbol{a}_i^\top \boldsymbol{w}^\star$ is also continuous. Now if we consider arguments given for $x_i^\star < 0$ and $x_i^\star > 0$ we conclude that $\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| = \lambda$ in all of the above two cases. Since $x_i^\star = 0$ is in the domain of the function (A), hence at $x_i^\star = 0$, $\boldsymbol{a}_i^\top \boldsymbol{w}^\star$ will lie in the open range of $-\lambda$ to λ. Which implies whenever $\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| < \lambda$, then $x_i^\star = 0$

Another view on the proof can be derived from the optimality condition (A.4a).
The optimization problem (A.100) can be taken as partially separable problem and from the optimality condition (A.4a) kk

$$-\boldsymbol{a}_i^\top \boldsymbol{w}^\star \in \partial g_i(x_i^\star) \tag{A.106}$$

$$\partial g_i(x_i^\star) \in \begin{cases} \lambda \frac{x_i^\star}{|x_i^\star|} & \text{if } x_i \neq 0 \\ [-\lambda, \lambda] & \text{if } x_i = 0 \end{cases} \tag{A.107}$$

From equations (A.114) and (A.115) we conclude that if

$$\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| < \lambda \ \Rightarrow\ x_i^\star = 0$$

□

Proof of Theorem A.5.1.1. From Equation (A.1a), we know that $\boldsymbol{w}^\star \in \partial f(A\boldsymbol{x}^\star)$. Hence from Lemma A.11.1.1,

$$\begin{aligned}
\left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| &= \left|\boldsymbol{a}_i^\top (\boldsymbol{w}^\star - \boldsymbol{w} + \boldsymbol{w})\right| \\
&\leq \left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| + \left|\boldsymbol{a}_i^\top (\boldsymbol{w}^\star - \boldsymbol{w})\right|
\end{aligned}$$

$$\leq \left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| + \|\boldsymbol{a}_i\|_2 \|\boldsymbol{w}^\star - \boldsymbol{w}\|_2$$
$$\leq \left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| + \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} \tag{A.108}$$

Eq. (A.108) comes from Corollary A.3.2.2. Now using Lemma A.11.1.1 and equation (A.108), we get

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| < \lambda - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$$

□

Penalized Lasso. Screening in this case can be derived from the existing "gap safe" approach [66, 161]. For completeness we here show that the same result follows from our Theorem A.5.1.1:

Proof of Corollary A.5.1.2. By observing the cost function for penalized lasso it can be concluded that

$$f(A\boldsymbol{x}) = \tfrac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2, \quad \boldsymbol{w} = A\boldsymbol{x} - \boldsymbol{b}, \quad \text{and } L = 1$$

Now results from Theorem A.5.1.1 can be directly applied here and hence the screening rule becomes

$$\left|\boldsymbol{a}_i^\top (A\boldsymbol{x} - \boldsymbol{b})\right| < \lambda - \|\boldsymbol{a}_i\| \sqrt{2G(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0.$$

□

This result is known in the literature [161], and we recover it using our proposed general approach in this paper by using Theorem A.5.1.1.

Also, by applying same trick as mentioned after the end of proof of Corollary A.4.4.2, we can show that we can get rid of the factor 2 here also. Here also it is not hard to see that primal and dual ((A) and (B)) both are 1 strongly convex in the dual variable $\boldsymbol{w}$. Hence by the same argument as made in the proof of Corollary A.4.4.2, we get that

$$G(\boldsymbol{w}) \geq \|\boldsymbol{w} - \boldsymbol{w}^\star\|_2^2$$

And the improved screening rule comes out to be

$$\left|\boldsymbol{a}_i^\top (A\boldsymbol{x} - \boldsymbol{b})\right| < \lambda - \|\boldsymbol{a}_i\| \sqrt{G(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0.$$

Logistic Regression with L_1-regularization

Proof. By observation we know that in equation (A.23)

$$f(A\boldsymbol{x}) = \sum_{i=1}^{n} \log(\exp([A\boldsymbol{x}]_i)+1) \text{ and } \boldsymbol{w} \text{ is elementwise vector of } w_i \text{ s.t} \quad w_i = \frac{\exp([A\boldsymbol{x}]_i)}{\exp([A\boldsymbol{x}]_i)+1}$$

According to [223, Lemma 5], we get that the function $f(A\boldsymbol{x})$ is 1-smooth. Hence $L = 1$ Now from theorem A.5.1.1, we derive the screening rule for logistic regression with L_1-regularization which is

$$\left| \boldsymbol{a}_i^\top \left(\frac{\exp(A\boldsymbol{x})}{\exp(A\boldsymbol{x})+1} \right) \right| < \lambda - \|\boldsymbol{a}_i\|_2 \sqrt{2G(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$$

where $\left(\frac{\exp(A\boldsymbol{x})}{\exp(A\boldsymbol{x})+1}\right)$ is element wise vector whose i_{th} element is $\left(\frac{\exp([A\boldsymbol{x}]_i)}{\exp([A\boldsymbol{x}]_i)+1}\right)$ This result is also known in the literature in [161] (or see also [257] for a similar approach) and we recover it using our prosed general approach in this paper by using Theorem A.5.1.1.

□

Elastic-net regularized regression

Proof of Corollary A.5.2.1.

$$\begin{aligned}
&\frac{1}{2}\|A\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \lambda_2\|\boldsymbol{x}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 \\
&= \frac{1}{2}[\boldsymbol{x}^\top A^\top A\boldsymbol{x} - 2\boldsymbol{b}^\top A\boldsymbol{x} + \boldsymbol{b}^\top\boldsymbol{b}] + \lambda_2\boldsymbol{x}^\top\boldsymbol{x} + \lambda_1\|\boldsymbol{x}\|_1 \\
&= \frac{1}{2}[\boldsymbol{x}^\top (A^\top A + 2\lambda_2 I)\boldsymbol{x} - 2\boldsymbol{b}^\top A\boldsymbol{x} + \boldsymbol{b}^\top\boldsymbol{b}] + \lambda_1\|\boldsymbol{x}\|_1 \qquad \text{(A.109)}
\end{aligned}$$

Now consider $A^\top A + 2\lambda_2 I = Q^\top Q$ and choose vector $\boldsymbol{m}$ such that $A^\top\boldsymbol{b} = Q^\top\boldsymbol{m}$. Hence line (A.109) can be written as

$$\begin{aligned}
&\tfrac{1}{2}\left[\boldsymbol{x}^\top (A^\top A + 2\lambda_2 I)\boldsymbol{x} - 2\boldsymbol{b}^\top A\boldsymbol{x} + \boldsymbol{b}^\top\boldsymbol{b}\right] + \lambda_1\|\boldsymbol{x}\|_1 \\
&= \tfrac{1}{2}\left[\boldsymbol{x}^\top Q^\top Q\boldsymbol{x} - 2\boldsymbol{m}^\top Q\boldsymbol{x} + \boldsymbol{m}^\top\boldsymbol{m} - \boldsymbol{m}^\top\boldsymbol{m} + \boldsymbol{b}^\top\boldsymbol{b}\right] + \lambda_1\|\boldsymbol{x}\|_1 \\
&= \tfrac{1}{2}\|Q\boldsymbol{x}-\boldsymbol{m}\|_2^2 + \tfrac{1}{2}\left[\boldsymbol{b}^\top\boldsymbol{b} - \boldsymbol{m}^\top\boldsymbol{m}\right] + \lambda_1\|\boldsymbol{x}\|_1
\end{aligned}$$

Now the optimization problem (A.24) can be written as

$$\min_{\boldsymbol{x}} \; \tfrac{1}{2}\|Q\boldsymbol{x}-\boldsymbol{m}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 \qquad \text{(A.110)}$$

Now results from Corollary A.5.1.2 can be directly applied to (A.110).
From observation, we know that $f(Q\boldsymbol{x}) = \frac{1}{2}\|Q\boldsymbol{x}-\boldsymbol{m}\|^2, \quad \boldsymbol{w} = Q\boldsymbol{x}-\boldsymbol{m}, \quad$ and $L = 1$

Simplification,

$$\begin{aligned}\left|\boldsymbol{q}_i^\top(Q\boldsymbol{x}-\boldsymbol{m})\right| &= \left|\boldsymbol{q}_i^\top Q\boldsymbol{x}-\boldsymbol{q}_i^\top\boldsymbol{m}\right| \\ &= \left|\boldsymbol{q}_i^\top Q\boldsymbol{x}-\boldsymbol{a}_i^\top\boldsymbol{b}\right| \\ &= \left|(\boldsymbol{a}_i^\top A+2\lambda_2\boldsymbol{e}_i^\top)\boldsymbol{x}-\boldsymbol{a}_i^\top\boldsymbol{b}\right| \end{aligned} \tag{A.111}$$

$$\begin{aligned}|\boldsymbol{q}_i|\sqrt{2G(\boldsymbol{x})} &= \sqrt{\boldsymbol{a}_i^\top\boldsymbol{a}_i+2\lambda_2}\sqrt{2G(\boldsymbol{x})} \\ &= \sqrt{2(\boldsymbol{a}_i^\top\boldsymbol{a}_i+2\lambda_2)G(\boldsymbol{x})}\end{aligned} \tag{A.112}$$

Now using results from Corollary A.5.1.2, equations (A.111) and (A.112), we get screening rules for elastic norm regularization regression problem as:

$$\left|(\boldsymbol{a}_i^\top A+2\lambda_2\boldsymbol{e}_i^\top)\boldsymbol{x}-\boldsymbol{a}_i^\top\boldsymbol{b}\right| < \lambda_1-\sqrt{2(\boldsymbol{a}_i^\top\boldsymbol{a}_i+2\lambda_2)(G(\boldsymbol{x}))} \;\Rightarrow\; x_i^\star=0.$$

□

Lemma A.11.1.2 (Conjugate of the Elastic Net Regularizer [Lemma 6 [223]]). *] For $\alpha\in(0,1]$, the elastic net function $g(\boldsymbol{x})=\frac{1-\alpha}{2}\|\boldsymbol{x}\|_2^2+\alpha\|\boldsymbol{x}\|_1$ is the convex conjugate of*

$$g^*(\boldsymbol{x})=\sum_i\left[\frac{1}{2(1-\alpha)}\left(\left[|x_i|-\alpha\right]_+\right)^2\right]=\sum_i g_i^*(x_i)$$

where $g_i(\beta_i)=\left[\frac{1-\alpha}{2}\beta_i^2+\alpha|\beta_i|\right]$ and $[.]_+$ is the positive part operator, $[s]_+=s\;\; for\;\; s>0$, and zero otherwise. Furthermore, this g^* is is smooth, i.e. has Lipschitz continuous gradient with constant $1/(1-\alpha)$.

Proof. The complete proof has been given in [223, Lemma 6] but we also provide proof here below.
From the definition of convex conjugate function,

$$\begin{aligned}g^*(\boldsymbol{x}) &= \sup_{\boldsymbol{\beta}}[\boldsymbol{x}^\top\boldsymbol{\beta}-g(\boldsymbol{\beta})] \\ &= \sup_{\boldsymbol{\beta}}\left[\boldsymbol{x}^\top\boldsymbol{\beta}-\left(\frac{1-\alpha}{2}\|\boldsymbol{\beta}\|_2^2+\alpha\|\boldsymbol{\beta}\|_1\right)\right] \\ &= \sup_{\beta_i}\left[\sum_i x_i\beta_i-\left(\sum_i\left(\frac{1-\alpha}{2}\beta_i^2+\alpha|\beta_i|\right)\right)\right]\;\;\forall\;\; i\in[n] \\ &= \sum_i\sup_{\beta_i}\left[x_i\beta_i-\left(\frac{1-\alpha}{2}\beta_i^2+\alpha|\beta_i|\right)\right]\;\;\forall\;\; i\in[n]\end{aligned}$$

$$= \sum_i g_i^*(x_i) \text{ , where } \quad g_i(\beta_i) = \frac{1-\alpha}{2}\beta_i^2 + \alpha|\beta_i|$$

Now,

$$g_i^*(x_i) = \sup_{\beta_i} \left[x_i\beta_i - \left(\frac{1-\alpha}{2}\beta_i^2 + \alpha|\beta_i|\right)\right]$$

Consider three cases now :

Case 1: $\beta > 0$.

$$g_i^*(x_i) = \sup_{\beta_i} \left[x_i\beta_i - \left(\frac{1-\alpha}{2}\beta_i^2 + \alpha\beta_i\right)\right]$$
$$\Rightarrow \beta_i = \frac{(x_i-\alpha)}{(1-\alpha)} \text{ that also implies } \quad x_i > \alpha$$
$$\text{Hence, } g_i^*(x_i) = \frac{(x_i-\alpha)^2}{2(1-\alpha)} \text{ whenevr } \quad x_i > \alpha$$

Case 2: $\beta < 0$.

$$g_i^*(x_i) = \sup_{\beta_i} \left[x_i\beta_i - \left(\frac{1-\alpha}{2}\beta_i^2 - \alpha\beta_i\right)\right]$$
$$\Rightarrow \beta_i = \frac{(x_i+\alpha)}{(1-\alpha)} \text{ that also implies } \quad x_i < -\alpha$$
$$\text{Hence, } g_i^*(x_i) = \frac{(x_i+\alpha)^2}{2(1-\alpha)} \text{ whenevr } \quad x_i < -\alpha$$

Case 3: $\beta = 0$.

$$g_i^*(x_i) = 0 \text{ that also implies } \quad |x_i| \leq \alpha$$

Hence,

$$g_i^*(x_i) = \frac{1}{2(1-\alpha)}\left(\left[|x_i| - \alpha\right]_+\right)^2$$

From all of the above arguments, $g^*(\boldsymbol{x}) = \sum_i \left[\frac{1}{2(1-\alpha)}\left(\left[|x_i| - \alpha\right]_+\right)^2\right] = \sum_i g_i^*(x_i)$ □

Theorem' A.5.2.2. *If we consider the general elastic net formulation of the form*

$$\min_{\boldsymbol{x}} \; f(A\boldsymbol{x}) + (1-\alpha)\tfrac{1}{2}\|\boldsymbol{x}\|_2^2 + \alpha\|\boldsymbol{x}\|_1 \tag{A.113}$$

If f is L-smooth, then the following screening rule holds for all $i \in [n]$:

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| < \alpha - \|\boldsymbol{a}_i\|_2\sqrt{2LG(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$$

Proof. Since the optimization problem (A.113) comes under the partially separable framework and we can use the first order optimality condition (A.4a) as well as (A.4b) to derive screening rules for the problem.

By optimality condition (A.4b), we know that

$$x_i \in \; \partial g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w})$$

From lemma A.11.1.2, $g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w}^\star) = \frac{1}{2(1-\alpha)}\left(\left[|\boldsymbol{a}_i^\top \boldsymbol{w}^\star| - \alpha\right]_+\right)^2$ and also $\partial g_i^*(-\boldsymbol{a}_i^\top \boldsymbol{w}) = 0$ whenever $\left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| \le \alpha$
Hence whenever $\left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| \le \alpha \Rightarrow x_i = 0$.

The same screening rule for elastic net regularized problem can be derived from the optimality condition (A.4a). The optimization problem (A.113) can be taken as partially separable problem and from the optimality condition (A.4a)

$$-\boldsymbol{a}_i^\top \boldsymbol{w}^\star \in \partial g_i(x_i^\star) \tag{A.114}$$

$$\partial g_i(x_i^\star) \in \begin{cases} \alpha\frac{x_i^\star}{|x_i^\star|} + (1-\alpha)x_i & \text{if } x_i \neq 0 \\ [-\alpha, \alpha] & \text{if } x_i = 0 \end{cases} \tag{A.115}$$

Hence, whenever $\left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| \le \alpha \Rightarrow x_i = 0$.

The above arguments also show the significance of symmetry in our formulation as structure (A) and (B). This formulation provides our framework more flexibility to be used in larger class of problem.
Now,

$$\begin{aligned} \left|\boldsymbol{a}_i^\top \boldsymbol{w}^\star\right| &= \left|\boldsymbol{a}_i^\top (\boldsymbol{w}^\star - \boldsymbol{w} + \boldsymbol{w})\right| \\ &\le \left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| + \left|\boldsymbol{a}_i^\top (\boldsymbol{w}^\star - \boldsymbol{w})\right| \\ &\le \left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| + \|\boldsymbol{a}_i\|_2\|\boldsymbol{w}^\star - \boldsymbol{w}\|_2 \\ &\le \left|\boldsymbol{a}_i^\top \boldsymbol{w}\right| + \|\boldsymbol{a}_i\|_2\sqrt{2LG(\boldsymbol{x})} \end{aligned} \tag{A.116}$$

Equation (A.116) comes directly from Corollary A.3.2.2. Hence finally we get the screening rules for general elastic net penalty problem which is very similar to screening for $L_1 - penalized$ problems:

$$\left|\boldsymbol{a}_i^\top \nabla f(A\boldsymbol{x})\right| < \alpha - \|\boldsymbol{a}_i\|_2 \sqrt{2LG(\boldsymbol{x})} \;\Rightarrow\; \boldsymbol{x}_i^\star = 0$$

Now the above mentioned rule can be made a bit tighter under some condition which is not very interesting to discuss here. □

A.11.2 Screening for Structured Norms

We use the same notation as mentioned in Section A.5.3 i.e., we use the notation $\{\boldsymbol{x}_1 \cdots \boldsymbol{x}_G\}$ to express a vector $\boldsymbol{x}$ as a partition of non-overlapping groups $g \in \mathcal{G}$ of variables, such that $\boldsymbol{x}^\top = \left[\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top \cdots \boldsymbol{x}_G^\top\right]$. Correspondingly, the matrix A can be denoted as the concatenation of the respective column groups $A = [A_1\, A_2 \cdots A_G]$, and $\sum_{g\in\mathcal{G}} |g| = n$.

Lemma A.11.2.1. *Now if we consider an optimization problem of the form*

$$\arg\min_{\boldsymbol{x}} f(A\boldsymbol{x}) + \sum_{g=1}^{G} \sqrt{\rho_g}\|\boldsymbol{x}_g\|_2$$

At the optimal point $\boldsymbol{x}^\star$ and dual optimal points $\boldsymbol{w}^\star$, we get rules according to the following equation:

$$\|A_g^\top \boldsymbol{w}^\star\|_2 < \sqrt{\rho_g} \;\Rightarrow\; \boldsymbol{x}_g^\star = 0$$

Proof. Dual of the problem is given by

$$\mathcal{O}_B(\boldsymbol{w}) = f^*(\boldsymbol{w}) + \sum_g \sqrt{\rho_g}\, \boldsymbol{\imath}_{L_\infty}\left(\frac{\|A_g^\top \boldsymbol{w}\|_2}{\sqrt{\rho_g}}\right) \tag{A.117}$$

Hence for the indicator function g_g^* by Lemma A.10.0.1

$$\begin{aligned}\partial g_g^*(-A_g^\top \boldsymbol{w}^\star) &= \left\{ \boldsymbol{s} \mid \; \forall \boldsymbol{z}\; s.t \;\; \|\frac{A_g^\top \boldsymbol{z}}{\sqrt{\rho_g}}\|_2 \le 1\,;\; \boldsymbol{s}^\top(-A_g^\top \boldsymbol{z} + A_g^\top \boldsymbol{w}^\star) \le 0 \right\} \\ &= \left\{ \boldsymbol{s} \mid \; \forall \boldsymbol{z}\; s.t \;\; \|A_g^\top \boldsymbol{z}\|_2 \le \sqrt{\rho_g};\; \boldsymbol{s}^\top(A_g^\top \boldsymbol{z}) \ge \boldsymbol{s}^\top(A_g^\top \boldsymbol{w}^\star) \right\}\end{aligned}$$

Now, by the optimality condition (A.4b) $\boldsymbol{x}_g \in \partial g_g^*(-A_g^\top \boldsymbol{w}^\star)$, and since this holds, hence $xv_g^\star$ should satisfy the required constrained which is needed to be in the set of subgradients of $\partial g^*(-A_g^\top \boldsymbol{w}^\star)$ according to conditions given above. Hence,

$$
\begin{aligned}
&-\boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{z}) \leq -\boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{w}^\star) \quad \forall \boldsymbol{z}\ s.t\ \ \|A_g^\top \boldsymbol{z}\|_2 \leq \sqrt{\rho_g} \\
&\Rightarrow \boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{z}) \geq \boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{w}^\star) \quad \forall \boldsymbol{z}\ s.t\ \ \|A_g^\top \boldsymbol{z}\|_2 \leq \sqrt{\rho_g} \\
&\Rightarrow\ \boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{w}^\star) \leq \min_z \boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{z}) \quad s.t\ \ \|A_g^\top \boldsymbol{z}\|_2 \leq \sqrt{\rho_g} \\
&\Rightarrow\ \boldsymbol{x}_g^{\star\top}(A_g^\top \boldsymbol{w}^\star) \leq \min_z \|\boldsymbol{x}_g\|_2 \|A_g^\top \boldsymbol{z}\|_2 \quad s.t\ \ \|A_g^\top \boldsymbol{z}\|_2 \leq \sqrt{\rho_g} \\
&\Rightarrow\ \boldsymbol{x}_g^{\top}(A_g^\top \boldsymbol{w}^\star) \leq -\|\boldsymbol{x}_g^\star\|_2 \sqrt{\rho_g} \\
&\Rightarrow \|A_g^\top \boldsymbol{w}^\star\|_2 = \sqrt{\rho_g}
\end{aligned} \tag{A.118}
$$

Equation (A.118) comes from the cauchy inequality and true $\forall \boldsymbol{x}_g^\star \ :\ \boldsymbol{x}_g^\star \neq 0$. Whenever $\|A_g^\top \boldsymbol{w}^\star\|_2 < \sqrt{\rho_g}$ then $\boldsymbol{x}_g^\star = 0$

Another view on the screening of above optimization problem can be seen from the optimality condition (A.4a). The optimization problem in Lemma A.11.2.1 can be taken as partially separable problem and from the optimality condition (A.4a)

$$
-A_g^\top \boldsymbol{w}^\star \in \partial g(\boldsymbol{x}_g^\star) \tag{A.119}
$$

$$
\partial g(\boldsymbol{x}_g^\star) \in \begin{cases} \sqrt{\rho_g} \frac{\boldsymbol{x}_g}{\|x_g\|_2} & \text{if } \boldsymbol{x}_g \neq 0 \\ \mathcal{B}_2 & \text{if } \boldsymbol{x}_g = 0 \text{ and } \mathcal{B}_2 \text{ is norm ball of radius } \sqrt{\rho_g} \end{cases} \tag{A.120}
$$

From Equations (A.119) and (A.120), we conclude that if

$$
\|A_g^\top \boldsymbol{w}^\star\|_2 < \sqrt{\rho_g} \ \Rightarrow\ \boldsymbol{x}_g^\star = 0
$$

□

Theorem' A.5.3.1. *For ℓ_2/ℓ_1-regularized optimization problem of the form*

$$
\min_{\boldsymbol{x}} f(A\boldsymbol{x}) + \sum_{g=1}^{G} \sqrt{\rho_g} \|\boldsymbol{x}_g\|_2
$$

Assuming f is L-smooth, then the following (group-level) screening rule holds for all groups g:

$$
\|A_g^\top \nabla f(A\boldsymbol{x})\|_2 + \sqrt{2L} \left\|A_g\right\|_{Fro} < \sqrt{\rho_g} \ \Rightarrow\ \boldsymbol{x}_g^\star = \mathbf{0} \in \mathbb{R}^{|g|}\ .
$$

Proof. From Equation (A.1a), we know that $\boldsymbol{w} \in \nabla f(A\boldsymbol{x})$. Now

$$
\begin{aligned}
\|A_g^\top \boldsymbol{w}^\star\|_2 &= \|A_g^\top (\boldsymbol{w} + \boldsymbol{w}^\star - \boldsymbol{w})\|_2 \leq \|A_g^\top \boldsymbol{w}\|_2 + \|A_g^\top (\boldsymbol{w}^\star - \boldsymbol{w})\|_2 \\
&= \|A_g^\top \boldsymbol{w}\|_2 + \sqrt{tr((A_g^\top (\boldsymbol{w}^\star - \boldsymbol{w}))((\boldsymbol{w}^\star - \boldsymbol{w})^\top) A_g)^\top}
\end{aligned}
$$

$$\begin{aligned} &\leq \|A_g^\top \boldsymbol{w}\|_2 + \sqrt{tr((\boldsymbol{w}^\star - \boldsymbol{w})^\top (\boldsymbol{w}^\star - \boldsymbol{w}))}\sqrt{tr(A_g^\top A_g)} \\ &= \|A_g^\top \boldsymbol{w}\|_2 + \|\boldsymbol{w}^\star - \boldsymbol{w}\|_2 \left\|A_g\right\|_{\text{Fro}} \end{aligned} \tag{A.121}$$

Using Corollary A.3.2.2 with Equation (A.121), we get

$$\|A_g^\top \boldsymbol{w}^\star\|_2 \leq \|A_g^\top \nabla f(A\boldsymbol{x})\|_2 + \sqrt{2LG(\boldsymbol{x})}\left\|A_g\right\|_{\text{Fro}}$$

Hence using previous Lemma A.11.2.1,

$$\|A_g^\top \nabla f(A\boldsymbol{x})\|_2 + \sqrt{2LG(\boldsymbol{x})}\left\|A_g\right\|_{\text{Fro}} < \sqrt{\rho_g} \;\Rightarrow\; \boldsymbol{x}_g^\star = 0$$

□

Proof of Corollary A.5.3.2. This is an explicit case of the optimization problem mentioned in Lemma A.11.2.1, see also [162]. By observation we know that,

$$f(A\boldsymbol{x}) = \tfrac{1}{2}\|A\boldsymbol{x} - b\|^2, \quad \boldsymbol{w} = A\boldsymbol{x} - b \quad \textit{and} \quad L = 1$$

Now applying the findings of Theorem A.5.3.1, we get

$$\|A_g^\top (A\boldsymbol{x} - \boldsymbol{b})\|_2 + \sqrt{2G(\boldsymbol{x})}\left\|A_g\right\|_{\text{Fro}} < \lambda\sqrt{\rho_g} \;\Rightarrow\; \boldsymbol{x}_g^\star = 0$$

□

In Lemma A.11.2.2 mentioned below, we show that the structured norm setting of [161] can be derived from our more general (A) and (B) structure.

Lemma A.11.2.2. *Sparse Multi-Task and Multi Class Model [161] - If we consider general problem of the form*

$$\min_{X \in \mathbb{R}^{p\times q}} \sum_{i=1}^{n} f_i(\boldsymbol{a}_i^\top X) + \lambda\Omega(X) \tag{A.122}$$

where the regularization function $\Omega : \mathbb{R}^{p\times q} \to \mathbb{R}_+$ *is such that* $\Omega(X) = \sum_{g=1}^{p} \|\boldsymbol{x}_g\|_2$ *and* $X = [\boldsymbol{x}_1, \boldsymbol{x}_2 \cdots \boldsymbol{x}_G]$. *We write* $W = [\boldsymbol{w}_1, \boldsymbol{w}_2 \cdots \boldsymbol{w}_G]$ *for variable of the dual problem. Then the screening rule becomes*

$$\|\boldsymbol{a}^{(g)^\top} W\|_2 < \lambda - \|\boldsymbol{a}^{(g)}\|_2 \|W - W^\star\|_2 \;\Rightarrow\; \boldsymbol{x}_g^\star = 0$$

Here $\boldsymbol{a}^{(g)}$ *is the vector of the* g^{th} *element group of each vector* $\boldsymbol{a}_i$.

Proof. Equations pair (A) and (B) can be used interchangeably by replacing primal with dual and f with g. Hence the partial separable primal-dual pair (B.17) and (SB) can

also be used interchangeably. By comparing Equation (A.122) with (B.17) and (SB), we observe that separable function $\sum_{i=1}^{n} f_i(\boldsymbol{a}_i^\top X)$ takes the place of separable g^* in (SB) and $\lambda\Omega(X)$ takes the place of f^*. Hence we apply the optimality condition (A.1b) to get (with exchanged primal dual variable)

$$AW^\star \in \partial\lambda\Omega(X^\star)$$

Hence if,

$$\|\boldsymbol{a}^{(g)^\top} W^\star\|_2 < \lambda \;\Rightarrow\; \boldsymbol{x}_g = 0 \tag{A.123}$$

Now,

$$\begin{aligned}\|\boldsymbol{a}^{(g)^\top} W^\star\|_2 &= \|\boldsymbol{a}^{(g)^\top}(W^\star - W + W)\|_2 \\ &\le \|\boldsymbol{a}^{(g)^\top} W\|_2 + \|\boldsymbol{a}^{(g)^\top}(W^\star - W)\|_2 \\ &\le \|\boldsymbol{a}^{(g)^\top} W\|_2 + \|\boldsymbol{a}^{(g)}\|_2 \|(W^\star - W)\|_2 \end{aligned} \tag{A.124}$$

Using equations (A.123) and (A.124), the screening rule comes out to be

$$\|\boldsymbol{a}^{(g)^\top} W\|_2 < \lambda - \|\boldsymbol{a}^{(g)}\|_2 \|W - W^\star\|_2 \;\Rightarrow\; \boldsymbol{x}_g^\star = 0$$

□

Corollary A.11.2.3. *If for all $i \in [n]$, f_i is L-Lipschitz gradient then screening rule for equation (A.122) is*

$$\|\boldsymbol{a}^{(g)^\top} W\|_2 < \lambda - \|\boldsymbol{a}^{(g)}\|_2 \sqrt{2LG(X)} \;\Rightarrow\; \boldsymbol{x}_g^\star = 0$$

Proof. Using Lemma A.11.2.2 and Corollary A.3.2.2, we get the desired expression. □

Appendix B

Approximate Steepest Coordinate Descent

Sebastian U. Stich[1], Anant Raj[2], Martin Jaggi[1]

1 – EPFL, Lausanne

2 – MPI for Intelligent Systems, Tübingen

Abstract

We propose a new selection rule for the coordinate selection in coordinate descent methods for huge-scale optimization. The efficiency of this novel scheme is provably better than the efficiency of uniformly random selection, and can reach the efficiency of steepest coordinate descent (SCD), enabling an acceleration of a factor of up to n, the number of coordinates. In many practical applications, our scheme can be implemented at no extra cost and computational efficiency very close to the faster uniform selection. Numerical experiments with Lasso and Ridge regression show promising improvements, in line with our theoretical guarantees.

B.1 Introduction

Coordinate descent (CD) methods have attracted a substantial interest the optimization community in the last few years [170, 204]. Due to their computational efficiency, scalability, as well as their ease of implementation, these methods are the state-of-the-art for a wide selection of machine learning and signal processing applications [74, 93, 262]. This is also theoretically well justified: The complexity estimates for CD methods are in general better than the estimates for methods that compute the full gradient in one batch pass [170, 176].

In many CD methods, the active coordinate is picked at random, according to a probability distribution. For smooth functions it is theoretically well understood how the

sampling procedure is related to the efficiency of the scheme and which distributions give the best complexity estimates [111, 170, 176, 193, 272]. For nonsmooth and composite functions — that appear in many machine learning applications — the picture is less clear. For instance in [71, 72, 215, 217] uniform sampling (UCD) is used, whereas other papers propose adaptive sampling strategies that change over time [49, 184, 185, 189].

A very simple deterministic strategy is to move along the direction corresponding to the component of the gradient with the maximal absolute value (steepest coordinate descent, SCD) [31, 252]. For smooth functions this strategy yields always better progress than UCD, and the speedup can reach a factor of the dimension [180]. However, SCD requires the computation of the whole gradient vector in each iteration which is prohibitive (except for special applications, cf. Dhillon *et al.* [57], Shrivastava and Li [221]).

In this paper we propose approximate steepest coordinate descent (ASCD), a novel scheme which combines the best parts of the aforementioned strategies: (i) ASCD maintains an approximation of the *full* gradient in each iteration and selects the active coordinate among the components of this vector that have large absolute values — similar to SCD; and (ii) in many situations the gradient approximation can be updated cheaply at no extra cost — similar to UCD. We show that regardless of the errors in the gradient approximation (even if they are infinite), ASCD performs always better than UCD.

Similar to the methods proposed in [252] we also present variants of ASCD for composite problems. We confirm our theoretical findings by numerical experiments for Lasso and Ridge regression on a synthetic dataset as well as on the RCV1 (binary) dataset.

Structure of the Paper and Contributions. In Sec. B.2 we review the existing theory for SCD and (i) extend it to the setting of smooth functions. We present (ii) a novel lower bound, showing that the complexity estimates for SCD and UCD can be equal in general. We (iii) introduce ASCD and the safe selection rules for both smooth (Sec. B.3) and to composite functions (Sec. B.5). We prove that (iv) ASCD performs always better than UCD (Sec. B.3) and (v) it can reach the performance of SCD (Sec. B.6). In Sec. B.4 we discuss important applications where the gradient estimate can efficiently be maintained. Our theory is supported by numerical evidence in Sec. B.7, which reveals that (vi) ASCD performs extremely well on real data.

Notation. Define $[\boldsymbol{x}]_i := \langle \boldsymbol{x}, \boldsymbol{e}_i \rangle$ with $\boldsymbol{e}_i$ the standard unit vectors in $\mathbb{R}^n$. We abbreviate $\nabla_i f := [\nabla f]_i$. A convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ with coordinate-wise L_i-Lipschitz continuous gradients[1] for constants $L_i > 0$, $i \in [n] := \{1, \dots, n\}$, satisfies by the standard reasoning

$$f(\boldsymbol{x} + \eta \boldsymbol{e}_i) \leq f(\boldsymbol{x}) + \eta \nabla_i f(\boldsymbol{x}) + \tfrac{L_i}{2} \eta^2 \tag{B.1}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\eta \in \mathbb{R}$. A function is coordinate-wise L-smooth if $L_i \leq L$ for $i = 1, \dots, n$. For an optimization problem $\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$ define $X^\star := \arg\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$ and denote by

[1] $|\nabla_i f(\boldsymbol{x} + \eta \boldsymbol{e}_i) - \nabla_i f(\boldsymbol{x})| \leq L_i |\eta|, \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \eta \in \mathbb{R}.$

$\boldsymbol{x}^\star \in \mathbb{R}^n$ an arbitrary element $\boldsymbol{x}^\star \in X^\star$.

B.2 Steepest Coordinate Descent

In this section we present SCD and discuss its theoretical properties. The functions of interest are composite convex functions $F\colon \mathbb{R}^n \to \mathbb{R}$ of the form

$$F(\boldsymbol{x}) := f(\boldsymbol{x}) + \Psi(\boldsymbol{x}) \tag{B.2}$$

where f is coordinate-wise L-smooth and Ψ convex and separable, that is that is $\Psi(\boldsymbol{x}) = \sum_{i=1}^n \Psi_i([\boldsymbol{x}]_i)$. In the first part of this section we focus on smooth problems, i.e. we assume that $\Psi \equiv 0$.

Coordinate descent methods with constant step size generate a sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ of iterates that satisfy the relation

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \tfrac{1}{L}\nabla_{i_t} f(\boldsymbol{x})\boldsymbol{e}_{i_t}\,. \tag{B.3}$$

In UCD the active coordinate i_t is chosen uniformly at random from the set $[n]$, $i_t \in_{u.a.r.} [n]$. SCD chooses the coordinate according to the Gauss-Southwell (GS) rule:

$$i_t = \operatorname*{arg\,max}_{i\in[n]} \nabla_i |f(\boldsymbol{x}_t)|\,. \tag{B.4}$$

B.2.1 Convergence analysis

With the quadratic upper bound (C.1) one can easily get a lower bound on the one step progress

$$\mathbb{E}\left[f(x_t) - f(x_{t+1}) \mid x_t\right] \geq \mathbb{E}_{i_t}\left[\tfrac{1}{2L}\left|\nabla_{i_t} f(\boldsymbol{x}_t)\right|^2\right]\,. \tag{B.5}$$

For UCD and SCD the expression on the right hand side evaluates to

$$\begin{aligned} \tau_{\mathrm{UCD}}(\boldsymbol{x}_t) &:= \tfrac{1}{2nL}\|\nabla f(\boldsymbol{x}_t)\|_2^2 \\ \tau_{\mathrm{SCD}}(\boldsymbol{x}_t) &:= \tfrac{1}{2L}\|\nabla f(\boldsymbol{x}_t)\|_\infty^2 \end{aligned} \tag{B.6}$$

With Cauchy-Schwarz we find

$$\tfrac{1}{n}\tau_{\mathrm{SCD}}(\boldsymbol{x}_t) \leq \tau_{\mathrm{UCD}}(\boldsymbol{x}_t) \leq \tau_{\mathrm{SCD}}(\boldsymbol{x}_t)\,. \tag{B.7}$$

Hence, the lower bound on the one step progress of SCD is always at least as large as the lower bound on the one step progress of UCD. Moreover, the one step progress could be even lager by a factor of n. However, it is very difficult to formally prove that this linear

speed-up holds for more than one iteration, as the expressions in (B.7) depend on the (a priori unknown) sequence of iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$.

Strongly Convex Objectives. Nutini *et al.* [180] present an elegant solution of this problem for μ_2-strongly convex functions[2]. They propose to measure the strong convexity of the objective function in the 1-norm instead of the 2-norm. This gives rise to the lower bound

$$\tau_{\text{SCD}}(\boldsymbol{x}_t) \geq \frac{\mu_1}{L}\left(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)\right), \tag{B.8}$$

where μ_1 denotes the strong convexity parameter. By this, they get a uniform upper bound on the convergence that does not directly depend on local properties of the function, like for instance $\tau_{\text{SCD}}(\boldsymbol{x}_t)$, but just on μ_1. It always holds $\mu_1 \leq \mu_2$, and for functions where both quantities are equal, SCD enjoys a linear speedup over UCD.

Smooth Objectives. When the objective function f is just smooth (but not necessarily strongly convex), then the analysis mentioned above is not applicable. We here extend the analysis from [180] to smooth functions.

Theorem B.2.1.1. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be convex and coordinate-wise L-smooth. Then for the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD it holds:*

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq \frac{2LR_1^2}{t}, \tag{B.9}$$

for $R_1 := \max_{\boldsymbol{x}^\star \in X^\star}\left\{\max_{\boldsymbol{x}\in\mathbb{R}^n}\left[\|\boldsymbol{x}-\boldsymbol{x}^\star\|_1 \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}_0)\right]\right\}$.

Proof. In the proof we first derive a lower bound on the one step progress (Lemma B.9.1.1), similar to the analysis in [170]. The lower bound for the one step progress of SCD can in each iteration differ up to a factor of n from the analogous bound derived for UCD (similar as in (B.7)). All details are given in Section B.9.1 in the appendix. □

Note that the R_1 is essentially the diameter of the level set at $f(\boldsymbol{x}_0)$ measured in the 1-norm. In the complexity estimate of UCD, R_1^2 in (B.9) is replaced by nR_2^2, where R_2 is the diameter of the level at $f(\boldsymbol{x}_0)$ measured in the 2-norm (cf. Nesterov [170], Wright [262]). As in (B.7) we observe with Cauchy-Schwarz

$$\tfrac{1}{n}R_1^2 \leq R_2^2 \leq R_1^2, \tag{B.10}$$

i.e. SCD can accelerate up to a factor of n over to UCD.

[2] A function is μ_p-strongly convex in the p-norm, $p \geq 1$, if $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y}-\boldsymbol{x}\rangle + \frac{\mu_p}{2}\|\boldsymbol{y}-\boldsymbol{x}\|_p^2$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

B.2.2 Lower bounds

In the previous section we provided complexity estimates for the methods SCD and UCD and showed that SCD can converge up to a factor of the dimension n faster than UCD. In this section we show that this analysis is tight. In Theorem B.2.2.1 below we give a function $q\colon \mathbb{R}^n \to \mathbb{R}$, for which the one step progress $\tau_{\mathrm{SCD}}(\boldsymbol{x}_t) \approx \tau_{\mathrm{UCD}}(\boldsymbol{x}_t)$ up to a constant factor, for all iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD.

By a simple technique we can also construct functions for which the speedup is exactly equal to an arbitrary factor $\lambda \in [1,n]$. For instance we can consider functions with a (separable) low dimensional structure. Fix integers s,n such that $\frac{n}{s} \approx \lambda$, define the function $f\colon \mathbb{R}^n \to \mathbb{R}$ as

$$f(\boldsymbol{x}) := q(\pi_s(\boldsymbol{x})) \tag{B.11}$$

where π_s denotes the projection to $\mathbb{R}^s$ (being the first s out of n coordinates) and $q\colon \mathbb{R}^s \to \mathbb{R}$ is the function from Theorem B.2.2.1. Then

$$\tau_{\mathrm{SCD}}(\boldsymbol{x}_t) \approx \lambda \cdot \tau_{\mathrm{UCD}}(\boldsymbol{x}_t)\,, \tag{B.12}$$

for all iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD.

Theorem B.2.2.1. *Consider the function $q(\boldsymbol{x}) = \frac{1}{2}\langle Q\boldsymbol{x},\boldsymbol{x}\rangle$ for $Q := I_n - \frac{99}{100n}J_n$, where $J_n = 1_n 1_n^T$, $n > 2$. Then there exists $\boldsymbol{x}_0 \in R^n$ such that for the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD it holds*

$$\|\nabla q(\boldsymbol{x}_t)\|_\infty^2 \leq \tfrac{4}{n}\|\nabla q(\boldsymbol{x}_t)\|_2^2\,. \tag{B.13}$$

Proof. In the appendix we discuss a family of functions defined by matrices $Q := (\alpha - 1)\frac{1}{n}J_n + I_n$ and define corresponding parameters $0 < c_\alpha < 1$ such that for $\boldsymbol{x}_0$ defined as $[\boldsymbol{x}_0]_i = c_\alpha^{i-1}$ for $i = 1,\ldots,n$, SCD cycles through the coordinates, that is, the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD satisfies

$$[\boldsymbol{x}_t]_{1+(t-1 \mod n)} = c_\alpha^n \cdot [\boldsymbol{x}_{t-1}]_{1+(t-1 \mod n)}\,. \tag{B.14}$$

We verify that for this sequence property (B.13) holds. □

B.2.3 Composite Functions

The generalization of the GS rule (B.4) to composite problems (B.2) with nontrival Ψ is not straight forward. The 'steepest' direction is not always meaningful in this setting; consider for instance a constrained problem where this rule could yield no progress at all when stuck at the boundary.

Nutini *et al.* [180] discuss several generalizations of the Gauss-Southwell rule for composite functions. The GS-s rule is defined to choose the coordinate with the most

negative directional derivative [263]. This rule is identical to (B.4) but requires the calculation of subgradients of Ψ_i. However, the length of a step could be arbitrarily small. In contrast, the GS-r rule was defined to pick the coordinate direction that yields the longest step [252]. The rule that enjoys the best theoretical properties (cf. Nutini *et al.* [180]) is the GS-q rule, which is defined as to maximize the progress assuming a quadratic upper bound on f [252]. Consider the coordinate-wise models

$$V_i(\boldsymbol{x}, y, s) := sy + \tfrac{L}{2}y^2 + \Psi_i([\boldsymbol{x}]_i + y)\,, \tag{B.15}$$

for $i \in [n]$. The GS-q rule is formally defined as

$$i_{\mathrm{GS-q}} = \operatorname*{arg\,min}_{i\in[n]} \min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x}))\,. \tag{B.16}$$

B.2.4 The Complexity of the GS rule

So far we only studied the iteration complexity of SCD, but we have disregarded the fact that the computation of the GS rule (B.4) can be as expensive as the computation of the whole gradient. The application of coordinate descent methods is only justified if the complexity to compute one directional derivative is approximately n times cheaper than the computation of the full gradient vector (cf. Nesterov [170]). By Theorem B.2.2.1 this reasoning also applies to SCD. A class of function with this property is given by functions $F\colon \mathbb{R}^n \to \mathbb{R}$

$$F(\boldsymbol{x}) := f(A\boldsymbol{x}) + \sum_{i=1}^{n} \Psi_i([\boldsymbol{x}]_i) \tag{B.17}$$

where A is a $d \times n$ matrix, and where $f\colon \mathbb{R}^d \to \mathbb{R}$, and $\Psi_i\colon \mathbb{R}^n \to \mathbb{R}$ are convex and simple, that is the time complexity T for computing their gradients is linear: $T(\nabla_{\boldsymbol{y}} f(\boldsymbol{y}), \nabla_{\boldsymbol{x}} \Psi(\boldsymbol{x}) = O(d+n)$. This class of functions includes least squares, logistic regression, Lasso, and SVMs (when solved in dual form).

Assuming the matrix is dense, the complexity to compute the full gradient of F is $T(\nabla_{\boldsymbol{x}} F(\boldsymbol{x})) = O(dn)$. If the value $\boldsymbol{w} = A\boldsymbol{x}$ is already computed, one directional derivative can be computed in time $T(\nabla_i F(\boldsymbol{x})) = O(d)$. The recursive update of $\boldsymbol{w}$ after one step needs the addition of one column of matrix A with some factors and can be done in time $O(d)$. However, we note that recursively updating the full gradient vector takes time $O(dn)$ and consequently the computation of the GS rule *cannot* be done efficiently.

Nutini *et al.* [180] consider sparse matrices, for which the computation of the Gauss-Southwell rule becomes traceable. In this paper, we propose an alternative approach. Instead of updating the exact gradient vector, we keep track of an approximation of the gradient vector and recursively update this approximation in time $O(n \log n)$. With these updates, the use of coordinate descent is still justified in case $d = \Omega(n)$.

Algorithm 3 Approximate SCD (ASCD)

Input: f, $\boldsymbol{x}_0$, T, δ-gradient oracle g, method $\mathcal{M}$
Initialize $[\tilde{\boldsymbol{g}}_0]_i = 0$, $[\boldsymbol{r}_0]_i = \infty$ for $i \in [n]$.
for $t = 0$ **to** T **do**
For $i \in [n]$ define *compute u.-and l.-bounds*
$[\boldsymbol{u}_t]_i := \max\{|[\tilde{\boldsymbol{g}}_t]_i - [\boldsymbol{r}_t]_i|, |[\tilde{\boldsymbol{g}}_t]_i + [\boldsymbol{r}_t]_i|\}$
$[\boldsymbol{\ell}_t]_i := \min_{y\in\mathbb{R}}\{|y| \mid [\tilde{\boldsymbol{g}}_t]_i - [\boldsymbol{r}_t]_i \le y \le [\tilde{\boldsymbol{g}}_t]_i + [\boldsymbol{r}_t]_i\}$
$\mathrm{av}(\mathcal{I}) := \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}[\boldsymbol{\ell}_t]_i^2$ *compute active set*
$\mathcal{I}_t := \arg\min_{\mathcal{I}} \left|\left\{\mathcal{I} \subseteq [n] \mid [\boldsymbol{u}_t]_i^2 < \mathrm{av}(\mathcal{I}), \forall i \notin \mathcal{I}\right\}\right|$
Pick $i_t \in_{\text{u.a.r.}} \arg\max_{i\in\mathcal{I}_t}\{[\boldsymbol{\ell}]_i\}$ *active coordinate*
$(\boldsymbol{x}_{t+1}, [\tilde{\boldsymbol{g}}_{t+1}]_{i_t}, [\boldsymbol{r}_{t+1}]_{i_t}) := \mathcal{M}(\boldsymbol{x}_t, \nabla_{i_t} f(\boldsymbol{x}_t))$
$\gamma_t := [\boldsymbol{x}_{t+1}]_{i_t} - [\boldsymbol{x}_t]_{i_t}$ *update $\nabla f(\boldsymbol{x}_{t+1})$ estimate*
Update $[\tilde{\boldsymbol{g}}_{t+1}]_j := [\tilde{\boldsymbol{g}}_t]_j + g_{i_t j}(\boldsymbol{x}_t)$, $j \neq i_t$
Update $[\boldsymbol{r}_{t+1}]_j := [\boldsymbol{r}_t]_j + \gamma_t \delta_{i_t j}$, $j \neq i_t$
end for

B.3 Algorithm

Is it possible to get the significantly improved convergence speed from SCD, when one is only willing to pay the computational cost of only the much simpler UCD? In this section, we give a formal definition of our proposed approximate SCD method which we denote ASCD.

The core idea of the algorithm is the following: While performing coordinate updates, ideally we would like to efficiently track the evolution of *all* elements of the gradient, not only the one coordinate which is updated in the current step. The formal definition of the method is given in Algorithm 5 for smooth objective functions. In each iteration, only one coordinate is modified according to some arbitrary update rule $\mathcal{M}$. The coordinate update rule $\mathcal{M}$ provides two things: First the new iterate $\boldsymbol{x}_{t+1}$, and secondly also an estimate $\tilde{g}$ of the i_t-th entry of the gradient at the new iterate[3]. Formally,

$$(\boldsymbol{x}_{t+1}, \tilde{g}, r) := \mathcal{M}(\boldsymbol{x}_t, \nabla_{i_t} f(\boldsymbol{x}_t)) \tag{B.18}$$

such that the quality of the new gradient estimate $\tilde{g}$ satisfies

$$|\nabla_{i_t} f(\boldsymbol{x}_{t+1}) - \tilde{g}| \le r\,. \tag{B.19}$$

The non-active coordinates are updated with the help of gradient oracles with accuracy $\delta \ge 0$ (see next subsection for details). The scenario of exact updates of all gradient entries is obtained for accuracy parameters $\delta = r = 0$ and in this case ASCD is identical

[3] For instance, for updates by exact coordinate optimization (line-search), we have $\tilde{g} = r = 0$.

to SCD.

B.3.1 Safe bounds for gradient evolution

ASCD maintains lower and upper bounds for the absolute values of each component of the gradient ($[\boldsymbol{\ell}]_i \leq |\nabla_i f(\boldsymbol{x})| \leq [\boldsymbol{u}]_i$). These bounds allow to identify the coordinates on which the absolute values of the gradient are small (and hence cannot be the steepest one). More precisely, the algorithm maintains a set $\mathcal{I}_t$ of active coordinates (similar in spirit as in active set methods, see e.g. Kim and Park [113], Wen *et al.* [261]). A coordinate j is excluded from $\mathcal{I}_t$ if the estimated progress in this direction (cf. (B.5)) is lower than the average of the estimated progress along coordinate directions in $\mathcal{I}_t$, $[\boldsymbol{u}_t]_j^2 < \frac{1}{|\mathcal{I}_t|}\sum_{i\in\mathcal{I}_t}[\boldsymbol{\ell}_t]_i^2$. The active set $\mathcal{I}_t$ can be computed in $O(n\log n)$ time by sorting. All other operations take linear $O(n)$ time.

Gradient Oracle. The selection mechanism in ASCD crucially relies on the following definition of a δ-gradient oracle. While the update $\mathcal{M}$ delivers the estimated active entry of the new gradient, the additional gradient oracle is used to update all other coordinates $j \neq i_t$ of the gradient; as in the last two lines of Algorithm 5.

Definition B.3.1.1 (δ-gradient oracle)**.** For a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and indices $i, j \in [n]$, a (i,j)-gradient oracle with error $\delta_{ij} \geq 0$ is a function $g_{ij}\colon \mathbb{R}^n \to \mathbb{R}$ satisfying $\forall \boldsymbol{x} \in \mathbb{R}^n, \forall \gamma \in \mathbb{R}$:

$$\left|\nabla_j f(\boldsymbol{x}+\gamma\boldsymbol{e}_i) - g_{ij}(\boldsymbol{x})\right| \leq |\gamma|\,\delta_{ij}\,. \tag{B.20}$$

We denote by a δ-gradient oracle a family $\{g_{ij}\}_{i,j\in[n]}$ of δ_{ij}-gradient oracles.

We discuss the availability of good gradient oracles for many problem classes in more detail in Section B.4. For example for least squares problems and general linear models, a δ-gradient oracle is for instance given by a scalar product estimator as in (B.24) below. Note that ASCD can also handle very bad estimates, as long as the property (B.20) is satisfied (possibly even with accuracy $\delta_{ij} = \infty$).

Initialization. In ASCD the initial estimate $\tilde{\boldsymbol{g}}_0$ of the gradient is just arbitrarily set to $\mathbf{0}$, with uncertainty $\boldsymbol{r}_0 = \infty$. Hence in the worst case it takes $\Theta(n\log n)$ iterations until each coordinate gets picked at least once (cf. Dawkins [53]) and until corresponding gradient estimates are set to a realistic value. If better estimates of the initial gradient are known, they can be used for the initialization as long as a strong error bound as in (B.19) is known as well. For instance the initialization can be done with $\nabla f(\boldsymbol{x}_0)$ if one is willing to compute this vector in one batch pass.

Convergence Rate Guarantee. We present our first main result showing that the performance of ASCD is provably between UCD and SCD. First observe that if in Algorithm 5

the gradient oracle is always exact, i.e. $\delta_{ij} \equiv 0$, and if $\tilde{\boldsymbol{g}}_0$ is initialized with $\nabla f(\boldsymbol{x}_0)$, then in each iteration $|\nabla_{i_t} f(\boldsymbol{x}_t)| = \|\nabla f(\boldsymbol{x}_t)\|_\infty$ and ASCD identical to SCD.

Lemma B.3.1.1. *Let $i_{\max} := \arg\max_{i\in[n]} |\nabla_i f(\boldsymbol{x}_t)|$. Then $i_{\max} \in \mathcal{I}_t$, for $\mathcal{I}_t$ as in Algorithm 5.*

Proof. This is immediate from the definitions of $\mathcal{I}_t$ and the upper and lower bounds. Suppose $i_{\max} \notin \mathcal{I}_t$, then there exists $j \neq i_{\max}$ such that $[\ell_t]_j > [u_t]_{i_{\max}}$, and consequently $\left|\nabla_j f(\boldsymbol{x}_t)\right| > |\nabla_{i_{\max}} f(\boldsymbol{x}_t)|$. □

Theorem B.3.1.2. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be convex and coordinate-wise L-smooth, let $\tau_{\text{UCD}}, \tau_{\text{SCD}}, \tau_{\text{ASCD}}$ denote the expected one step progress (B.6) of UCD, SCD and ASCD, respectively, and suppose all methods use the same step-size rule $\mathcal{M}$. Then*

$$\tau_{\text{UCD}}(\boldsymbol{x}) \leq \tau_{\text{ASCD}}(\boldsymbol{x}) \leq \tau_{\text{SCD}}(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{B.21}$$

Proof. By (B.5) we get $\tau_{\text{ASCD}}(\boldsymbol{x}) = \frac{1}{2L|\mathcal{I}|}\sum_{i\in\mathcal{I}} |\nabla_i f(\boldsymbol{x})|^2$, where $\mathcal{I}$ denotes the corresponding index set of ASCD when at iterate $\boldsymbol{x}$. Note that for $j \notin \mathcal{I}$ it must hold that $\left|\nabla_j f(\boldsymbol{x})\right|^2 \leq [\boldsymbol{u}]_j^2 < \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} [\boldsymbol{\ell}]_i^2 \leq \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} |\nabla_i f(\boldsymbol{x})|^2$ by definition of $\mathcal{I}$. □

Observe that the above theorem holds for all gradient oracles and coordinate update variants, as long as they are used with corresponding quality parameters r (as in (B.19)) and δ_{ij} (as in (B.20)) as part of the algorithm.

Heuristic variants. Below also propose three heuristic variants of ASCD. For all these variants the active set $\mathcal{I}_t$ can be computed $O(n)$, but the statement of Theorem B.3.1.2 does not apply. These variants only differ from ASCD in the choice of the active set in Algorithm 5:

- u-ASCD: $\mathcal{I}_t := \arg\max_{i\in[n]} [\boldsymbol{u}_t]_i$
- ℓ-ASCD: $\mathcal{I}_t := \arg\max_{i\in[n]} [\boldsymbol{\ell}_t]_i$
- a-ASCD: $\mathcal{I}_t := \left\{ i \in [n] \mid [\boldsymbol{u}_t]_i \geq \max_{i\in[n]} [\boldsymbol{\ell}_t]_i \right\}$

B.4 Approximate Gradient Update

In this section we argue that for a large class of objective functions of interest in machine learning, the change in the gradient along every coordinate direction can be estimated efficiently.

Lemma B.4.0.1. *Consider $F\colon \mathbb{R}^n \to \mathbb{R}$ as in (B.17) with twice-differentiable $f\colon \mathbb{R}^d \to \mathbb{R}$. Then for two iterates $\boldsymbol{x}_t, \boldsymbol{x}_{t+1} \in \mathbb{R}^n$ of a coordinate descent algorithm, i.e. $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t \boldsymbol{e}_{i_t}$, there exists a $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ on the line segment between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$, $\tilde{\boldsymbol{x}} \in [\boldsymbol{x}_t, \boldsymbol{x}_{t+1}]$ with*

$$\nabla_i F(\boldsymbol{x}_{t+1}) - \nabla_i F(\boldsymbol{x}_t) = \gamma_t \langle \boldsymbol{a}_i, \nabla^2 f(A\tilde{\boldsymbol{x}}) \boldsymbol{a}_{i_t} \rangle \quad \forall i \neq i_t \tag{B.22}$$

where $\boldsymbol{a}_i$ denotes the i-th column of the matrix A.

Proof. For coordinates $i \neq i_t$ the gradient (or subgradient set) of $\Psi_i([\boldsymbol{x}_t]_i)$ does not change. Hence it suffices to calculate the change $\nabla f(\boldsymbol{x}_{t+1}) - \nabla f(\boldsymbol{x}_t)$. This is detailed in the appendix. □

Least-Squares with Arbitrary Regularizers. The least squares problem is defined as problem (B.17) with $f(A\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2$ for a $\boldsymbol{b} \in \mathbb{R}^d$. This function is twice differentiable with $\nabla^2 f(A\boldsymbol{x}) = I_n$. Hence (B.22) reduces to

$$\nabla_i F(\boldsymbol{x}_{t+1}) - \nabla_i F(\boldsymbol{x}_t) = \gamma_t \langle \boldsymbol{a}_i, \boldsymbol{a}_{i_t} \rangle \quad \forall i \neq i_t \,. \tag{B.23}$$

This formulation gives rise to various gradient oracles (B.20) for the least square problems. For for $i \neq i_t$ we easily verify that the condition (B.20) is satisfied:

1. $g^1_{ij} := \langle \boldsymbol{a}_i, \boldsymbol{a}_{i_t} \rangle$; $\delta_{ij} = 0$,
2. $g^2_{ij} := \max\left\{-\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|, \min\left\{S(i,j), \|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|\right\}\right\}$; $\delta_{ij} = \varepsilon\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|$, where $S\colon [n] \times [n]$ denotes a function with the property

$$\left|S(i,j) - \langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle\right| \leq \varepsilon \|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\| \,, \quad \forall i,j \in [n] \tag{B.24}$$

3. $g^3_{ij} := 0$; $\delta_{ij} = \|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|$,
4. $g^4_{ij} \in_{\text{u.a.r.}} [-\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|, \|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|]$; $\delta_{ij} = \|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|$.

Oracle g^1 can be used in the rare cases where the dot product matrix is accessible to the optimization algorithm without any extra cost. In this case the updates will all be exact. If this matrix is not available, then the computation of each scalar product takes time $O(d)$. Hence, they cannot be recomputed on the fly, as argued in Section B.2.4. In contrast, the oracles g^3 and g^4 are extremely cheap to compute, but the error bounds are worse. In the numerical experiments in Section B.7 we demonstrate that these oracles perform surprisingly well.

The oracle g^2 can for instance be realized by low-dimensional embeddings, such as given by the Johnson-Lindenstrauss lemma (cf. Achlioptas [1], Matoušek [151]). By embedding each vector in a lower-dimensional space of dimension $O\left(\varepsilon^{-2}\log n\right)$ and computing the scalar products of the embedding in time $O(\log n)$, relation (B.24) is satisfied.

Updating the gradient of the active coordinate. So far we only discussed the update of the passive coordinates. For the active coordinate the best strategy depends on the update rule $\mathcal{M}$ from (B.18). If exact line search is used, then $0 \in \nabla_{i_t} f(\boldsymbol{x}_{t+1})$. For other update rules we can update the gradient $\nabla_{i_t} f(\boldsymbol{x}_{t+1})$ with the same gradient oracles as for the other coordinates, however we need also to take into account the change of the

gradient of $\Psi_i([\boldsymbol{x}_t]_i)$. If Ψ_i is simple, like for instance in ridge or lasso, the subgradients at the new point can be computed efficiently.

Bounded variation. In many applications the Hessian $\nabla^2 f(A\tilde{\boldsymbol{x}})$ is not so simple as in the case of square loss. If we assume that the Hessian of f is bounded, i.e. $\nabla^2 f(A\boldsymbol{x}) \preceq M \cdot I_n$ for a constant $M \geq 0$, $\forall \boldsymbol{x} \in \mathbb{R}^n$, then it is easy to see that the following holds :

$$-M\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\| \leq \langle \boldsymbol{a}_i, \nabla^2 f(A\tilde{\boldsymbol{x}})\boldsymbol{a}_{i_t}\rangle \leq M\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|\,.$$

Using this relation, we can define gradient oracles for more general functions, by taking the additional approximation factor M into account. The quality can be improved, if we have access to local bounds on $\nabla^2 f(A\boldsymbol{x})$.

Heuristic variants. By design, ASCD is robust to high errors in the gradient estimations – the steepest descent direction is always contained in the active set. However, instead of using only the very crude oracle g^4 to approximate *all* scalar products, it might be advantageous to compute some scalar products with higher precision. We propose to use a caching technique to compute the scalar products with high precision for all vectors in the active set (and storing a matrix of size $O(\mathcal{I}_t \times n)$). This presumably works well if the active set does not change much over time.

B.5 Extension to Composite Functions

The key ingredients of ASCD are the coordinate-wise upper and lower bounds on the gradient and the definition of the active set $\mathcal{I}_t$ which ensures that the steepest descent direction is always kept and that only provably bad directions are removed from the active set. These ideas can also be generalized to the setting of composite functions (B.2). We already discussed some popular GS-$*$ update rules in the introduction in Section B.2.3.

Implementing ASCD for the GS-s rule is straight forward, and we comment on the GS-r in the appendix in Sec. B.12.2. Here we exemplary detail the modification for the GS-q rule (B.16), which turns out to be the most evolved (the same reasoning also applies to the GSL-q rule from [180]). In Algo. 4 we show the construction — based just on approximations of the gradient of the smooth part f — of the active set $\mathcal{I}$. For this we compute upper and lower bounds $\boldsymbol{v}, \boldsymbol{w}$ on $\min_{y\in\mathbb{R}} V(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x}))$, such that

$$[\boldsymbol{v}]_i \leq \min_{y\in\mathbb{R}} V(\boldsymbol{x}, y, \nabla_i f(x) \leq [\boldsymbol{w}]_i \quad \forall i \in [n]\,. \tag{B.25}$$

The selection of the active coordinate is then based on these bounds. Similar as in Lemma B.3.1.1 and Theorem B.3.1.2 this set has the property $i_{\mathrm{GS-q}} \in \mathcal{I}$, and directions are only discarded in such a way that the efficiency of ASCD-q cannot drop below the efficiency of UCD. The proof can be found in the appendix in Section B.12.1.

Algorithm 4 Adaptation of ASCD for GS-q rule

Input: Gradient estimate $\tilde{\boldsymbol{g}}$, error bounds $\boldsymbol{r}$.
For $i \in [n]$ define: *compute u.-and l.-bounds*
$[\boldsymbol{u}]_i := [\tilde{\boldsymbol{g}}]_i + [\boldsymbol{r}]_i$, $[\boldsymbol{\ell}]_i := [\tilde{\boldsymbol{g}}]_i - [\boldsymbol{r}]_i$
$[\boldsymbol{u}^\star]_i := \arg\min_{y\in\mathbb{R}} V(\boldsymbol{x}, y, [\boldsymbol{u}]_i)$ *minimize the model*
$[\boldsymbol{\ell}^\star]_i := \arg\min_{y\in\mathbb{R}} V(\boldsymbol{x}, y, [\boldsymbol{\ell}]_i)$
compute u.-and l. bounds on $\min_{y\in\mathbb{R}} V(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x}))$
$[\boldsymbol{\omega}_u]_i := V(\boldsymbol{x}, [\boldsymbol{u}^\star]_i, [\boldsymbol{u}]_i) + \max\{0, [\boldsymbol{u}^\star]_i([\boldsymbol{\ell}]_i - [\boldsymbol{u}]_i)\}$
$[\boldsymbol{\omega}_\ell]_i := V(\boldsymbol{x}, [\boldsymbol{\ell}^\star]_i, [\boldsymbol{\ell}]_i) + \max\{0, [\boldsymbol{\ell}^\star]_i([\boldsymbol{u}]_i - [\boldsymbol{\ell}]_i)\}$
$[\boldsymbol{v}]_i := \min\{V(\boldsymbol{x}, [\boldsymbol{u}^\star]_i, [\boldsymbol{u}]_i), V(\boldsymbol{x}, [\boldsymbol{\ell}^\star]_i, [\boldsymbol{\ell}]_i)\}$
$[\boldsymbol{w}]_i := \min\{[\boldsymbol{\omega}_u]_i, [\boldsymbol{\omega}_\ell]_i, \Psi_i([\boldsymbol{x}]_i)\}$
$\mathrm{av}(\mathcal{I}) := \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}[\boldsymbol{w}]_i$ *compute active set*
$\mathcal{I}_t := \arg\min_{\mathcal{I}} |\{\mathcal{I} \subseteq [n] \mid [\boldsymbol{v}]_i > \mathrm{av}(\mathcal{I}), \forall i \notin \mathcal{I}\}|$

B.6 Analysis of Competitive Ratio

In Section B.3 we derived in Thm. B.3.1.2 that the one step progress of ASCD is between the bounds on the onestep progress of UCD and SCD. However, we know that the efficiency of the latter two methods can differ much, up to a factor of n. In this section we will argue that in certain cases where SCD performs much better than UCD, ASCD will accelerate as well. To measure this effect, we could for instance consider the ratio:

$$\rho_t := \frac{\left|\left\{i \in \mathcal{I}_t \mid |\nabla_i f(\boldsymbol{x}_t)| \geq \frac{1}{2}\|\nabla f(\boldsymbol{x}_t)\|_\infty\right\}\right|}{|\mathcal{I}_t|}, \tag{B.26}$$

For general functions this expression is a bit cumbersome to study, therefore we restrict our discussion to the class of objective functions (B.11) as introduced in Sec. B.2.2. Of course not all real-world objective functions will fall into this class, however this problem class is still very interesting in our study, as we will see in the following, because it will highlight the ability (or disability) of the algorithms to eventually identify the right set of 'active' coordinates.

For the functions with the structure (B.11) (and q as in Thm. B.2.2.1), the active set falls into the first s coordinates. Hence it is reasonable to approximate ρ_t by the competitive ratio

$$\rho_t := \frac{|\mathcal{I}_t \cap [s]|}{|\mathcal{I}_t|}. \tag{B.27}$$

It is also reasonable to assume that in the limit, $(t \to \infty)$, a constant fraction of the $[s]$ will be contained in the active set $\mathcal{I}_t$ (it might not hold $[s] \subseteq \mathcal{I}_t\ \forall t$, as for instance with exact line search the directional derivative vanishes just after the update). In the following

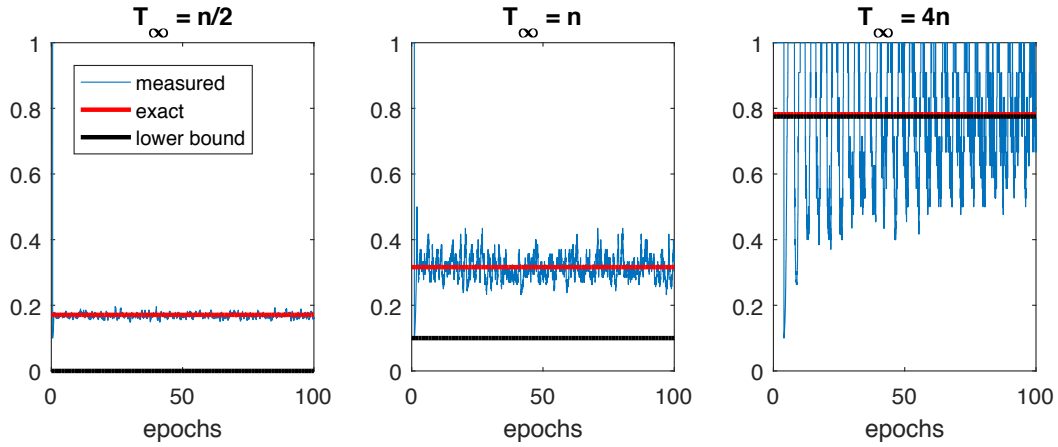


Figure B.1: Competitive ratio ρ_t (*blue*) in comparison with ρ_∞ (B.28) (*red*) and the lower bound $\rho_\infty \geq 1 - \frac{n-s}{T_\infty}$ (*black*). Simulation for parameters $n = 100$, $s = 10$, $c = 1$ and $T_\infty \in \{50, 100, 400\}$.

theorem we calculate ρ_t for $(t \to \infty)$, the proof is given in the appendix.

Theorem B.6.0.1. *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be of the form (B.11). For indices $i \notin [s]$ define $\mathcal{K}_i := \{t \mid i \notin \mathcal{I}_t, i \in \mathcal{I}_{t-1}\}$. For $j \in \mathcal{K}_i$ define $T_j^i := \min\{t - j \mid i \in \mathcal{I}_{j+t}\}$, i.e. the number of iterations outside the active set, $T_\infty^i := \lim_{t\to\infty} \mathbb{E}_{j\in\mathcal{K}_i}\left[T_j^i \mid j > k\right]$, and the average $T_\infty := \mathbb{E}_{i\notin[s]}\left[T_\infty^i\right]$. If there exists a constant $c > 0$ such that $\lim_{t\to\infty} |[s] \cap \mathcal{I}_t| = cs$, then (with the notation $\rho_\infty := \lim_{t\to\infty} \mathbb{E}[\rho_t]$),*

$$\rho_\infty \geq \frac{2cs}{cs + n - s - T_\infty + \sqrt{\theta}}, \tag{B.28}$$

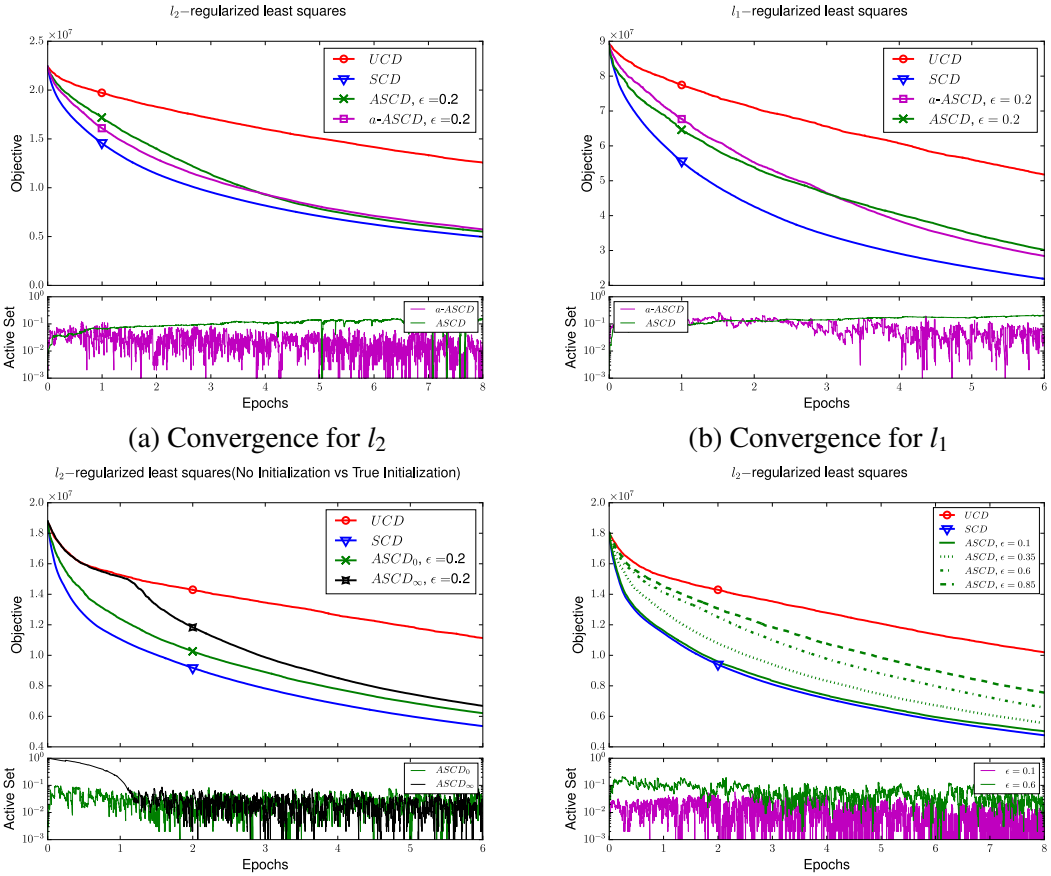
where $\theta \equiv \theta := n^2 + (c-1)^2 s^2 + 2n((c-1)s - T_\infty) + 2(1+c)sT_\infty + T_\infty^2$. Especially, $\rho_\infty \geq 1 - \frac{n-s}{T_\infty}$.

In Figure B.1 we compare the lower bound (B.28) of the competitive ratio in the limit $(t \to \infty)$ with actual measurements of ρ_t for simulated example with parameters $n = 100$, $s = 10$, $c = 1$ and various $T_\infty \in \{50, 100, 400\}$. We initialized the active set $\mathcal{I}_0 = [s]$, but we see that the equilibrium is reached quickly.

B.6.1 Estimates of the competitive ratio

Based on this Thm. B.6.0.1, we can now estimate the competitive ratio in various scenarios. On the class (B.11) it holds $c \approx 1$ as we argued before. Hence the competitive ratio (B.28) just depends on T_∞. This quantity measures how many iterations a coordinate $j \notin [s]$ is in average outside of the active set $\mathcal{I}_t$. From the lower bound we see that the competitive ratio ρ_t approaches a constant for $(t \to \infty)$ if $T_\infty = \Theta(n)$, for instance $\rho_\infty \geq 0.8$ if $T_\infty \geq 5n$.

As an approximation to T_∞, we estimate the quantities $T_{t_0}^j$ defined in Thm. B.6.0.1. $T_{t_0}^j$



(a) Convergence for l_2

(b) Convergence for l_1

(c) True vs No Initialization for l_2

(d) Error Variation (ASCD)

Figure B.2: Experimental results on synthetically generated datasets

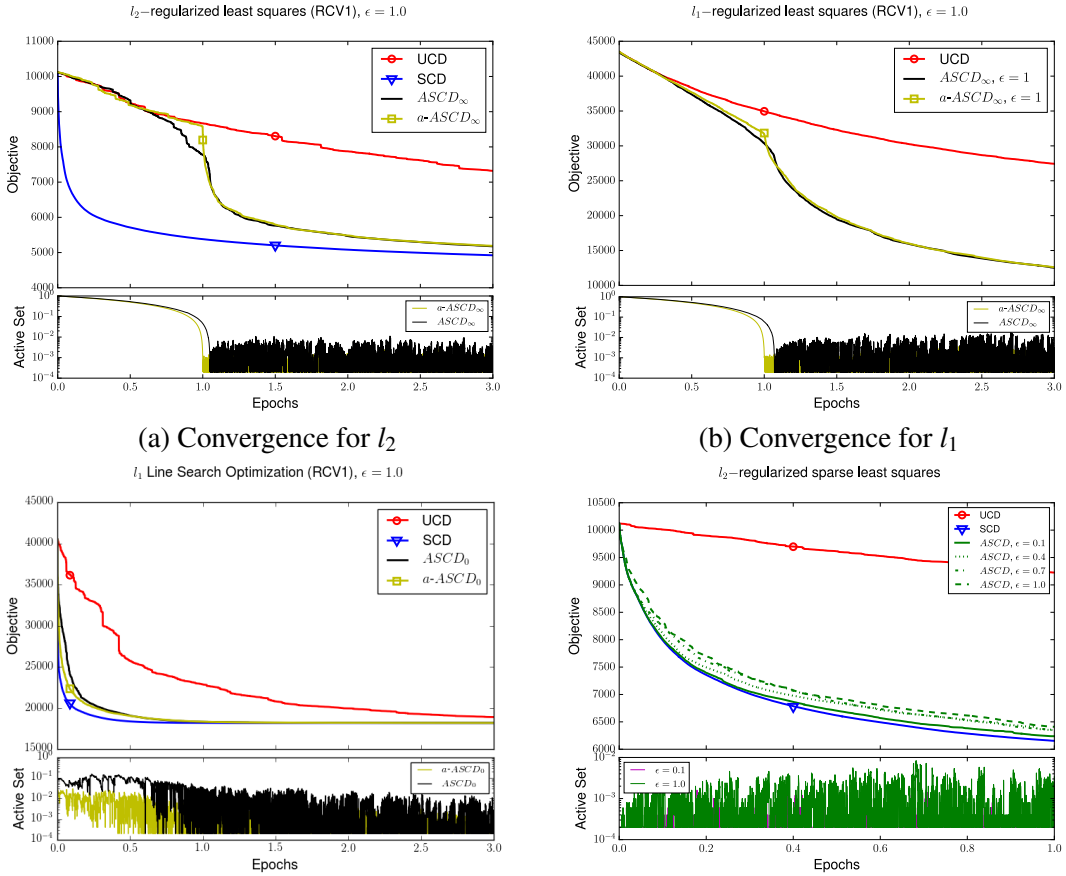
denotes the number of iterations it takes until coordinate j enters the active set again, assuming it left the active set at iteration $t_0 - 1$. We estimate $T_{t_0}^j \geq \hat{T}$, where $\hat{T}$ denotes maximum number of iterations such that

$$\sum_{t=t_0}^{t_0+\hat{T}} \gamma_t \delta_{i_t j} \leq \frac{1}{s} \sum_{k=1}^{s} \left| \nabla_k f\left(\boldsymbol{x}_{t_0+\hat{T}}\right) \right| \quad \forall j \notin [s]. \tag{B.29}$$

For smooth functions, the steps $\gamma_t = \Theta(|\nabla_{i_t} f(\boldsymbol{x}_t)|)$ and if we additionally assume that the errors of the gradient oracle are uniformly bounded $\delta_{ij} \leq \delta$, the sum in (B.29) simplifies to $\delta \sum_{t=t_0}^{t_0+\hat{T}} |\nabla_{i_t} f(\boldsymbol{x}_t)|$.

For smooth, but not strongly convex function q, the norms of the gradient changes very slowly, with a rate independent of s or n, and we get $\hat{T} = \Theta\left(\frac{1}{\delta}\right)$. Hence, the competitive ratio is constant for $\delta = \Theta\left(\frac{1}{n}\right)$.

For strongly convex function q, the norm of the gradient decreases linearly, say $\|\nabla f(\boldsymbol{x}_t)\|_2^2 \propto e^{\kappa t}$ for $\kappa \approx \frac{1}{s}$. I.e. it decreases by half after each $\Theta(s)$ iterations. Therefore



(a) Convergence for l_2

(b) Convergence for l_1

(c) Line search for l_1

(d) Error Variation (ASCD)

Figure B.3: Experimental results on the RCV1-binary dataset

to guarantee $\hat{T} = \Theta(n)$ it needs to hold $\delta = e^{-\Theta\left(\frac{n}{s}\right)}$. This result seems to indicate that the use of ACDM is only justified if s is large, for instance $s \geq \frac{1}{4}n$. Otherwise the convergence on q is too fast, and the gradient approximations are too weak. However, notice that we assumed δ to be an uniform bound on all errors. If the errors have large discrepancy the estimates become much better (this holds for instance on datasets where the norm data vectors differs much, or when caching techniques as mentioned in Sec. B.4 are employed).

B.7 Empirical Observations

In this section we evaluate the empirical performance of ASCD on synthetic and real datasets. We consider the following regularized general linear models:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \tfrac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \tfrac{\lambda}{2}\|\boldsymbol{x}\|_2^2, \tag{B.30}$$

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \tfrac{1}{2}\|A\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1\,, \tag{B.31}$$

that is, l_2-regularized least squares (B.30) as well as l_1-regularized linear regression (Lasso) in (B.31), respectively.

Datasets. The datasets $A \in \mathbb{R}^{d\times n}$ in problems (B.30) and (B.31) were chosen as follows for our experiments. For the synthetic data, we follow the same generation procedure as described in [180], which generates very sparse data matrices. For completeness, full details of the data generation process are also provided in the appendix in Sec. B.13. For the synthetic data we choose $n = 5000$ for problem (B.31) and $n = 1000$ for problem (B.30). Dimension $d = 1000$ is fixed for both cases.
For real datasets, we perform the experimental evaluation on RCV1 (binary,training), which consists of $20,242$ samples, each of dimension $47,236$ [135]. We use the unnormalized version with all non-zeros values set to 1 (bag-of-words features).

Gradient oracles and implementation details. On the RCV1 dataset, we approximate the scalar products with the oracle g^4 that was introduced in Sec. B.4. This oracle is extremely cheap to compute, as the norms $\|\boldsymbol{a}_i\|$ of the columns of A only need to be computed once.
On the synthetic data, we simulate the oracle g^2 for various precisions values ε. For this, we sample a value uniformly at random from the allowed error interval (B.24). Figs. B.2d and B.3d show the convergence for different accuracies.
For the l_1-regularized problems, we used ASCD with the GS-s rule (the experiments in [180] revealed almost identical performance of the different GS-$*$ rules).
We compare the performance of UCD, SCD and ASCD. We also implement the heuristic version a-ASCD that was introduced in Sec. B.3. All algorithm variants use the same step size rule (i.e. the method $\mathcal{M}$ in Algorithm 5). We use exact line search for the experiment in Fig. B.3c, for all others we used a fixed step size rule (the convergence is slower for all algorithms, but the different effects of the selection of the active coordinate is more distinctly visible).
ASCD is either initialized with the true gradient (Figs. B.2a, B.2b, B.2d, B.3c, B.3d) or arbitrarely (with error bounds $\delta = \infty$) in Figs. B.3a and B.3b (Fig. B.2c compares both initializations).
Fig. B.2 shows results on the synthetic data, Fig. B.3 on the RCV1 dataset. All plots show also the size of the active set $\mathcal{I}_t$. The plots B.3c and B.3d are generated on a subspace of RCV1, with 10000 and 5000 randomly chosen columns, respectively.

Here are the highlights of our experimental study:

1. **No initialization needed.** We observe (see e.g. Figs. B.2c,B.3a, B.3b) that initialization with the true gradient values is *not* needed at beginning of the optimization process (the cost of the initialization being as expensive as one epoch of ASCD). Instead, the algorithm performs strong in terms of learning the active set on its own, and the set

converges very fast after just one epoch.

2. **High errors toleration.** The gradient oracle g^4 gives very crude approximations, however the convergence of ASCD is excellent on RCV1 (Fig. B.3). Here the size of the true active set is very small (in the order of 0.1% on RCV1) and ASCD is able to identify this set. Fig. B.3d shows that almost nothing can be gained from more precise (and more expensive) oracles.
3. **Heuristic a-ASCD performs well.** The convergence behavior of ASCD follows theory. For the heuristic version a-ASCD (which computes the active set slightly faster, but Thm. B.3.1.2 does not hold) performs identical to ASCD in practice (cf. Figs. B.2, B.3), and sometimes slightly better. This is explained by the active set used in ASCD typically being larger than the active set of a-ASCD (Figs. B.2a, B.2b, B.3a, B.3b).

B.8 Concluding Remarks

We proposed ASCD, a novel selection mechanism for the active coordinate in CD methods. Our scheme enjoys three favorable properties: (i) its performance can reach the performance steepest CD — both in theory and practice, (ii) the performance is never worse than uniform CD, (iii) in many important applications, the scheme it can be implemented at no extra cost per iteration.

ASCD calculates the active set in a safe manner, and picks the active coordinate uniformly at random from this smaller set. It seems possible that an adaptive sampling strategy on the active set could boost the performance even further. Here we only study CD methods where a single coordinate gets updated in each iteration. ASCD can immediately also be generalized to block-coordinate descent methods. However, the exact implementation in a distributed setting can be challenging.

Finally, it is an interesting direction to extend ASCD also to the stochastic gradient descent setting (not only heuristically, but with the same strong guarantees as derived in this paper).

Proofs for Main Results

B.9 On Steepest Coordinate Descent

B.9.1 Convergence on Smooth Functions

Lemma B.9.1.1 (Lower bound on the one step progress on smooth functions)**.** *Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be convex and coordinate-wise L-smooth. For a sequence of iterates $\{\boldsymbol{x}_t\}_{t\geq 0}$ define the progress measure*

$$\Delta(\boldsymbol{x}_t) := \frac{1}{\mathbb{E}\left[f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}^\star) \mid \boldsymbol{x}_t\right]} - \frac{1}{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}\,. \tag{B.32}$$

For sequences $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD it holds:

$$\Delta_{\mathrm{SCD}}(\boldsymbol{x}_t) \geq \frac{1}{2L\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1^2}\,, \qquad t \geq 0\,, \tag{B.33}$$

and for a sequences generated by UCD:

$$\Delta_{\mathrm{UCD}}(\boldsymbol{x}_t) \geq \frac{1}{2nL\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_2^2}\,, \qquad t \geq 0\,. \tag{B.34}$$

It is important to note that the lower bounds presented in Equations (B.33) and (B.34) are quite tight and equality is almost achievable under special conditions. When comparing the per-step progress of these two methods, we find — similarly as in (B.7) — the relation

$$\frac{1}{n}\Delta_{\mathrm{SCD}}(\boldsymbol{x}_t) \leq \Delta_{\mathrm{UCD}}(\boldsymbol{x}_t) \leq \Delta_{\mathrm{SCD}}(\boldsymbol{x}_t)\,, \tag{B.35}$$

that is, SCD can boost the performance over the random coordinate descent up to the factor of n. This also holds for a sequence of consecutive updates, as show in Theorem B.2.1.1.

***Proof of Lemma** B.9.1.1.* Define $f^\star := f(\boldsymbol{x}^\star)$. From the smoothness assumption (C.1), we get

$$\begin{aligned} f(\boldsymbol{x}_{t+1}) &\overset{(B.5)}{\leq} f(\boldsymbol{x}_t) - \frac{1}{2L}\|\nabla f(\boldsymbol{x}_t)\|_\infty^2 \\ \Rightarrow \left(f(\boldsymbol{x}_{t+1}) - f^\star\right) &\leq \left(f(\boldsymbol{x}_t) - f^\star\right) - \frac{1}{2L}\|\nabla f(\boldsymbol{x}_t)\|_\infty^2 \end{aligned} \tag{B.36}$$

Now from the property of a convex function and Hölder's inequality:

$$f(\boldsymbol{x}_t) - f^\star \leq \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}^\star \rangle \leq \|\nabla f(\boldsymbol{x}_t)\|_\infty \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1 \tag{B.37}$$

Hence,

$$\begin{aligned} \big(f(\boldsymbol{x}_t) - f^\star\big)^2 &\leq \|\nabla f(\boldsymbol{x}_t)\|_\infty^2 \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1^2 \\ \Rightarrow \|\nabla f(\boldsymbol{x}_t)\|_\infty^2 &\geq \frac{\big(f(\boldsymbol{x}_t) - f^\star\big)^2}{\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1^2} \end{aligned} \tag{B.38}$$

From Equations (B.36) and (B.38),

$$\frac{1}{\big(f(\boldsymbol{x}_{t+1}) - f^\star\big)} - \frac{1}{\big(f(\boldsymbol{x}_t) - f^\star\big)} \geq \frac{1}{2L\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1^2} \tag{B.39}$$

Which concludes the proof. □

We like to remark, that the one step progress for UCD can be written as [170, 262]:

$$\frac{1}{\big(\mathbb{E}[f(\boldsymbol{x}_{t+1})|\boldsymbol{x}_t] - f^\star\big)} - \frac{1}{\big(f(\boldsymbol{x}_t) - f^\star\big)} \geq \frac{1}{2Ln\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_2^2} \tag{B.40}$$

***Proof of Theorem** B.2.1.1.* From Lemma B.9.1.1,

$$\frac{1}{\big(f(\boldsymbol{x}_{t+1}) - f^\star\big)} - \frac{1}{\big(f(\boldsymbol{x}_t) - f^\star\big)} \geq \frac{1}{2L\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1^2}$$

Now summing up the above equation for $t = 0$ till $t - 1$, we get:

$$\begin{aligned} \frac{1}{\big(f(\boldsymbol{x}_t) - f^\star\big)} - \frac{1}{\big(f(\boldsymbol{x}_0) - f^\star\big)} &\geq \frac{1}{2L} \sum_{i=0}^{t-1} \frac{1}{\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|_1^2} \\ \Rightarrow \frac{1}{\big(f(\boldsymbol{x}_t) - f^\star\big)} &\geq \frac{1}{2L} \sum_{i=0}^{t-1} \frac{1}{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_1^2} \\ \Rightarrow \frac{1}{\big(f(\boldsymbol{x}_t) - f^\star\big)} &\geq \frac{t}{2LR_1^2} \\ \Rightarrow f(\boldsymbol{x}_t) - f^\star &\leq \frac{2LR_1^2}{t} \end{aligned}$$

Which concludes the proof. □

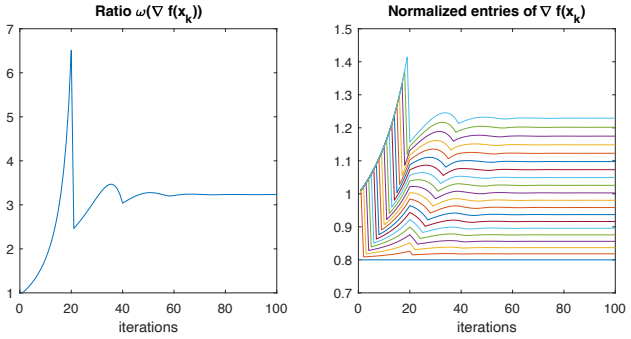


Figure B.4: SCD on the function from Theorem B.9.2.1 in dimension $n = 20$ with $\boldsymbol{x}_0 = \mathbf{1}_n$ (i.e. not the worst starting point constructed in the proof of Theorem B.9.2.1). On the right the (normalized and sorted) components of $\nabla f(\boldsymbol{x}_t)$.

B.9.2 Lower bounds

In this section we provide the proof of Theorem B.2.2.1. Our result is slightly more general, we will proof the following (and Theorem B.2.2.1 follows by the choice $\alpha = 0.01 < \frac{1}{3}$).

Theorem B.9.2.1. *Consider the function $q(\boldsymbol{x}) = \frac{1}{2}\langle Q\boldsymbol{x}, \boldsymbol{x}\rangle$ for $Q := (\alpha - 1)\frac{1}{n}J_n + I_n$, where $J_n = \mathbf{1}_n\mathbf{1}_n^T$ and $0 < \alpha < \frac{1}{2}$, $n > 2$. Then there exists $\boldsymbol{x}_0 \in R^n$ such that for the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD it holds*

$$\|\nabla q(\boldsymbol{x}_t)\|_\infty^2 \leq \frac{3+3\alpha}{n}\|\nabla q(\boldsymbol{x}_t)\|_2^2. \tag{B.41}$$

In the proof below we will construct a special $\boldsymbol{x}_0 \in R^n$ that has the claimed property. However, we would like to remark that this is not very crucial. We observe that for functions as in Theorem B.9.2.1 almost any initial iterate ($\boldsymbol{x}$ not aligned with the coordinate axes) the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ of iterates generated by SCD suffers from the same issue, i.e. relation (B.41) holds for iteration counter t sufficiently large. We do not prove this formally, but demonstrate this behavior in Figure B.4. We see that the steady state is almost reached after $2n$ iterations.

Proof of Theorem B.9.2.1. Define the parameter c_α by the equation

$$\left(1 + \frac{\alpha - 1}{n}\right) c_\alpha^{n-1} = \left(\frac{1-\alpha}{n}\right) S_{n-1}(c_\alpha) \tag{B.42}$$

$$c_\alpha^{n-1} = \left(\frac{1-\alpha}{n}\right) S_n(c_\alpha) \tag{B.43}$$

where $S_n(c_\alpha) = \sum_{i=0}^{n-1} c_\alpha^n$; and define $\boldsymbol{x}_0$ as $[\boldsymbol{x}_0]_i = c_\alpha^{i-1}$ for $i = 1, \ldots, n$. In Lemma B.9.2.2 below we show that $c_\alpha \geq 1 - \frac{3}{n}\alpha$.

We now show that SCD cycles through the coordinates, i.e. the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by SCD satisfies

$$[\boldsymbol{x}_t]_{1+(t-1 \mod n)} = c_\alpha^n \cdot [\boldsymbol{x}_{t-1}]_{1+(t-1 \mod n)} \,. \tag{B.44}$$

Observe $\nabla f(\boldsymbol{x}_0) = Q\boldsymbol{x}_0$. Hence the GS rule picks $i_1 = 1$ in the first iteration. The iterate is updated as follows:

$$\begin{aligned}
[\boldsymbol{x}_1]_1 &\overset{(B.3)}{=} [\boldsymbol{x}_0]_1 - \frac{[Q\boldsymbol{x}_0]_1}{Q_{11}} && \text{(B.45)}\\
&= 1 - \frac{(\alpha-1)\frac{1}{n}S_n(c_\alpha)+1}{(\alpha-1)\frac{1}{n}+1} && \text{(B.46)}\\
&= \frac{(\alpha-1)\frac{1}{n}(1-S_n(c_\alpha))}{(\alpha-1)\frac{1}{n}+1} && \text{(B.47)}\\
&= \frac{(\alpha-1)\frac{1}{n}(c_\alpha^n - c_\alpha S_n(c_\alpha))}{(\alpha-1)\frac{1}{n}+1} && \text{(B.48)}\\
&\overset{(B.42)}{=} \frac{(\alpha-1)\frac{1}{n}c_\alpha^n + c_\alpha^n}{(\alpha-1)\frac{1}{n}+1} = c_\alpha^n && \text{(B.49)}
\end{aligned}$$

The relation (B.44) can now easily be checked by the same reasoning and induction.

It remains to verify that for this sequence property (B.41) holds. This is done in Lemma B.9.2.3. Note that $\nabla f(\boldsymbol{x}_0) = Q\boldsymbol{x}_0 = \boldsymbol{g}$, where $\boldsymbol{g}$ is defined as in the lemma, and that all gradients $\nabla f(\boldsymbol{x}_t)$ are up to scaling and reordering of the coordinates equivalent to the vector $\boldsymbol{g}$. □

Lemma B.9.2.2. *Let $0 < \alpha < \frac{1}{2}$ and $0 < c_\alpha < 1$ defined by equation (B.42), where $S_n(c_\alpha) = \sum_{i=0}^{n-1} c_\alpha^n$. Then $c_\alpha \geq 1 - \frac{4}{n}\alpha$ for $\alpha \in [0, \frac{1}{2}]$.*

Proof. Using the summation formula for geometric series, $S_n(c_\alpha) = \frac{1-c_\alpha^n}{1-c_\alpha}$ we derive

$$\alpha \overset{(B.42)}{=} 1 - \frac{nc_\alpha^{n-1}}{S_n(c_\alpha)} = \underbrace{1 - \frac{n(1-c_\alpha)c_\alpha^{n-1}}{1-c_\alpha^n}}_{:=\Psi(c_\alpha)} . \tag{B.50}$$

With Taylor expansion we observe that

$$\Psi\left(1 - \frac{3\alpha}{n}\right) \geq \alpha\,, \qquad \Psi\left(1 - \frac{2\alpha}{n}\right) \leq \alpha \tag{B.51}$$

where the first inequality only hold for $n > 2$ and $\alpha \leq \in [0, \frac{1}{2}]$. Hence any solution to (B.50) must satisfy $c_\alpha \geq 1 - \frac{3}{n}\alpha$. □

Lemma B.9.2.3. *Let c_α as in (B.42). Let $\boldsymbol{g} \in \mathbb{R}^n$ be defined as*

$$[\boldsymbol{g}]_i = \frac{(\alpha-1)\frac{1}{n}S_n(c_\alpha) + c_\alpha^{i-1}}{1+\frac{\alpha-1}{n}} \tag{B.52}$$

Then

$$\max_{i\in[n]} \frac{\|\boldsymbol{g}\|_\infty^2}{\frac{1}{n}\|\boldsymbol{g}\|_2^2} \leq 3+3\alpha\,. \tag{B.53}$$

Proof. Observe

$$[\boldsymbol{g}]_i = \frac{(\alpha-1)\frac{1}{n}\left(S_{n-1}(c_\alpha)+c_\alpha^{n-1}\right)+c_\alpha^{n-1}+(c_\alpha^{i-1}-c_\alpha^{n-1})}{1+\frac{\alpha-1}{n}} \tag{B.54}$$

$$\overset{(B.42)}{=} \frac{c_\alpha^{i-1}-c_\alpha^{n-1}}{1+\frac{\alpha-1}{n}} \tag{B.55}$$

Thus $[\boldsymbol{g}]_1 > [\boldsymbol{g}]_2 > \cdots > [\boldsymbol{g}]_n$ and the maximum is attained at

$$\omega(\boldsymbol{g}) := \frac{[\boldsymbol{g}]_1^2}{\frac{1}{n}\sum_{i=1}^n [\boldsymbol{g}]_i^2} = \frac{c_\alpha^2\left(c_\alpha^2-1\right)\left(1-c_\alpha^{n-1}\right)^2 n}{2c_\alpha^{n+1}+2c_\alpha^{n+2}-2c_\alpha^{2n+1}+(n-1)c_\alpha^{2n+2}-c_\alpha^2-nc_\alpha^{2n}} \tag{B.56}$$

For $c_\alpha \geq 1-\frac{3}{n}\alpha$ and $\alpha \leq \frac{1}{2}$, this latter expression can be estimated as

$$\omega(\boldsymbol{g}) \leq 3+3\alpha \tag{B.57}$$

especially $\omega(\boldsymbol{g}) \leq 4$ for $\alpha \leq \frac{1}{3}$. □

B.10 Approximate Gradient Update

In this section we will prove Lemma B.4.0.1. Consider first the following simpler case, where we assume f is given as in least squares, i.e. $f(\boldsymbol{x}) := \frac{1}{2}\|A\boldsymbol{x}-\boldsymbol{b}\|^2$.

In the t_{th} iteration, we choose coordinate i_t to optimize upon and the update from $\boldsymbol{x}_{t+1}$ to $\boldsymbol{x}_t$ can be written as $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma_t \boldsymbol{e}_{i_t}$. Now for any coordinate i other than i_t, it is fairly easy to compute the change in the gradient of the other coordinates. We already observed that $[\boldsymbol{x}_t]_j$ does not change, hence the sub-gradient set of $\Psi_j([\boldsymbol{x}_t]_j)$ and $\Psi_j([\boldsymbol{x}_{t+1}]_j)$ are equal. For the change in ∇f, consider the analysis below:

$$\nabla_i F(\boldsymbol{x}_{t+1}) - \nabla_i F(\boldsymbol{x}_t) = \boldsymbol{a}_i^\top(A\boldsymbol{x}_{t+1}-b) - \boldsymbol{a}_i^\top(A\boldsymbol{x}_t-b) \tag{B.58}$$

$$= \boldsymbol{a}_i^\top\left(A(\boldsymbol{x}_{t+1}-\boldsymbol{x}_t)\right) \tag{B.59}$$

$$= \boldsymbol{a}_i^\top\left(A(\boldsymbol{x}_t+\gamma_t\boldsymbol{e}_{i_t}-\boldsymbol{x}_t)\right) \tag{B.60}$$

$$= \boldsymbol{a}_i^\top \left(\gamma A \boldsymbol{e}_{i_t}\right) = \gamma_t \boldsymbol{a}_i^\top \boldsymbol{a}_{i_t} \tag{B.61}$$

Equation (B.60) comes from the update of $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$.

By the same reasoning, we can now derive the general proof.

***Proof of Lemma** B.4.0.1.* Consider a composite function F as given in Lemma B.4.0.1. By the same reasoning as above, the two sub-gradient sets of $\Psi_j([\boldsymbol{x}_t]_j)$ and $\Psi_j([\boldsymbol{x}_{t+1}]_j)$ are identical, for every passive coordinate $j \neq i_t$. The gradient of F can be written as:

$$\nabla_i F(\boldsymbol{\alpha}_t) = \boldsymbol{a}_i^\top \nabla f(A\boldsymbol{\alpha}_t)$$

For any arbitrary passive coordinate $j \neq i_t$ the change of the gradient can be computed as follows:

$$\begin{aligned}
\nabla_j F(\boldsymbol{x}_{t+1}) - \nabla_j F(\boldsymbol{x}_t) &= \boldsymbol{a}_j^\top \nabla f(A\boldsymbol{x}_{t+1}) - \boldsymbol{a}_j^\top \nabla f(A\boldsymbol{x}_t) \\
&= \boldsymbol{a}_j^\top \left(\nabla f(A\boldsymbol{x}_{t+1}) - \nabla f(A\boldsymbol{x}_t)\right) \qquad \text{(B.62)} \\
&= \boldsymbol{a}_j^\top \Big(\nabla f\big(A(\boldsymbol{x}_t + \gamma_t \boldsymbol{e}_{i_t})\big) - \nabla f\big(A\boldsymbol{x}_t\big)\Big) \\
&\overset{*}{=} \left\langle A^\top \nabla^2 f(A\tilde{\boldsymbol{x}}) \boldsymbol{a}_j, \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \right\rangle \\
&= \left\langle \gamma_t \nabla^2 f(A\tilde{\boldsymbol{x}}) \boldsymbol{a}_j, A(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) \right\rangle \\
&= \gamma_t \boldsymbol{a}_j^\top \nabla^2 f(A\tilde{\boldsymbol{x}}) \boldsymbol{a}_{i_t} \qquad \text{(B.63)}
\end{aligned}$$

Here $\tilde{\boldsymbol{x}}$ is a point on the line segment between $[\boldsymbol{x}_t]_{i_t}$ and $[\boldsymbol{x}_{t+1}]_{i_t}$ which can be found by the Mean Value Theorem. □

B.11 Algorithm and Stability

***Proof of Theorem** B.6.0.1.* As we are interested to study the expected competitive ration $\mathbb{E}[\rho_t]$ for $t \to \infty$, we can assume mixing and consider only the steady state.

Define $\alpha_t \in [0,1]$ s.t. $\alpha_t(n-s) = |\{i \in \mathcal{I}_t \mid i > s\}|$. I.e. $\alpha_t(n-s)$ denotes the number of indices in $|\mathcal{I}_t|$ which do not belong to the set $[s]$.

Denote $\alpha_\infty := \lim_{t\to\infty} \alpha_t$. By equilibrium considerations, the probability that an index $i \notin [s]$ gets picked (and removed from the active set), i.e. $1-\rho_\infty$, must be equal to the probability that an index $j \notin [s]$ enters the active set. Hence

$$\frac{(1-\alpha_\infty)(n-s)}{T_\infty} = 1 - \rho_\infty = \frac{\alpha_\infty(n-s)}{\alpha_\infty(n-s) + cs}. \tag{B.64}$$

We deduce the quadratic relation $\alpha_\infty T_\infty = (1-\alpha_\infty)\left(\alpha_\infty(n-s)+cs\right)$ with solution

$$\alpha_\infty = \frac{n-(1+c)s-T_\infty+\sqrt{n^2+(c-1)^2s^2+2n((c-1)s-T_\infty)+2(1+c)sT_\infty+T_\infty^2}}{2(n-s)}. \tag{B.65}$$

Denote $\theta := n^2+(c-1)^2s^2+2n((c-1)s-T_\infty)+2(1+c)sT+T_\infty^2$. Hence,

$$\rho_\infty \overset{(B.64)}{=} \frac{cs}{\alpha_\infty(n-s)+cs} \overset{(B.65)}{=} \frac{2cs}{cs+n-s-T_\infty+\sqrt{\theta}}. \tag{B.66}$$

We now verify the provided lower bound on ρ_∞:

$$\rho_\infty \overset{(B.64)}{=} 1-\frac{(1-\alpha_\infty)(n-s)}{T_\infty} \geq 1-\frac{n-s}{T_\infty}. \tag{B.67}$$

This bound is sharp for large values of T_∞, ($T_\infty > 2n$, say), but trivial for $T_\infty \leq n-s$. □

B.12 GS rule for Composite Functions

B.12.1 GS-q rule

In this section we show how ASCD can be implemented for the GS-q rule. Define the coordinate-wise model

$$V_i(\boldsymbol{x},y,s) := sy+\frac{L}{2}y^2+\Psi_i(x_i+y) \tag{B.68}$$

The GS-q rule is defined as (cf. Nutini *et al.* [180])

$$i = \arg\min_{i\in[n]} \min_{y\in\mathbb{R}} V(\boldsymbol{x},y,\nabla_i f(\boldsymbol{x})) \tag{B.69}$$

First we show that the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ defined in Algorithm 4 gives valid upper and lower bounds on the value of $\min_{y\in\mathbb{R}} V(\boldsymbol{x},y,\nabla_i f(\boldsymbol{x}))$. We start with the lower bound $\boldsymbol{v}$:

Suppose we have upper and lower bounds, $\ell \leq \nabla_i f(\boldsymbol{x}) \leq u$ on one component of the gradient. Define $\alpha \in [0,1]$ such that $\nabla_i f(\boldsymbol{x}) = (1-\alpha)\ell+\alpha u$. Note that

$$(1-\alpha)V_i(\boldsymbol{x},y,\ell)+\alpha V_i(\boldsymbol{x},y,u) = V_i(\boldsymbol{x},y,\nabla_i f(\boldsymbol{x})) \tag{B.70}$$

Hence,

$$\min\left\{\min_y V_i(\boldsymbol{x},y,u), \min_y V_i(\boldsymbol{x},y,\ell)\right\} \leq \min_y V_i(\boldsymbol{x},y,\nabla_i f(\boldsymbol{x})). \tag{B.71}$$

The derivation of the upper bounds $\boldsymbol{w}$ is a bit more cumbersome. Define $\ell^\star := \arg\min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, \ell)$, $u^\star := \arg\min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, u)$ and observe:

$$V_i(\boldsymbol{x}, u^\star, \nabla_i f(\boldsymbol{x})) = V_i(\boldsymbol{x}, u^\star, u) - (u - \nabla_i f(\boldsymbol{x}))u^\star \leq V_i(\boldsymbol{x}, u^\star, u) - uu^\star + \max\{uu^\star, \ell u^\star\} =: \omega_u \tag{B.72}$$

$$V_i(\boldsymbol{x}, \ell^\star, \nabla_i f(\boldsymbol{x})) = V_i(\boldsymbol{x}, \ell^\star, \ell) - (\ell - \nabla_i f(\boldsymbol{x}))\ell^\star \leq V_i(\boldsymbol{x}, \ell^\star, \ell) - \ell\ell^\star + \max\{u\ell^\star, \ell\ell^\star\} =: \omega_\ell \tag{B.73}$$

$$V_i(\boldsymbol{x}, 0, \nabla_i f(\boldsymbol{x})) = \Psi_i([\boldsymbol{x}]_i) \tag{B.74}$$

Hence $\min_y V_i(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x})) \leq \min\{\omega_\ell, \omega_u, \Psi_i([\boldsymbol{x}]_i)\}$.

Note

$$\omega_u = V_i(\boldsymbol{x}, u^\star, u) + \max\{0, (\ell - u)u^\star\} \tag{B.75}$$

$$\omega_\ell = V_i(\boldsymbol{x}, \ell^\star, \ell) + \max\{0, (u - \ell)\ell^\star\} \tag{B.76}$$

which coincides with the formulas in Algorithm 4.

It remains to show that the computation of the active set is safe, i.e. that the progress achieved by ASCD as defined in Algorithm 4 is always better than the progress achieved by UCD. Let $\mathcal{I}$ be defined as in Algorithm 4. Then

$$\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x})) \leq \frac{1}{n}\sum_{i\in[n]} \min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x})) \tag{B.77}$$

$$= \frac{1}{n} \min_{\boldsymbol{y}\in\mathbb{R}^n} \sum_{i\in[n]} V_i(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x}))\,. \tag{B.78}$$

Using this observation, and the same lines of reasoning as given in [129, Section H.3], it follows immediately that the one step progress of ASCD is at least as good as the for UCD.

B.12.2 GS-r rule

With the notation $[\boldsymbol{y}^\star]_i := \arg\min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, \nabla_i f(\boldsymbol{x}))$, the GS-r rule is defined as (cf. Lee and Seung [129])

$$i = \arg\max_{i\in[n]} |[\boldsymbol{y}^\star]_i|\,. \tag{B.79}$$

In order to implement ASCD for GS-r, we need therefore to maintain lower and upper bounds on the values $|[\boldsymbol{y}^\star]_i|$.

Suppose we have upper and lower bounds, $\ell \leq \nabla_i f(\boldsymbol{x}) \leq u$ on one component of the gradient. Define $\ell^\star := \arg\min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, \ell)$, $u^\star := \arg\min_{y\in\mathbb{R}} V_i(\boldsymbol{x}, y, u)$, then $y^\star$ is contained in the line segment between $\ell^\star$ and $u^\star$. Hence as in Algorithm 5, the lower and

upper bounds can be defined as

$$[\boldsymbol{u}_t]_i := \max_{y\in\mathbb{R}}\{\ell^\star \le y \le u^\star\} \tag{B.80}$$

$$[\boldsymbol{\ell}_t]_i := \min_{y\in\mathbb{R}}\{\ell^\star \le y \le u^\star\} \tag{B.81}$$

However, note that in [180] it is established that GS-r rule can be worse than UCD in general. Hence we cannot expect that ASCD for the GS-r rule is better than UCD in general. However, the by the choice of the active set, the index chosen by the GS-r rule is always contained in the active set, and ASCD approaches GS-r for small errors.

B.13 Experimental Details

We generate a matrix $A \in \mathbb{R}^{m\times n}$ from the standard normal $\mathcal{N}(0,1)$ distribution. m is kept fixed at 1000 but n is chosen 1000 for the l_2 regularized least squares regression and 5000 for l_1 regularized counterpart. 1 is added to each entry (to induce a dependency between columns), multiplied each column by a sample from $\mathcal{N}(0,1)$ multiplied by ten (to induce different Lipschitz constants across the coordinates), and only kept each entry of A non-zero with probability $10\frac{\log(n)}{n}$. This is exactly the same procedure which has been discussed in [180].

Appendix C

Safe Adaptive Importance Sampling

Sebastian U. Stich[1], Anant Raj[2], Martin Jaggi[1]

1 – EPFL, Lausanne

2 – MPI for Intelligent Systems, Tübingen

Abstract

Importance sampling has become an indispensable strategy to speed up optimization algorithms for large-scale applications. Improved adaptive variants—using importance values defined by the complete gradient information which changes during optimization—enjoy favorable theoretical properties, but are typically computationally infeasible. In this paper we propose an efficient approximation of gradient-based sampling, which is based on safe bounds on the gradient. The proposed sampling distribution is (i) provably the *best sampling* with respect to the given bounds, (ii) always better than uniform sampling and fixed importance sampling and (iii) can efficiently be computed—in many applications at negligible extra cost. The proposed sampling scheme is generic and can easily be integrated into existing algorithms. In particular, we show that coordinate-descent (CD) and stochastic gradient descent (SGD) can enjoy significant a speed-up under the novel scheme. The proven efficiency of the proposed sampling is verified by extensive numerical testing.

C.1 Introduction

Modern machine learning applications operate on massive datasets. The algorithms that are used for data analysis face the difficult challenge to cope with the enormous amount of data or the vast dimensionality of the problems. A simple and well established strategy to reduce the computational costs is to split the data and to operate only on a small part of it, as for instance in coordinate descent (CD) methods and stochastic

gradient (SGD) methods. These kind of methods are state of the art for a wide selection of machine learning, deep leaning and signal processing applications [74, 93, 219, 262]. The application of these schemes is not only motivated by their practical preformance, but also well justified by theory [12, 172, 176].

Deterministic strategies are seldom used for the data selection—examples are steepest coordinate descent [31, 180, 252] or screening algorithms [141, 163]. Instead, randomized selection has become ubiquitous, most prominently uniform sampling [71, 72, 215, 217, 219] but also non-uniform sampling based on a *fixed* distribution, commonly referred to as *importance sampling* [12, 48, 164, 172, 176, 194, 203, 234]. While these sampling strategies typically depend on the input data, they do not adapt to the information of the current parameters during optimization. In contrast, *adaptive* importance sampling strategies constantly re-evaluate the relative importance of each data point during training and thereby often surpass the performance of static algorithms [49, 87, 184, 185, 189, 209]. Common strategies are *gradient-based* sampling [185, 272, 273] (mostly for SGD) and *duality gap-based* sampling for CD [49, 189].

The drawbacks of adaptive strategies are twofold: often the provable theoretical guarantees can be worse than the complexity estimates for uniform sampling [16, 189] and often it is computationally inadmissible to compute the optimal adaptive sampling distribution. For instance gradient based sampling requires the computation of the full gradient in each iteration [185, 272, 273]. Therefore one has to rely on approximations based on upper bounds [272, 273], or stale values [4, 185]. But in general these approximations can again be worse than uniform sampling.

This makes it necessary to develop adaptive strategies that can efficiently be computed in every iteration and that come with theoretical guarantees that show their advantage over fixed sampling.

Our contributions. In this paper we propose an efficient approximation of the gradient-based sampling in the sense that (i) it can efficiently be computed in every iteration, (ii) is provably better than uniform or fixed importance sampling and (iii) recovers the gradient-based sampling in the full-information setting. The scheme is completely generic and can easily be added as an improvement to both CD and SGD type methods.

As our key contributions, we

1. show that gradient-based sampling in CD methods is theoretically better than the classical fixed sampling, the speed-up can reach a factor of the dimension n (Section C.2);

2. propose a generic and efficient *adaptive importance sampling* strategy that can be applied in CD and SGD methods and enjoys favorable properties—such as mentioned above (Section C.3);

3. demonstrate how the novel scheme can efficiently be integrated in CD and SGD on an important class of structured optimization problems (Section C.4);

4. supply numerical evidence that the novel sampling performs well on real data (Section C.5).

Notation. For $\boldsymbol{x} \in \mathbb{R}^n$ define $[\boldsymbol{x}]_i := \langle \boldsymbol{x}, \boldsymbol{e}_i \rangle$ with $\boldsymbol{e}_i$ the standard unit vectors in $\mathbb{R}^n$. We abbreviate $\nabla_i f := [\nabla f]_i$. A convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ with L-Lipschitz continuous gradient satisfies

$$f(\boldsymbol{x}+\eta\boldsymbol{u}) \leq f(\boldsymbol{x}) + \eta \langle \boldsymbol{u}, \nabla f(\boldsymbol{x}) \rangle + \tfrac{\eta^2 L_{\boldsymbol{u}}}{2}\|\boldsymbol{u}\|^2 \qquad \forall \boldsymbol{x} \in \mathbb{R}^n, \forall \eta \in \mathbb{R}, \tag{C.1}$$

for every direction $\boldsymbol{u} \in \mathbb{R}^n$ and $L_{\boldsymbol{u}} = L$. A function with coordinate-wise L_i-Lipschitz continuous gradients[1] for constants $L_i > 0$, $i \in [n] := \{1, \dots, n\}$, satisfies (C.1) just along coordinate directions, i.e. $\boldsymbol{u} = \boldsymbol{e}_i$, $L_{\boldsymbol{e}_i} = L_i$ for every $i \in [n]$. A function is coordinate-wise L-smooth if $L_i \leq L$ for $i = 1, \dots, n$. For convenience we introduce vector $\boldsymbol{l} = (L_1, \dots, L_n)^\top$ and matrix $\mathbf{L} = \operatorname{diag}(\boldsymbol{l})$. A probability vector $\boldsymbol{p} \in \Delta^n := \{\boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \colon \|\boldsymbol{x}\|_1 = 1\}$ defines a probability distribution $\mathcal{P}$ over $[n]$ and we denote by $i \sim \boldsymbol{p}$ a sample drawn from $\mathcal{P}$.

C.2 Adaptive Importance Sampling with Full Information

In this section we argue that adaptive sampling strategies are theoretically well justified, as they can lead to significant improvements over static strategies. In our exhibition we focus first on CD methods, as we also propose a novel stepsize strategy for CD in this contribution. Then we revisit the results regarding stochastic gradient descent (SGD) already present in the literature.

C.2.1 Coordinate Descent with Adaptive Importance Sampling

We address general minimization problems $\min_{\boldsymbol{x}} f(\boldsymbol{x})$. Let the objective $f\colon \mathbb{R}^n \to \mathbb{R}$ be convex with coordinate-wise L_i-Lipschitz continuous gradients. Coordinate descent methods generate sequences $\{\boldsymbol{x}_k\}_{k\geq 0}$ of iterates that satisfy the relation

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma_k \nabla_{i_k} f(\boldsymbol{x}_k) \boldsymbol{e}_{i_k}. \tag{C.2}$$

Here, the direction i_k is either chosen deterministically (cyclic descent, steepest descent), or randomly picked according to a probability vector $\boldsymbol{p}_k \in \Delta^n$. In the classical literature, the stepsize is often chosen such as to minimize the quadratic upper bound (C.1), i.e. $\gamma_k = L_{i_k}^{-1}$. In this work we propose to set $\gamma_k = \alpha_k [\boldsymbol{p}_k]_{i_k}^{-1}$ where α_k does not depend on the chosen direction i_k. This leads to directionally-unbiased updates, like it is common

[1] $|\nabla_i f(\boldsymbol{x}+\eta\boldsymbol{e}_i) - \nabla_i f(\boldsymbol{x})| \leq L_i |\eta|, \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \forall \eta \in \mathbb{R}.$

among SGD-type methods. It holds

$$\mathbb{E}_{i_k\sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k\right] \overset{(C.1)}{\leq} \mathbb{E}_{i_k\sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_k) - \frac{\alpha_k}{[\boldsymbol{p}_k]_{i_k}}\left(\nabla_{i_k} f(\boldsymbol{x}_k)\right)^2 + \frac{L_{i_k}\alpha_k^2}{2[\boldsymbol{p}_k]_{i_k}^2}\left(\nabla_{i_k} f(\boldsymbol{x}_k)\right)^2 \mid \boldsymbol{x}_k\right]$$
$$= f(\boldsymbol{x}_k) - \alpha_k\|\nabla f(\boldsymbol{x}_k)\|_2^2 + \sum_{i=1}^n \frac{L_i\alpha_k^2}{2[\boldsymbol{p}_k]_i}\left(\nabla_i f(\boldsymbol{x}_k)\right)^2. \tag{C.3}$$

In adaptive strategies we have the freedom to chose both variables α_k and $\boldsymbol{p}_k$ as we like. We therefore propose to chose them in such a way that they *minimize* the upper bound (C.3) in order to maximize the expected progress. The optimal $\boldsymbol{p}_k$ in (C.3) is independent of α_k, but the optimal α_k depends on $\boldsymbol{p}_k$. We can state the following useful observation.

Lemma C.2.1.1. *If $\alpha_k = \alpha_k(\boldsymbol{p}_k)$ is the minimizer of (C.3), then $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \frac{\alpha_k}{[\boldsymbol{p}_k]_{i_k}}\nabla_{i_k} f(\boldsymbol{x}_k)\boldsymbol{e}_{i_k}$ satisfies*

$$\mathbb{E}_{i_k\sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k\right] \leq f(\boldsymbol{x}_k) - \frac{\alpha_k(\boldsymbol{p}_k)}{2}\|\nabla f(\boldsymbol{x}_k)\|_2^2. \tag{C.4}$$

Consider two examples. In the first one we pick a sub-optimal, but very common [172] distribution:

Example C.2.1.1 (L_i-based sampling)**.** *Let $\boldsymbol{p}_{\mathbf{L}} \in \Delta^n$ defined as $[\boldsymbol{p}_{\mathbf{L}}]_i = \frac{L_i}{\mathrm{Tr}[\mathbf{L}]}$ for $i \in [n]$, where $\mathbf{L} = \mathrm{diag}(L_1,\ldots,L_n)$. Then $\alpha_k(\boldsymbol{p}_{\mathbf{L}}) = \frac{1}{\mathrm{Tr}[\mathbf{L}]}$.*

The distribution $\boldsymbol{p}_{\mathbf{L}}$ is often referred to as (fixed) *importance* sampling. In the special case when $L_i = L$ for all $i \in [n]$, this boils down to uniform sampling.

Example C.2.1.2 (Optimal sampling[2])**.** *Equation (C.3) is minimized for probabilities $[\boldsymbol{p}_k^\star]_i = \frac{\sqrt{L_i}|\nabla_i f(\boldsymbol{x}_k)|}{\|\sqrt{\mathbf{L}}\nabla f(\boldsymbol{x}_k)\|_1}$ and $\alpha_k(\boldsymbol{p}_k^\star) = \frac{\|\nabla f(\boldsymbol{x}_k)\|_2^2}{\|\sqrt{\mathbf{L}}\nabla f(\boldsymbol{x}_k)\|_1^2}$. Observe $\frac{1}{\mathrm{Tr}[\mathbf{L}]} \leq \alpha_k(\boldsymbol{p}_k^\star) \leq \frac{1}{L_{\min}}$, where $L_{\min} := \min_{i\in[n]} L_i$.*

To prove this result, we rely on the following Lemma—the proof of which, as well as for the claims above, is deferred to Section C.7.1 of the appendix. Here $|\cdot|$ is applied entry-wise.

Lemma C.2.1.2. *Define $V(\boldsymbol{p},\boldsymbol{x}) := \sum_{i=1}^n \frac{L_i[\boldsymbol{x}]_i^2}{[\boldsymbol{p}]_i}$. Then $\arg\min_{\boldsymbol{p}\in\Delta^n} V(\boldsymbol{p},\boldsymbol{x}) = \frac{|\sqrt{\mathbf{L}}\boldsymbol{x}|}{\|\sqrt{\mathbf{L}}\boldsymbol{x}\|_1}$.*

The ideal adaptive algorithm. We propose to chose the stepsize and the sampling distribution for CD as in Example C.2.1.2. One iteration of the resulting CD method is illustrated in Algorithm 5. Our bounds on the expected one-step progress can be used to derive convergence rates of this algorithm with the standard techniques. This is

[2] Here "optimal" refers to the fact that $\boldsymbol{p}_k^\star$ is optimal with respect to the given model (C.1) of the objective function. If the model is not accurate, there might exist a sampling that yields larger expected progress on f.

exemplified in Appendix C.7.1. In the next Section C.3 we develop a practical variant of the ideal algorithm.

Efficiency gain. By comparing the estimates provided in the examples above, we see that the expected progress of the proposed method is always at least as good as for the fixed sampling. For instance in the special case where $L = L_i$ for $i \in [n]$, the L_i-based sampling is just uniform sampling with $\alpha_k(\boldsymbol{p}_{\text{unif}}) = \frac{1}{Ln}$. On the other hand $\alpha_k(\boldsymbol{p}_k^\star) = \frac{\|\nabla f(\boldsymbol{x}_k)\|_2^2}{L\|\nabla f(\boldsymbol{x}_k)\|_1^2}$, which can be n times larger than $\alpha_k(\boldsymbol{p}_{\text{unif}})$. The expected one-step progress in this extreme case coincides with the one-step progress of steepest coordinate descent [180].

C.2.2 SGD with Adaptive Sampling

SGD methods are applicable to objective functions which decompose as a sum

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x}) \tag{C.5}$$

with each $f_i\colon \mathbb{R}^d \to \mathbb{R}$ convex. In previous work [185, 272, 273] is has been argued that the following gradient-based sampling $[\tilde{\boldsymbol{p}}_k^\star]_i = \frac{\|\nabla f_i(\boldsymbol{x}_k)\|_2}{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}$ is optimal in the sense that it maximizes the expected progress (C.3). Zhao and Zhang [272] derive complexity estimates for composite functions. For non-composite functions it becomes easier to derive the complexity estimate. For completeness, we add this simpler proof in Appendix C.7.2.

C.3 Safe Adaptive Importance Sampling with Limited Information

In the previous section we have seen that gradient-based sampling (Example C.2.1.2) can yield a massive speed-up compared to a static sampling distribution (Example C.2.1.1). However, sampling according to $\boldsymbol{p}_k^\star$ in CD requires the knowledge of the full gradient $\nabla f(\boldsymbol{x}_k)$ in each iteration. And likewise, sampling from $\tilde{\boldsymbol{p}}_k^\star$ in SGD requires the knowledge of the gradient norms of all components—both these operations are in general inadmissible, i.e. the compute cost would void all computational benefits of the iterative (stochastic) methods over full gradient methods.

However, it is often possible to efficiently compute *approximations* of $\boldsymbol{p}_k^\star$ or $\tilde{\boldsymbol{p}}_k^\star$ instead. In contrast to previous contributions, we here propose a *safe* way to compute such approximations. By this we mean that our approximate sampling is provably never worse than static sampling, and moreover, we show that our solution is the *best possible* with respect to the limited information at hand.

C.3.1 An Optimization Formulation for Sampling

Algorithm 5 Optimal sampling

(compute full gradient)

Compute $\nabla f(\boldsymbol{x}_k)$

(define optimal sampling)

Define $(\boldsymbol{p}_k^\star, \alpha_k^\star)$ as in Example C.2.1.2

$i_k \sim \boldsymbol{p}_k^\star$

$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \frac{\alpha_k^\star}{[\boldsymbol{p}_k^\star]_{i_k}} \nabla_{i_k} f(\boldsymbol{x}_k)$

Algorithm 6 Proposed safe sampling

(update l.- and u.-bounds)

Update $\boldsymbol{\ell}$, $\boldsymbol{u}$

(compute safe sampling)

Define $(\hat{\boldsymbol{p}}_k, \hat{\alpha}_k)$ as in (C.7)

$i_k \sim \hat{\boldsymbol{p}}_k$

Compute $\nabla_{i_k} f(\boldsymbol{x}_k)$

$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \frac{\hat{\alpha}_k}{[\hat{\boldsymbol{p}}_k]_{i_k}} \nabla_{i_k} f(\boldsymbol{x}_k)$

Algorithm 7 Fixed sampling

(define fixed sampling)

Define $(\boldsymbol{p}_L, \bar{\alpha})$ as in Example C.2.1.1

$i_k \sim \boldsymbol{p}_L$

Compute $\nabla_{i_k} f(\boldsymbol{x}_k)$

$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \frac{\bar{\alpha}}{[\boldsymbol{p}_L]_{i_k}} \nabla_{i_k} f(\boldsymbol{x}_k)$

Figure C.1: CD with different sampling strategies. Whilst Alg. 5 requires to compute the full gradient, the compute operation in Alg. 6 is as cheap as for fixed importance sampling, Alg. 7. Defining the safe sampling $\hat{\boldsymbol{p}}_k$ requires $O(n \log n)$ time.

Formally, we assume that we have in each iteration access to two vectors $\boldsymbol{\ell}_k, \boldsymbol{u}_k \in \mathbb{R}^n_{\geq 0}$ that provide safe upper and lower bounds on either the absolute values of the gradient entries ($[\boldsymbol{\ell}_k]_i \leq |\nabla_i f(\boldsymbol{x}_k)| \leq [\boldsymbol{u}_k]_i$) for CD, or of the gradient norms in SGD: ($[\boldsymbol{\ell}_k]_i \leq \|\nabla f_i(\boldsymbol{x}_k)\|_2 \leq [\boldsymbol{u}_k]_i$). We postpone the discussion of this assumption to Section C.4, where we give concrete examples.

The minimization of the upper bound (C.3) amounts to the equivalent problem[3]

$$\min_{\alpha_k} \min_{\boldsymbol{p}_k \in \Delta^n} \left[-\alpha_k \|\boldsymbol{c}_k\|_2^2 + \frac{\alpha_k^2}{2} V(\boldsymbol{p}_k, \boldsymbol{c}_k) \right] \quad \Leftrightarrow \quad \min_{\boldsymbol{p}_k \in \Delta^n} \frac{V(\boldsymbol{p}_k, \boldsymbol{c}_k)}{\|\boldsymbol{c}_k\|_2^2} \tag{C.6}$$

where $\boldsymbol{c}_k \in \mathbb{R}^n$ represents the *unknown* true gradient. That is, with respect to the bounds $\boldsymbol{\ell}_k, \boldsymbol{u}_k$, we can write $\boldsymbol{c}_k \in C_k := \{\boldsymbol{x} \in \mathbb{R}^n \colon [\boldsymbol{\ell}_k]_i \leq [\boldsymbol{x}]_i \leq [\boldsymbol{u}_k]_i, i \in [n]\}$. In Example C.2.1.2 we derived the optimal solution for a fixed $\boldsymbol{c}_k \in C_k$. However, this is not sufficient to find the optimal solution for an arbitrary $\boldsymbol{c}_k \in C_k$. Just computing the optimal solution for an arbitrary (but fixed) $\boldsymbol{c}_k \in C_k$ is unlikely to yield a good solution. For instance both extreme cases $\boldsymbol{c}_k = \ell_k$ and $\boldsymbol{c}_k = \boldsymbol{u}_k$ (the latter choice is quite common, cf. [189, 272]) might be poor. This is demonstrated in the next example.

Example C.3.1.1. *Let* $\boldsymbol{\ell} = (1,2)^\top$, $\boldsymbol{u} = (2,3)^\top$, $\boldsymbol{c} = (2,2)^\top$ *and* $L_1 = L_2 = 1$. *Then* $V\left(\frac{\boldsymbol{\ell}}{\|\boldsymbol{\ell}\|_1}, \boldsymbol{c}\right) = \frac{9}{4}\|\boldsymbol{c}\|_2^2$, $V\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_1}, \boldsymbol{c}\right) = \frac{25}{12}\|\boldsymbol{c}\|_2^2$, *whereas for uniform sampling* $V\left(\frac{\boldsymbol{c}}{\|\boldsymbol{c}\|_1}, \boldsymbol{c}\right) = 2\|\boldsymbol{c}\|_2^2$.

The proposed sampling. As a consequence of these observations, we propose to solve the following optimization problem to find the best sampling distribution with respect to

[3] Although only shown here for CD, an equivalent optimization problem arises for SGD methods, cf. [272].

Algorithm 8 Computing the Safe Sampling for Gradient Information $\boldsymbol{\ell}, \boldsymbol{u}$

Input: $\mathbf{0}_n \leq \boldsymbol{\ell} \leq \boldsymbol{u}$, $\mathbf{L}$, **Initialize:** $\boldsymbol{c} = \mathbf{0}_n$, $u = 1$, $\ell = n$, $D = \emptyset$.
$\boldsymbol{\ell}^{\text{sort}} := \text{sort\_asc}(\sqrt{\mathbf{L}^{-1}}\boldsymbol{\ell})$, $\boldsymbol{u}^{\text{sort}} := \text{sort\_asc}(\sqrt{\mathbf{L}^{-1}}\boldsymbol{u})$, $m = \max(\boldsymbol{\ell}^{\text{sort}})$
while $u \leq \ell$ **do**
  if $[\boldsymbol{\ell}^{\text{sort}}]_\ell > m$ **then** *(largest undecided lower bound is violated)*
    Set corresponding $[\boldsymbol{c}]_{\text{index}} := [\sqrt{\mathbf{L}}\boldsymbol{\ell}^{\text{sort}}]_\ell$; $\ell := \ell - 1$; $D := D \cup \{\text{index}\}$
  else if $[\boldsymbol{u}^{\text{sort}}]_u < m$ **then** *(smallest undecided upper bound is violated)*
    Set corresponding $[\boldsymbol{c}]_{\text{index}} := [\sqrt{\mathbf{L}}\boldsymbol{u}^{\text{sort}}]_u$; $u := u + 1$; $D := D \cup \{\text{index}\}$
  else
    break *(no constraints are violated)*
  end if
  $m := \|\boldsymbol{c}\|_2^2 \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^{-1}$ *(update m as in (C.9))*
end while
Set $[\boldsymbol{c}]_i := \sqrt{L_i} m$ for all $i \notin D$ and **Return** $\left(\boldsymbol{c}, \boldsymbol{p} = \frac{\sqrt{\mathbf{L}}\boldsymbol{c}}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}, v = \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2}\right)$

C_k:

$$v_k := \min_{\boldsymbol{p} \in \Delta^n} \max_{\boldsymbol{c} \in C_k} \frac{V(\boldsymbol{p}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2}, \qquad \text{and to set} \qquad (\alpha_k, \boldsymbol{p}_k) := \left(\tfrac{1}{v_k}, \hat{\boldsymbol{p}}_k\right), \tag{C.7}$$

where $\hat{\boldsymbol{p}}_k$ denotes a solution of (C.7). The resulting algorithm for CD is summarized in Alg. 6.

In the remainder of this section we discuss the properties of the solution $\hat{\boldsymbol{p}}_k$ (Theorem C.3.2.1) and how such a solution can be efficiently be computed (Theorem C.3.2.2, Algorithm 8).

C.3.2 Proposed Sampling and its Properties

Theorem C.3.2.1. *Let $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{c}}) \in \Delta^n \times \mathbb{R}^n_{\geq 0}$ denote a solution of (C.7). Then $L_{\min} \leq v_k \leq \operatorname{Tr}[\mathbf{L}]$ and*

1. $\max_{\boldsymbol{c} \in C_k} \frac{V(\hat{\boldsymbol{p}}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \leq \max_{\boldsymbol{c} \in C_k} \frac{V(\boldsymbol{p}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2}$, $\forall \boldsymbol{p} \in \Delta^n$; *($\hat{\boldsymbol{p}}$ has the best worst-case guarantee)*
2. $V(\hat{\boldsymbol{p}}, \boldsymbol{c}) \leq \operatorname{Tr}[\mathbf{L}] \cdot \|\boldsymbol{c}\|_2^2$, $\forall \boldsymbol{c} \in C_k$. *($\hat{\boldsymbol{p}}$ is always better than L_i-based sampling)*

Remark C.3.2.1. *In the special case $L_i = L$ for all $i \in [n]$, the L_i-based sampling boils down to uniform sampling (Example C.2.1.1) and $\hat{\boldsymbol{p}}$ is better than uniform sampling: $V(\hat{\boldsymbol{p}}, \boldsymbol{c}) \leq Ln\|\boldsymbol{c}\|_2^2$, $\forall \boldsymbol{c} \in C_k$.*

Proof. Property *(i)* is an immediate consequence of (C.7). Moreover, observe that the L_i-based sampling $\boldsymbol{p}_L$ is a feasible solution in (C.7) with value $\frac{V(\boldsymbol{p}_L, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \equiv \operatorname{Tr}[\mathbf{L}]$ for all

$\boldsymbol{c} \in C_k$. Hence

$$L_{\min} \leq \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2} \overset{\text{C.2.1.2}}{=} \min_{\boldsymbol{p}\in\Delta^n} \frac{V(\boldsymbol{p},\boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \leq \frac{V(\hat{\boldsymbol{p}},\boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \overset{(*)}{\leq} \frac{V(\hat{\boldsymbol{p}},\hat{\boldsymbol{c}})}{\|\hat{\boldsymbol{c}}\|_2^2} \overset{\text{(C.7)}}{\leq} \max_{\boldsymbol{c}\in C_k} \frac{V(\boldsymbol{p}_L,\boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} = \operatorname{Tr}[\mathbf{L}]\,, \tag{C.8}$$

for all $\boldsymbol{c} \in C_k$, thus $v_k \in [L_{\min}, \operatorname{Tr}[\mathbf{L}]]$ and *(ii)* follows. We prove inequality $(*)$ in the appendix, by showing that min and max can be interchanged in (C.7). □

A geometric interpretation. We show in Appendix C.8 that the optimization problem (C.7) can equivalently be written as $\sqrt{v_k} = \max_{\boldsymbol{c}\in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}{\|\boldsymbol{c}\|_2} = \max_{\boldsymbol{c}\in C_k} \frac{\langle\sqrt{\boldsymbol{l}},\boldsymbol{c}\rangle}{\|\boldsymbol{c}\|_2}$, where $[\boldsymbol{l}]_i = L_i$ for $i \in [n]$. The maximum is thus attained for vectors $\boldsymbol{c} \in C_k$ that minimize the angle with the vector $\boldsymbol{l}$.

Theorem C.3.2.2. *Let $\boldsymbol{c} \in C_k$, $\boldsymbol{p} = \frac{\sqrt{\mathbf{L}}\boldsymbol{c}}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}$ and denote $m = \|\boldsymbol{c}\|_2^2 \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^{-1}$. If*

$$[\boldsymbol{c}]_i = \begin{cases} [\boldsymbol{u}_k]_i & \text{if } [\boldsymbol{u}_k]_i \leq \sqrt{L_i}m\,, \\ [\boldsymbol{\ell}_k]_i & \text{if } [\boldsymbol{\ell}_k]_i \geq \sqrt{L_i}m\,, \\ \sqrt{L_i}m & \text{otherwise}, \end{cases} \qquad \forall i \in [n]\,, \tag{C.9}$$

then $(\boldsymbol{p},\boldsymbol{c})$ is a solution to (C.7). Moreover, such a solution can be computed in time $O(n\log n)$.

Proof. This can be proven by examining the optimality conditions of problem (C.7). This is deferred to Section C.8.1 of the appendix. A procedure that computes such a solution is depicted in Algorithm 8. The algorithm makes extensive use of (C.9). For simplicity, assume first $\mathbf{L} = \mathbf{I}_n$ for now. In each iteration t , a potential solution vector $\boldsymbol{c}_t$ is proposed, and it is verified whether this vector satisfies all optimality conditions. In Algorithm 8, $\boldsymbol{c}_t$ is just implicit, with $[\boldsymbol{c}_t]_i = [\boldsymbol{c}]_i$ for decided indices $i \in D$ and $[\boldsymbol{c}_t]_i = [\sqrt{\mathbf{L}}m]_i$ for undecided indices $i \notin D$. After at most n iterations a valid solution is found. By sorting the components of $\sqrt{\mathbf{L}^{-1}}\boldsymbol{\ell}_k$ and $\sqrt{\mathbf{L}^{-1}}\boldsymbol{u}_k$ by their magnitude, at most a linear number of inequality checks in (C.9) have to be performed in total. Hence the running time is dominated by the $O(n\log n)$ complexity of the sorting algorithm. A formal proof is given in the appendix. □

Competitive Ratio. We now compare the proposed sampling distribution $\hat{\boldsymbol{p}}_k$ with the optimal sampling solution in *hindsight*. We know that if the true (gradient) vector $\tilde{\boldsymbol{c}} \in C_k$ would be given to us, then the corresponding optimal probability distribution would be $\boldsymbol{p}^\star(\tilde{\boldsymbol{c}}) = \frac{\sqrt{\mathbf{L}}\tilde{\boldsymbol{c}}}{\|\sqrt{\mathbf{L}}\tilde{\boldsymbol{c}}\|_1}$ (Example C.2.1.2). Thus, for this $\tilde{\boldsymbol{c}}$ we can now analyze the ratio $\frac{V(\hat{\boldsymbol{p}}_k,\tilde{\boldsymbol{c}})}{V(\boldsymbol{p}^\star(\tilde{\boldsymbol{c}}),\tilde{\boldsymbol{c}})}$. As we are interested in the worst case ratio among all possible candidates

$\tilde{\boldsymbol{c}} \in C_k$, we define

$$\rho_k := \max_{\boldsymbol{c} \in C_k} \frac{V(\hat{\boldsymbol{p}}, \boldsymbol{c})}{V(\boldsymbol{p}^\star(\boldsymbol{c}), \boldsymbol{c})} = \max_{\boldsymbol{c} \in C_k} \frac{V(\hat{\boldsymbol{p}}, \boldsymbol{c})}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2} \,. \tag{C.10}$$

Lemma C.3.2.3. *Let* $w_k := \min_{\boldsymbol{c} \in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2}$. *Then* $L_{\min} \leq w_k \leq v_k$, *and* $\rho_k \leq \frac{v_k}{w_k} (\leq \frac{v_k}{L_{\min}})$.

Lemma C.3.2.4. *Let* $\gamma \geq 1$. *If* $[C_k]_i \cap \gamma [C_k]_i = \emptyset$ *and* $\gamma^{-1}[C_k]_i \cap [C_k]_i = \emptyset$ *for all* $i \in [n]$ *(here* $[C_k]_i$ *denotes the projection on the i-th coordinate), then* $\rho_k \leq \gamma^4$.

These two lemma provide bounds on the competitive ratio. Whilst Lemma C.3.2.4 relies on a relative accuracy condition, Lemma C.3.2.3 can always be applied. However, the corresponding minimization problem is non-convex. Note that knowledge of ρ_k is not needed to run the algorithm.

C.4 Example Safe Gradient Bounds

In this section, we argue that for a large class of objective functions of interest in machine learning, suitable safe upper and lower bounds $\boldsymbol{\ell}, \boldsymbol{u}$ on the gradient along every coordinate direction can be estimated and maintained efficiently during optimization. A similar argument can be given for the efficient approximation of component wise gradient norms in finite sum objective based stochastic gradient optimization.

As the guiding example, we will here showcase the training of generalized linear models (GLMs) as e.g. in regression, classification and feature selection. These models are formulated in terms of a given data matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ with columns $\boldsymbol{a}_i \in \mathbb{R}^d$ for $i \in [n]$.

Coordinate Descent - GLMs with Arbitrary Regularizers. Consider general objectives of the form $f(\boldsymbol{x}) := h(\mathbf{A}\boldsymbol{x}) + \sum_{i=1}^n \psi_i([\boldsymbol{x}]_i)$ with an arbitrary convex separable regularizer term given by the $\psi_i \colon \mathbb{R} \to \mathbb{R}$ for $i \in [n]$. A key example is when $h \colon \mathbb{R}^d \to \mathbb{R}$ describes the *least-squares* regression objective $h(\mathbf{A}\boldsymbol{x}) = \frac{1}{2}\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ for a $\boldsymbol{b} \in \mathbb{R}^d$. Using that this h is twice differentiable with $\nabla^2 h(\mathbf{A}\boldsymbol{x}) = \mathbf{I}_n$, it is easy to see that we can track the evolution of all gradient entries, when performing CD steps, as follows:

$$\nabla_i f(\boldsymbol{x}_{k+1}) - \nabla_i f(\boldsymbol{x}_k) = \gamma_k \langle \boldsymbol{a}_i, \boldsymbol{a}_{i_k} \rangle \,, \quad \forall i \neq i_k \,. \tag{C.11}$$

for i_k being the coordinate changed in step k (here we also used the separability of the regularizer).

Therefore, all gradient changes can be tracked exactly if the inner products of all datapoints are available, or approximately if those inner products can be upper and lower bounded. For computational efficiency, we in our experiments simply use Cauchy-Schwarz $|\langle \boldsymbol{a}_i, \boldsymbol{a}_{i_k} \rangle| \leq \|\boldsymbol{a}_i\| \cdot \|\boldsymbol{a}_{i_k}\|$. This results in safe upper and lower bounds $[\boldsymbol{\ell}_{k+1}]_i \leq \nabla_i f(\boldsymbol{x}_{k+1}) \leq [\boldsymbol{u}_{k+1}]_i$ for all inactive coordinates $i \neq i_k$. (For the active coordinate i_k itself

one observes the true value without uncertainty). These bounds can be updated in linear time $O(n)$ in every iteration.

For general smooth h (again with arbitrary separable regularizers ψ_i), (C.11) can readily be extended to hold [232, Lemma 4.1], the inner product change term becoming $\langle \boldsymbol{a}_i, \nabla^2 f(\mathbf{A}\tilde{\boldsymbol{x}})\boldsymbol{a}_{i_k}\rangle$ instead, when assuming h is twice-differentiable. Here $\tilde{\boldsymbol{x}}$ will be an element of the line segment $[\boldsymbol{x}_k, \boldsymbol{x}_{k+1}]$.

Stochastic Gradient Descent - GLMs. We now present a similar result for finite sum problems (C.5) for the use in SGD based optimization, that is $f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n h_i(\boldsymbol{a}_i^\top \boldsymbol{x})$.

Lemma C.4.0.1. *Consider $f\colon \mathbb{R}^d \to \mathbb{R}$ as above, with twice differentiable $h_i\colon \mathbb{R} \to \mathbb{R}$. Let $\boldsymbol{x}_k, \boldsymbol{x}_{k+1} \in \mathbb{R}^d$ denote two successive iterates of SGD, i.e. $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k - \eta_k\, \boldsymbol{a}_{i_k} \nabla h_{i_k}(\boldsymbol{a}_{i_k}^\top \boldsymbol{x}_k) = \boldsymbol{x}_k + \gamma_k\, \boldsymbol{a}_{i_k}$. Then there exists $\tilde{\boldsymbol{x}} \in \mathbb{R}^d$ on the line segment between $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$, $\tilde{\boldsymbol{x}} \in [\boldsymbol{x}_k, \boldsymbol{x}_{k+1}]$ with*

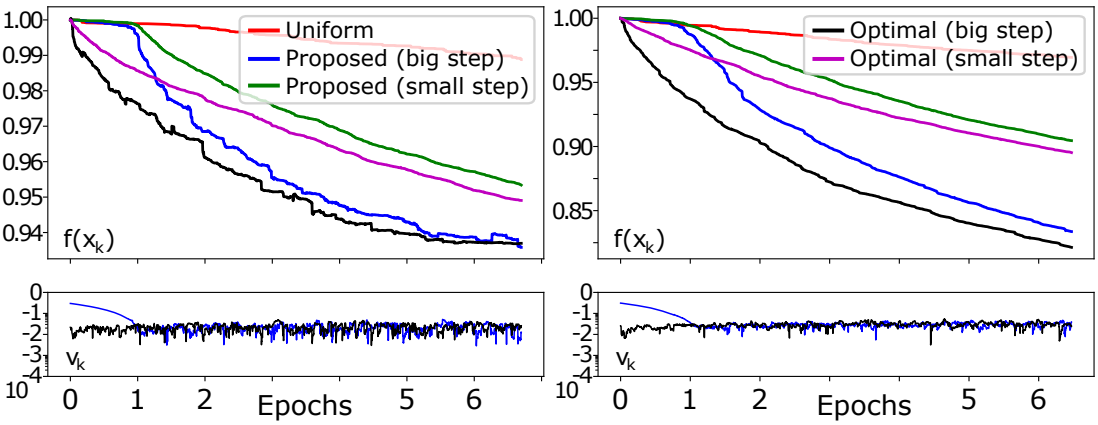
$$\nabla f_i(\boldsymbol{x}_{k+1}) - \nabla f_i(\boldsymbol{x}_k) \;=\; \gamma_k\, \nabla^2 h_i(\boldsymbol{a}_i^\top \tilde{\boldsymbol{x}})\, \langle \boldsymbol{a}_i, \boldsymbol{a}_{i_k}\rangle\, \boldsymbol{a}_i\,, \quad \forall\, i \neq i_k\,. \tag{C.12}$$

This leads to safe upper and lower bounds for the norms of the partial gradient, $[\boldsymbol{\ell}_k]_i \leq \|\nabla f_i(\boldsymbol{x}_k)\|_2 \leq [\boldsymbol{u}_k]_i$, that can be updated in linear time $O(n)$, analogous to the coordinate case discussed above.[4]

We note that there are many other ways to track safe gradient bounds for relevant machine learning problems, including possibly more tight ones. We here only illustrate the simplest variants, highlighting the fact that our new sampling procedure works for any safe bounds $\boldsymbol{\ell}, \boldsymbol{u}$.

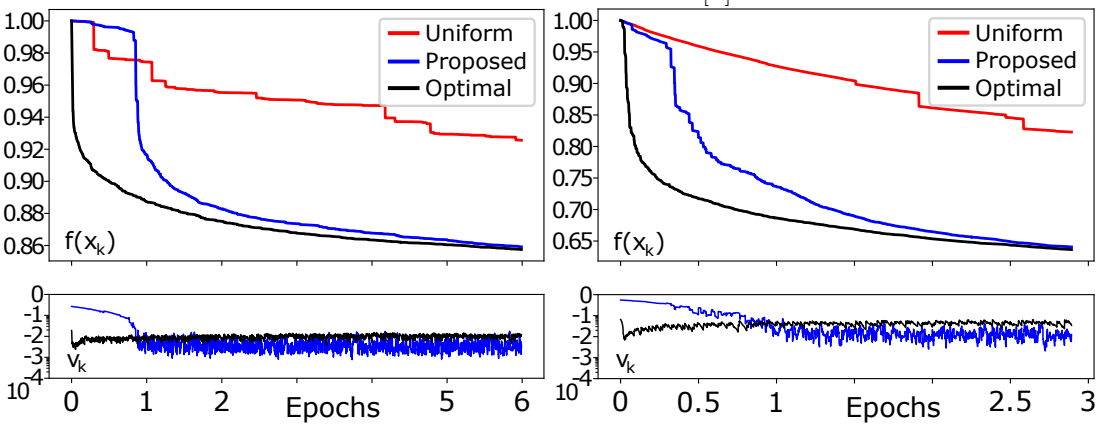
Computational Complexity. In this section, we have demonstrated how safe upper and lower bounds $\boldsymbol{\ell}, \boldsymbol{u}$ on the gradient information can be obtained for GLMs, and argued that these bounds can be updated in time $O(n)$ per iteration of CD and SGD. The computation of the proposed sampling takes $O(n \log n)$ time (Theorem C.3.2.2). Hence, the introduced overhead in Algorithm 6 compared to fixed sampling (Algorithm 7) is of the order $O(n \log n)$ in every iteration. The computation of one coordinate of the gradient, $\nabla_{i_k} f(\boldsymbol{x}_k)$, takes time $\Theta(d)$ for general data matrices. Hence, when $d = \Omega(n)$, the introduced overhead reduces to $O(\log n)$ per iteration.

[4] Here we use the efficient representation $\nabla f_i(\boldsymbol{x}) = \theta(\boldsymbol{x}) \cdot \boldsymbol{a}_i$ for $\theta(\boldsymbol{x}) \in \mathbb{R}$.



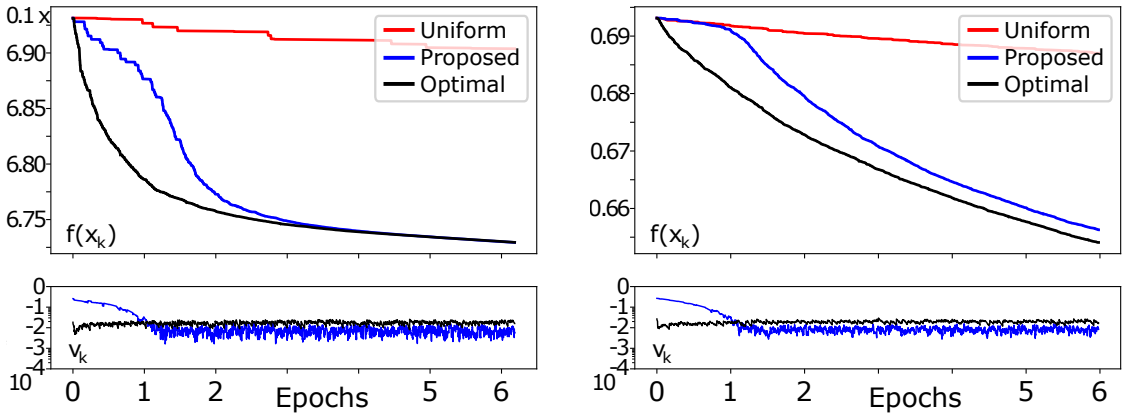
(a) *rcv1'*, *L*1 reg. (b) *rcv1'*, *L*2 reg.

Figure C.2: (CD, square loss) Fixed vs. adaptive sampling strategies, and dependence on stepsizes. With "big" $\alpha_k = v_k^{-1}$ and "small" $\alpha_k = \frac{1}{\mathrm{Tr}[\mathbf{L}]}$.



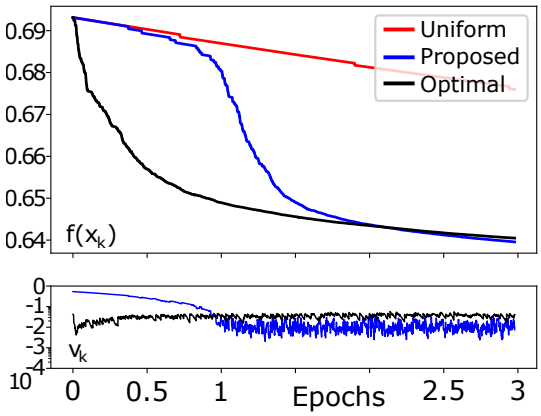
(a) *rcv1'*, *L*1 reg. (b) *real-sim'*, *L*2 reg.

Figure C.3: (CD, squared hinge loss) Function value vs. number of iterations for optimal stepsize $\alpha_k = v_k^{-1}$.

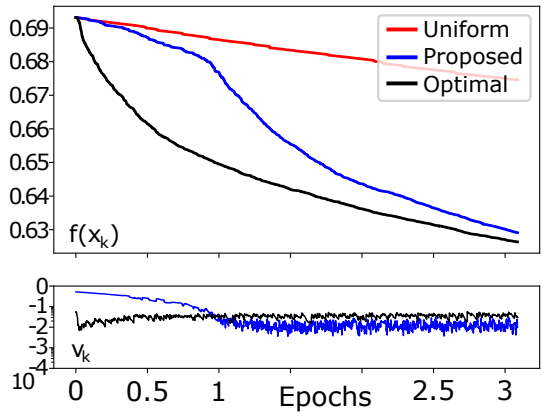


(a) *rcv1'*, $L1$ reg.

(b) *rcv1'*, $L2$ reg.

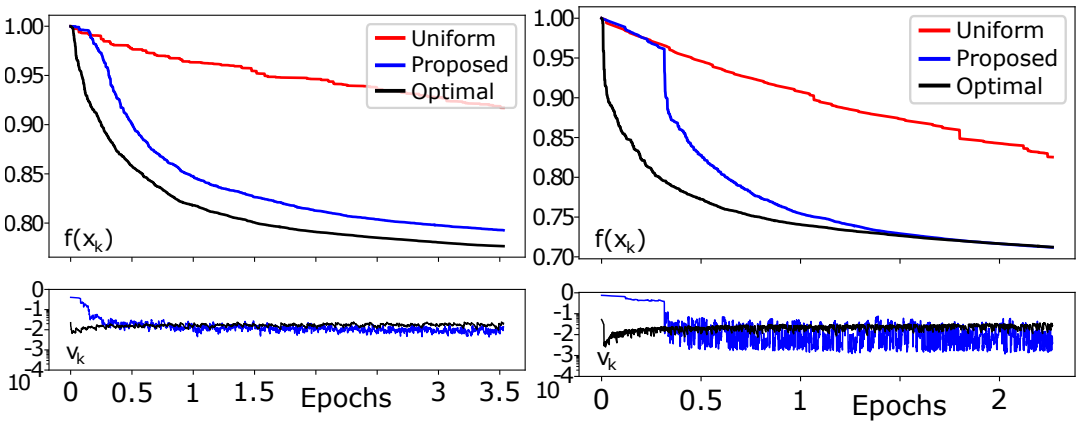


(c) *real-sim'*, $L1$ reg.



(d) *real-sim'*, $L2$ reg.

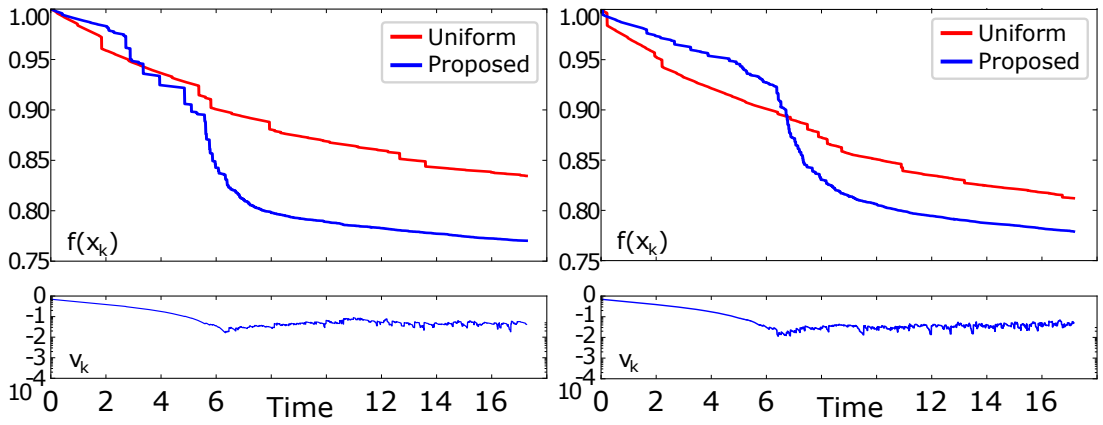
Figure C.4: (CD, logistic loss) Function value vs. number of iterations for different sampling strategies. Bottom: Evolution of the value v_k which determines the optimal stepsize ($\hat{\alpha}_k = v_k^{-1}$). The plots show the normalized values $\frac{v_k}{\mathrm{Tr}[\mathbf{L}]}$, i.e. the relative improvement over L_i-based importance sampling.



(a) *rcv1*, *L*1 reg.

(b) *real-sim*, *L*1 reg.

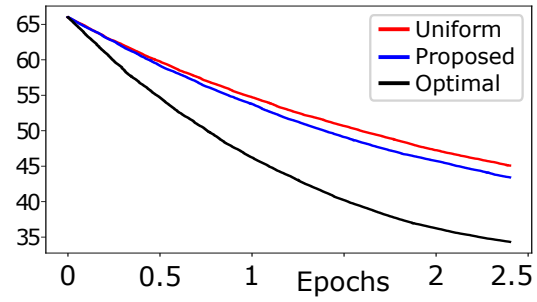
Figure C.5: (CD, square loss) Function value vs. number of iterations on the full datasets.



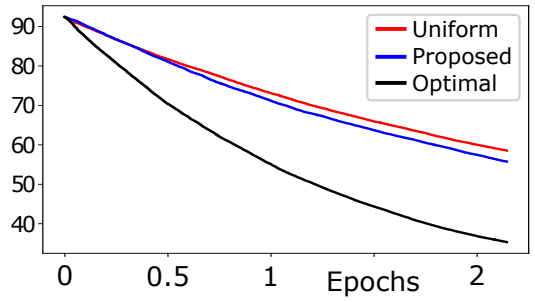
(a) *real-sim*, *L*1 reg.

(b) *real-sim*, *L*2 reg.

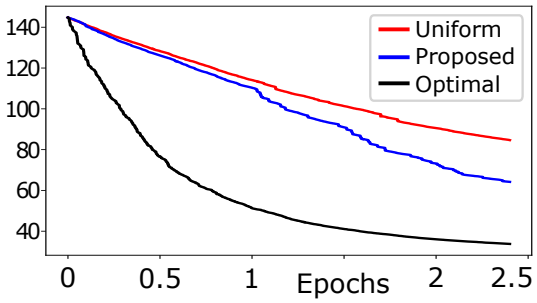
Figure C.6: (CD, square loss) Function value vs. clock time on the full datasets. (Data for the optimal sampling omitted, as this strategy is not competitive time-wise.)



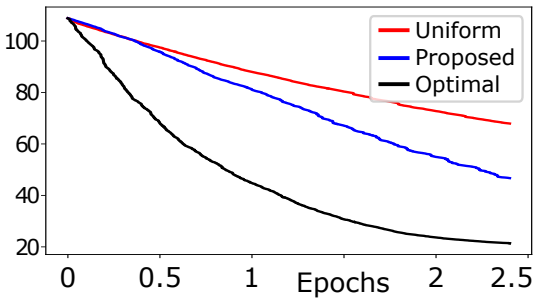
(a) *rcv1'*, $L1$ reg.



(b) *rcv1'*, $L2$ reg.

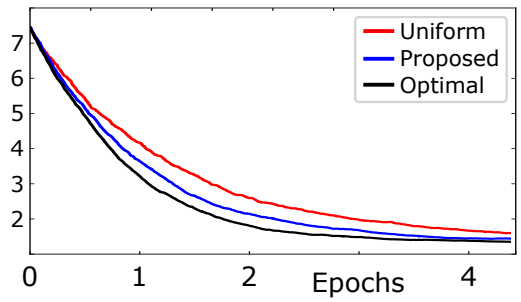


(c) *real-sim'*, $L1$ reg.



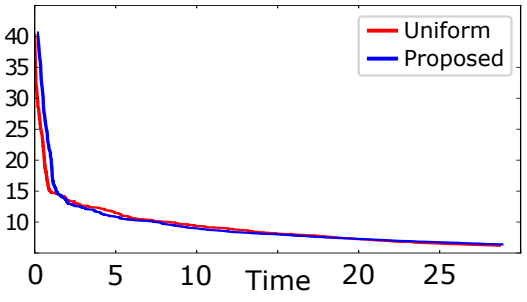
(d) *real-sim'*, $L2$ reg.

Figure C.7: (SGD, square loss) Function value vs. number of iterations.



(a) *news20'*, $L1$ reg.

Figure C.8: (SGD, square loss) Function value vs. number of iterations.



(a) *news20'*, $L1$ reg.

Figure C.9: (SGD square loss) Function value vs. clock time.

C.5 Empirical Evaluation

In this section we evaluate the empirical performance of our proposed adaptive sampling scheme on relevant machine learning tasks. In particular, we illustrate performance on generalized linear models with $L1$ and $L2$ regularization, as of the form (C.5),

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} h_i(\boldsymbol{a}_i^\top \boldsymbol{x}) + \lambda \cdot r(\boldsymbol{x}) \tag{C.13}$$

We use square loss, squared hinge loss as well as logistic loss for the data fitting terms h_i, and $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|_2^2$ for the regularizer $r(\boldsymbol{x})$. The datasets used in the evaluation are *rcv1*, *real-sim* and *news20*.[5] The *rcv1* dataset consists of 20,242 samples with 47,236 features, *real-sim* contains 72,309 datapoints and 20,958 features and *news20* contains 19,996 datapoints and 1,355,191 features. For all datasets we set unnormalized features with all the non-zero entries set to 1 (bag-of-words features). By *real-sim'* and *rcv1'* we denote a subset of the data chosen by randomly selecting 10,000 features and 10,000 datapoints. By *news20'* we denote a subset of the data chose by randomly selecting 15% of the features and 15% of the datapoints. A regularization parameter $\lambda = 0.1$ is used for all experiments.

Our results show the evolution of the optimization objective over time or number of epochs (an epoch corresponding to n individual updates). To compute safe lower and upper bounds we use the methods presented in Section C.4 with no special initialization, i.e. $\boldsymbol{\ell}_0 = \mathbf{0}_n$, $\boldsymbol{u}_0 = \boldsymbol{\infty}_n$.

Coordinate Descent. In Figure C.2 we compare the effect of the fixed stepsize $\alpha_k = \frac{1}{Ln}$ (denoted as "small") vs. the time varying optimal stepsize (denoted as "big") as discussed in Section C.2. Results are shown for optimal sampling $\boldsymbol{p}_k^\star$ (with optimal stepsize $\alpha_k(\boldsymbol{p}_k^\star)$, cf. Example C.2.1.2), our proposed sampling $\hat{\boldsymbol{p}}_k$ (with optimal stepsize $\alpha_k(\hat{\boldsymbol{p}}_k) = v_k^{-1}$, cf. (C.7)) and uniform sampling (with optimal stepsize $\alpha_k(\boldsymbol{p}_{\mathbf{L}}) = \frac{1}{Ln}$, as here $\mathbf{L} = L\mathbf{I}_n$, cf. Example C.2.1.1). As the experiment aligns with theory—confirming the advantage of the varying "big" stepsizes—we only show the results for Algorithms 5–7 in the remaining plots.

Performance for squared hinge loss, as well as logistic regression with $L1$ and $L2$ regularization is presented in Figure C.3 and Figure C.4 respectively. In Figures C.5 and C.6 we report the iteration complexity vs. accuracy as well as timing vs. accuracy results on the full dataset for coordinate descent with square loss and $L1$ (Lasso) and $L2$ regularization (Ridge).

Theoretical Sampling Quality. As part of the CD performance results in Figures C.2–C.6 we include an additional evolution plot on the bottom of each figure to illustrate the values v_k which determine the stepsize ($\hat{\alpha}_k = v_k^{-1}$) for the proposed Algorithm 6 (blue) and the optimal stepsizes of Algorithm 5 (black) which rely on the full gradient information. The plots show the normalized values $\frac{v_k}{\mathrm{Tr}[\mathbf{L}]}$, i.e. the relative improvement over L_i-based importance sampling. The results show that despite only relying on very loose safe gradient bounds, the proposed adaptive sampling is able to strongly benefit from the additional information.

[5] All data are available at www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Stochastic Gradient Descent. Finally, we also evaluate the performance of our approach when used within SGD with $L1$ and $L2$ regularization and square loss. In Figures C.7–C.8 we report the iteration complexity vs. accuracy results and in Figure C.9 the timing vs. accuracy results. The time units in Figures C.6 and C.9 are not directly comparable, as the experiments were conducted on different machines.

We observe that on all three datasets SGD with the optimal sampling performs only slightly better than uniform sampling. This is in contrast with the observations for CD, where the optimal sampling yields a significant improvement. Consequently, the effect of the proposed sampling is less pronounced in the three SGD experiments.

Summary. The main findings of our experimental study can be summarized as follows:

- **Adaptive importance sampling significantly outperforms fixed importance sampling in iterations and time.** The results show that (i) convergence in terms of iterations is almost as good as for the optimal (but not efficiently computable) gradient-based sampling and (ii) the introduced computational overhead is small enough to outperform fixed importance sampling in terms of total computation time.
- **Adaptive sampling requires adaptive stepsizes.** The adaptive stepsize strategies of Algorithms 5 and 6 allow for much faster convergence than conservative fixed-stepsize strategies. In the experiments, the measured value v_k was always significantly below the worst case estimate, in alignment with the observed convergence.
- **Very loose safe gradient bounds are sufficient.** Even the bounds derived from the the very naïve gradient information obtained by estimating scalar products resulted in significantly better sampling than using no gradient information at all. Further, no initialization of the gradient estimates is needed (at the beginning of the optimization process the proposed adaptive method performs close to the fixed sampling but accelerates after just one epoch).

C.6 Conclusion

In this paper we propose a safe adaptive importance sampling scheme for CD and SGD algorithms. We argue that optimal gradient-based sampling is theoretically well justified. To make the computation of the adaptive sampling distribution computationally tractable, we rely on safe lower and upper bounds on the gradient. However, in contrast to previous approaches, we use these bounds in a novel way: in each iteration, we formulate the problem of picking the optimal sampling distribution as a convex optimization problem and present an efficient algorithm to compute the solution. The novel sampling provably performs better than any fixed importance sampling—a guarantee which could not be established for previous samplings that were also derived from safe lower and upper bounds.

The computational cost of the proposed scheme is of the order $O(n \log n)$ per iteration—this is on many problems comparable with the cost to evaluate a single component (coordinate, sum-structure) of the gradient, and the scheme can thus be implemented at no extra computational cost. This is verified by timing experiments on real datasets.

We discussed one simple method to track the gradient information in GLMs during optimization. However, we feel that the machine learning community could profit from further research in that direction, for instance by investigating how such safe bounds can efficiently be maintained on more complex models. Our approach can immediately be applied when the tracking of the gradient is delegated to other machines in a distributed setting, like for instance in [4].

Proofs for Main Results

C.7 Efficiency of Adaptive Importance Sampling

In this section of the appendix we present the missing proofs from the main text and also add some additional comments.

C.7.1 In Coordinate Descent

In Section C.2 we only discussed the expected progress that can be proven using the quadratic upper bound (C.1). Here we show how to derive the convergence rate by the standard arguments.

Lemma C.7.1.1 (Proposed CD on strongly convex function—one step progress)**.** *Let $f:\mathbb{R}^n\to\mathbb{R}$ μ-strongly convex with coordinate-wise L_i-Lipschitz continuous gradient. Let $\boldsymbol{x}_k,\boldsymbol{x}_{k+1}\in\mathbb{R}^n$ denote two successive iterates generated by Algorithm 5, i.e. satisfying (C.2) and (C.4). Then*

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1})-f^\star \mid \boldsymbol{x}_k\right] \leq (f(\boldsymbol{x}_k)-f^\star)\cdot(1-\mu\,\alpha_k) \tag{C.14}$$

where $f^\star=\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$ and $\alpha_k=\alpha_k(\boldsymbol{p}_k)$ as in Lemma C.2.1.1.

Proof. By strong convexity

$$\frac{1}{2\mu}\|\nabla f(\boldsymbol{x}_k)\|_2^2 \geq f(\boldsymbol{x}_k)-f^\star, \tag{C.15}$$

and the claim follows directly from (C.4). □

For example for L_i-based importance sampling, $\alpha_k\equiv\frac{1}{\mathrm{Tr}[\mathbf{L}]}$ (Example C.2.1.1) and the statement simplifies to

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1})-f^\star \mid \boldsymbol{x}_k\right] \leq (f(\boldsymbol{x}_k)-f^\star)\cdot\left(1-\frac{\mu}{\mathrm{Tr}\left[\mathbf{L}\right]}\right) \tag{C.16}$$

in alignement with the results in [172, 230]. For the optimal sampling from Example C.2.1.2 it holds $\alpha_k(\boldsymbol{p}_k^\star)=\frac{\|\nabla f(\boldsymbol{x}_k)\|_2^2}{\|\sqrt{\mathbf{L}}\nabla f(\boldsymbol{x}_k)\|_1^2}$. For instance for $\mathbf{L}=L\cdot\mathbf{I}_n$ equation (C.14) simplifies to

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1})-f^\star \mid \boldsymbol{x}_k\right] \leq (f(\boldsymbol{x}_k)-f^\star)\cdot\left(1-\frac{\mu\|\nabla f(\boldsymbol{x}_k)\|_2^2}{L\|\nabla f(\boldsymbol{x}_k)\|_1^2}\right). \tag{C.17}$$

By Cauchy-Schwarz $\|\nabla f(\boldsymbol{x}_k)\|_2^2 \leq \|\nabla f(\boldsymbol{x}_k)\|_1^2 \leq n\|\nabla f(\boldsymbol{x}_k)\|_2^2$, hence the expected one step progress (C.17) is always as least as good as for uniform sampling (C.16) (we assumed $\mathbf{L} = L \cdot \mathbf{I}_n$), but the optimal sampling could yield an n times larger progress.

In Section C.2 we argued that it is natural to always chose the best possible stepsize in (C.3), i.e. $\alpha_k = \alpha_k(\boldsymbol{p}_k)$. Interestingly, even with a fixed stepsize (the worst case $\alpha_k = \frac{1}{\mathrm{Tr}[\mathbf{L}]}$) the optimal sampling $\boldsymbol{p}_k^\star$ has a slight advantage over the fixed importance sampling $\boldsymbol{p}_{\mathbf{L}}$. (This effect is also demonstrated in the experiments, cf. Figure C.2).

Remark C.7.1.1. *Let $\boldsymbol{p}_k^\star$ as in Example C.2.1.2. Then for suboptimal $\alpha_k = \frac{1}{\mathrm{Tr}[\mathbf{L}]}$ it holds*

$$\mathbb{E}_{i_k \sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k\right] \leq f(\boldsymbol{x}_k) - \frac{1}{2\,\mathrm{Tr}\left[\mathbf{L}\right]}\|\nabla f(\boldsymbol{x}_k)\|_2^2 \cdot \left(2 - \frac{\|\sqrt{\mathbf{L}}\nabla f(\boldsymbol{x}_k)\|_1^2}{\mathrm{Tr}\left[\mathbf{L}\right]\|\nabla f(\boldsymbol{x}_k)\|_2^2}\right). \quad \text{(C.18)}$$

The expression in the big bracket is bounded between 1 and $2 - \frac{1}{n}$. Hence the progress is always better then for the fixed distribution $\boldsymbol{p}_L$, but the speed-up is limited to a factor less than 2. In contrast, with the optimal $\alpha_k(\boldsymbol{p}_k^\star)$ the speed-up can reach a factor of n.

Proof. It suffices to just evaluate (C.3) with $\boldsymbol{p}_k^\star$ and $\alpha_k = \frac{1}{\mathrm{Tr}[\mathbf{L}]}$. □

Proof of Lemma C.2.1.1. For $c, d \geq 0$ consider $\min_\alpha -\alpha c + \frac{1}{2}\alpha^2 d$. This function is minimized for $\alpha^\star = \frac{c}{d}$ with value $-\frac{c^2}{2d} = -\frac{\alpha^\star c}{2}$. □

Proof of Example C.2.1.1. We evaluate (C.3) with $\boldsymbol{p}_{\mathbf{L}}$ and find

$$\mathbb{E}_{i_k \sim \boldsymbol{p}_k}\left[f(\boldsymbol{x}_{k+1}) \mid \boldsymbol{x}_k\right] \leq f(\boldsymbol{x}_k) - \alpha_k\|\nabla f(\boldsymbol{x}_k)\|_2^2 + \frac{1}{2}\alpha_k^2\,\mathrm{Tr}\left[\mathbf{L}\right]\|\nabla f(\boldsymbol{x}_k)\|_2^2 \quad \text{(C.19)}$$

which is minimized for $\alpha_k = \frac{1}{\mathrm{Tr}[\mathbf{L}]}$ as claimed. □

Proof of Example C.2.1.2. This is an immediate consequence of Lemma C.2.1.2. The provided estimates follow from $\|\boldsymbol{y}\|_2^2 \leq \|\boldsymbol{y}\|_1^2 \leq \frac{1}{L_{\min}}\|\sqrt{\mathbf{L}}\boldsymbol{y}\|_1^2$ and $\|\sqrt{\mathbf{L}}\boldsymbol{y}\|_1^2 \leq \mathrm{Tr}\left[\mathbf{L}\right]\|\boldsymbol{y}\|_2^2$ by Cauchy-Schwarz, for $\boldsymbol{y} \in \mathbb{R}^n$. □

Proof of Lemma C.2.1.2. Without loss of generality, assume $\mathbf{L} = \mathbf{I}$. The claim is verified by checking the optimality conditions: $-[\boldsymbol{x}]_i^2 + \lambda[\boldsymbol{p}]_i^2 = 0$ for all $i \in [n]$ and Lagrange multiplier $\lambda \geq 0$. Thus $\lambda = \frac{[\boldsymbol{x}]_i^2}{[\boldsymbol{p}]_i^2}$ for all $i \in [n]$ and this is satisfied for the proposed solution $\frac{|\boldsymbol{x}|}{\|\boldsymbol{x}\|_1} \in \Delta^n$. □

C.7.2 In SGD

SGD methods are applicable to objective functions which decompose as a sum

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x}). \quad \text{(C.20)}$$

Previous work [185, 272, 273] has argued that the gradient based sampling $[\tilde{\boldsymbol{p}}_k^\star]_i = \frac{\|\nabla f_i(\boldsymbol{x}_k)\|_2}{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}$ is also optimal in this setting. For the sake of completeness, we will now exhibit how this can be derived in the simplified setting where we assume f to be μ-strongly convex. The proof presented here is adapted from [166].

Theorem C.7.2.1. *Let $\mathcal{X} \in \mathbb{R}^d$ be a convex set, $f\colon \mathcal{X} \to \mathbb{R}$ μ-strongly convex with the structure $f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{x})$. Let $\{\boldsymbol{x}_k\}_{k\geq 0}$ denote a sequence of iterates satisfying*

$$\boldsymbol{x}_{k+1} := \Pi_{\mathcal{X}}\left(\boldsymbol{x}_k - \frac{\eta_k}{(n[\boldsymbol{p}_k]_{i_k})}\nabla f_{i_k}(\boldsymbol{x}_k)\right) \tag{C.21}$$

for stepsize $\eta_k = \frac{1}{\mu k}$, where index i_k is chosen at random $i_k \sim \boldsymbol{p}_k$ for probability vector $\boldsymbol{p}_k \in \Delta^n$ and $\Pi_{\mathcal{X}}$ denotes the orthogonal projection onto $\mathcal{X}$.

1. *If $[\boldsymbol{p}_k]_i \equiv \frac{1}{n}$ for all $i \in [n]$ and k (uniform sampling), then*

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{k=0}^T \boldsymbol{x}_k\right) - f^\star\right] \leq \frac{B_2}{\mu^2 T}(1+\log T)\,. \tag{C.22}$$

2. *If $[\boldsymbol{p}_k]_i = \frac{\|\nabla f_i(\boldsymbol{x}_k)\|_2}{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2} = [\tilde{\boldsymbol{p}}_k^\star]_i$, for $i \in [n]$ (optimal adaptive sampling), then*

$$\mathbb{E}\left[f\left(\frac{1}{T}\sum_{k=0}^T \boldsymbol{x}_k\right) - f^\star\right] \leq \frac{B_1^2}{\mu^2 T}(1+\log T)\,. \tag{C.23}$$

Where B_1 and B_2 are constants such that

$$\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x})\|_2}{n} \leq B_1 \qquad \frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x})\|_2^2}{n} \leq B_2 \qquad \forall \boldsymbol{x} \in \mathcal{X}. \tag{C.24}$$

It is clear that $\frac{B_2}{n} \leq B_1^2 \leq B_2$ from Cauchy-Schwarz. Comparing the upper bound we see that the importance sampling based approach might be n-times faster in convergence.

Proof. As orthogonal projections contract distances we have

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\|_2^2 \leq \|\boldsymbol{x}_k - \eta_k \frac{1}{n[\boldsymbol{p}_k]_{i_k}}\nabla f_{i_k}(\boldsymbol{x}_k) - \boldsymbol{x}^\star\|_2^2 \tag{C.25}$$

$$= \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2 - \frac{2\eta_k}{n[\boldsymbol{p}_k]_{i_k}}\langle \boldsymbol{x}_k - \boldsymbol{x}^\star, \nabla f_{i_k}(\boldsymbol{x}_k)\rangle + \frac{\eta_k^2}{n^2[\boldsymbol{p}_k]_{i_k}^2}\|\nabla f_{i_k}(\boldsymbol{x}_k)\|_2^2. \tag{C.26}$$

Thus

$$\mathbb{E}\left[\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\|_2^2 \mid \boldsymbol{x}_k\right] \leq \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2 - 2\eta_k\langle \boldsymbol{x}_k - \boldsymbol{x}^\star, \nabla f(\boldsymbol{x}_k)\rangle \tag{C.27}$$

$$+\sum_{i=1}^{n}\frac{\eta_k^2}{n^2[\boldsymbol{p}_k]_{i_k}}\|\nabla f_{i_k}(\boldsymbol{x}_k)\|_2^2. \tag{C.28}$$

It can be observed that the right hand side is minimized for probabilities given as follows:

$$[\tilde{\boldsymbol{p}}_k^\star]_i := \frac{\|\nabla f_i(\boldsymbol{x}_k)\|_2}{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}. \tag{C.29}$$

This justifies why these probabilities are denoted as optimal (cf. Section C.3 and [185, 272, 273]).

Hence the expression becomes :

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star\|_2^2 \mid \boldsymbol{x}_k\right] &\leq \|\boldsymbol{x}_k-\boldsymbol{x}^\star\|_2^2 - 2\eta_k\langle \boldsymbol{x}_k-\boldsymbol{x}^\star, \nabla f(\boldsymbol{x}_k)\rangle \\ &\quad + \eta_k^2\left(\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}{n}\right)^2 \qquad \text{(C.30)}\\ &\leq \|\boldsymbol{x}_k-\boldsymbol{x}^\star\|_2^2 - 2\eta_k\left[f(\boldsymbol{x}_k)-f^\star+\frac{\mu}{2}\|\boldsymbol{x}_k-\boldsymbol{x}^\star\|_2^2\right] \\ &\quad + \eta_k^2\left(\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}{n}\right)^2 \qquad \text{(C.31)}\end{aligned}$$

where the last inequality follows from strong convexity. Now we rearrange the terms and utilize the choice of the step size $\eta_k := \frac{1}{\mu k}$:

$$\begin{aligned}2\eta_k\left[f(\boldsymbol{x}_k)-f^\star\right] &\leq \eta_k^2\left(\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}{n}\right)^2 + (1-\mu\eta_k)\|\boldsymbol{x}_k-\boldsymbol{x}^\star\|_2^2 \\ &\quad - \mathbb{E}\left[\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star\|_2^2 \mid \boldsymbol{x}_k\right] \qquad \text{(C.32)}\\ \left[f(\boldsymbol{x}_k)-f^\star\right] &\leq \tfrac{1}{2}\eta_k\left(\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}{n}\right)^2 + \frac{1-\mu\eta_k}{2\eta_k}\|\boldsymbol{x}_k-\boldsymbol{x}^\star\|_2^2 \\ &\quad - \frac{1}{2\eta_k}\mathbb{E}\left[\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star\|_2^2 \mid \boldsymbol{x}_k\right] \qquad \text{(C.33)}\\ \left[f(\boldsymbol{x}_k)-f^\star\right] &\leq \frac{1}{2\mu k}\left(\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}{n}\right)^2 + \frac{\mu(k-1)}{2}\|\boldsymbol{x}_k-\boldsymbol{x}^\star\|_2^2 \\ &\quad - \frac{\mu k}{2}\mathbb{E}\left[\|\boldsymbol{x}_{k+1}-\boldsymbol{x}^\star\|_2^2 \mid \boldsymbol{x}_k\right] \qquad \text{(C.34)}\end{aligned}$$

If we compare the last equation and corresponding expression for uniform sampling then we see that the per iterate gain by the optimal sampling is approximately of the order of n due to the term $\left(\frac{\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2}{n}\right)^2$ in our case and $\frac{1}{n}\sum_{i=1}^n \|\nabla f_i(\boldsymbol{x}_k)\|_2^2$ in the uniform sampling.

We now take the expectation and sum the equation (C.34) for $k = 0, \ldots T$ and we get

the claim (this step is analogous as in [120]). □

C.8 Sampling

In this section we provide the remaining technical details regarding our proposed sampling scheme.

C.8.1 On the solution of the optimization problem

In the proof of Theorem C.3.2.1 we claimed that min and max in (C.7) can be interchanged. We will prove this now. This result will also be handy to describe the optimiality conditions of problem (C.7) in the proof of Theorem C.3.2.2 below.

Lemma C.8.1.1. *It holds*

$$v_k = \min_{\boldsymbol{p}\in\Delta^n}\max_{\boldsymbol{c}\in C_k}\frac{V(\boldsymbol{p},\boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \overset{(*)}{=} \max_{\boldsymbol{c}\in C_k}\min_{\boldsymbol{p}\in\Delta^n}\frac{V(\boldsymbol{p},\boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} = \max_{\boldsymbol{c}\in C_k}\frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2}\,. \tag{C.35}$$

Proof. The third equality follows directly from Lemma C.2.1.2. By transformation of the variable $[\boldsymbol{y}] := [\boldsymbol{c}]_i^2$ for $i \in [n]$ we can write the objective function as

$$\frac{V(\boldsymbol{p},\boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} = \frac{1}{\|\boldsymbol{y}\|_1}\cdot\sum_{i=1}^n\frac{L_i[\boldsymbol{y}]_i}{[\boldsymbol{p}]_i} =: \psi(\boldsymbol{p},\boldsymbol{y})\,. \tag{C.36}$$

Let $Y \subset \mathbb{R}^n_{\geq 0}$ denote appropriately transformed set of constraints, $Y := C_k^2$. To prove $(*)$ we will now rely on Sion's minimax theorem [116, 222]. The function $\psi(\cdot,\boldsymbol{y})$ is convex in $\boldsymbol{p}\in\Delta^n$ and Δ^n is a compact convex subset of $\mathbb{R}^n$. Clearly, Y is convex, and in order to apply the theorem it remains to show that $\psi(\boldsymbol{p},\cdot)$ is quasi-concave. For establish this, it is enough to show that the level sets of $\psi(\boldsymbol{p},\cdot)$ are convex. Let $\boldsymbol{u},\boldsymbol{v}\in Y$ with $\psi(\boldsymbol{p},\boldsymbol{u})\geq\beta$, $\psi(\boldsymbol{p},\boldsymbol{v})\geq\beta$ for some $\beta\geq 0$. Then for any $\lambda\in[0,1]$ it holds $\psi(\boldsymbol{p},\lambda\boldsymbol{u}+(1-\lambda)\boldsymbol{v})\geq\beta$ as is verified as follows:

$$0 \leq \lambda\underbrace{\left[\left(\sum_{i=1}^n\frac{[\boldsymbol{u}]_iL_i}{[\boldsymbol{p}]_i}\right)-\beta\|\boldsymbol{u}\|_1\right]}_{\geq 0} + (1-\lambda)\underbrace{\left[\left(\sum_{i=1}^n\frac{[\boldsymbol{v}]_iL_i}{[\boldsymbol{p}]_i}\right)-\beta\|\boldsymbol{v}\|_1\right]}_{\geq 0} \tag{C.37}$$

$$= \left(\sum_{i=1}^n\frac{\lambda[\boldsymbol{u}]_iL_i+(1-\lambda)[\boldsymbol{v}]_iL_i}{[\boldsymbol{p}]_i}\right)-\beta\left(\underbrace{\lambda\|\boldsymbol{u}\|_1+(1-\lambda)\|\boldsymbol{v}\|_1}_{=\|\lambda\boldsymbol{u}+(1-\lambda)\boldsymbol{v}\|_1}\right). \tag{C.38}$$

This proves the claim. □

Proof of Theorem C.3.2.2 – Part I: Structure of the solution. We will now proof that $\boldsymbol{c} \in C_k$ of the form

$$[\boldsymbol{c}]_i = \begin{cases} [\boldsymbol{u}_k]_i & \text{if } [\boldsymbol{u}_k]_i \leq \sqrt{L_i}m\,, \\ [\boldsymbol{\ell}_k]_i & \text{if } [\boldsymbol{\ell}_k]_i \geq \sqrt{L_i}m\,, \\ \sqrt{L_i}m & \text{otherwise,} \end{cases} \qquad \forall i \in [n]\,, \tag{C.9}$$

where $m = \|\boldsymbol{c}\|_2^2 \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^{-1}$ and probabilities $\boldsymbol{p} = \frac{\sqrt{\mathbf{L}}\boldsymbol{c}}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}$ solve the optimization problem (C.7). By Lemma C.8.1.1 is suffices to consider

$$\arg\max_{\boldsymbol{c}\in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2} = \arg\max_{\boldsymbol{c}\in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}{\|\boldsymbol{c}\|_2}\,. \tag{C.39}$$

We now write the Lagrangian of the problem on the right:

$$\mathcal{L}(\boldsymbol{c},\boldsymbol{\lambda},\boldsymbol{\mu}) = \frac{\|\sqrt{\boldsymbol{L}}\boldsymbol{c}\|_1}{\|\boldsymbol{c}\|_2} + \sum_{i=1}^n [\boldsymbol{\lambda}]_i([\boldsymbol{u}_k]_i - [\boldsymbol{c}]_i) + \sum_{i=1}^n [\boldsymbol{\mu}]_i([\boldsymbol{c}]_i - [\boldsymbol{\ell}_k]_i) \tag{C.40}$$

and derive the KKT conditions:

$$\frac{\partial \mathcal{L}}{\partial [\boldsymbol{c}]_i} = \frac{\sqrt{L_i}\|\boldsymbol{c}\|_2^2 - [\boldsymbol{c}]_i\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}{\|\boldsymbol{c}\|_2^3} - [\boldsymbol{\lambda}]_i + [\boldsymbol{\mu}_i] \leq 0\,; \qquad [\boldsymbol{c}]_i \geq 0\,; \qquad [\boldsymbol{c}]_i \frac{\partial \mathcal{L}}{\partial [\boldsymbol{c}]_i} = 0\,; \tag{C.41}$$

$$\frac{\partial \mathcal{L}}{\partial [\boldsymbol{\lambda}]_i} = [\boldsymbol{u}_k]_i - [\boldsymbol{c}]_i \geq 0\,; \qquad [\boldsymbol{\lambda}]_i \geq 0\,; \qquad [\boldsymbol{\lambda}]_i \frac{\partial \mathcal{L}}{\partial [\boldsymbol{\lambda}]_i} = 0\,; \tag{C.42}$$

$$\frac{\partial \mathcal{L}}{\partial [\boldsymbol{\mu}]_i} = [\boldsymbol{c}]_i - [\boldsymbol{\ell}_k]_i \geq 0\,; \qquad [\boldsymbol{\mu}]_i \geq 0\,; \qquad [\boldsymbol{\mu}]_i \frac{\partial \mathcal{L}}{\partial [\boldsymbol{\mu}]_i} = 0\,; \tag{C.43}$$

For all non-binding constraints, the Lagrange multipliers are zero, and hence from the topmost equation see that it must hold $\sqrt{L_i}\|\boldsymbol{c}\|_2^2 - [\boldsymbol{c}]_i\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1 = 0$ (or equivalently $[\boldsymbol{c}]_i = \sqrt{L_i}m$) for all variables with non-binding constraints. Furthermore if $[\boldsymbol{c}]_i < \sqrt{L_i}m$), then $[\boldsymbol{\lambda}]_i$ must be positive, and hence the upper bound must be binding. And vice versa for the lower bounds. Clearly, the given $\boldsymbol{c}$ in (C.9) satisfies these conditions. By Lemma C.2.1.2 we also have $\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{c}) = \frac{\sqrt{\mathbf{L}}\boldsymbol{c}}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1}$ as claimed. □

C.8.2 Algorithm

Here we argue on the correctness of Algorithm 8.

Proof of Theorem C.3.2.2 – Part II: Algorithm. We now show that Algorithm 8 indeed

computes a solution of the form (C.9). For this, we have to show that performed optimization steps—the sorting in line 2 and the efficient comparisons in line 4 and 6—do not hamper the correctness for the algorithm. For clarity, we now introduce iteration indices for the quantities $\boldsymbol{c}_t$ (see main text), and m_t.

Suppose the check in line 4 is true, i.e. $[\boldsymbol{\ell}^{\text{sort}}]_\ell > m_t$, where $m_t = \frac{\|\boldsymbol{c}_t\|_2^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1}$. Now we show $m_{t+1} \in [m_t, [\boldsymbol{\ell}^{\text{sort}}]_\ell]$. The claim can easily be checked. Let L_τ denote the corresponding L_i-value, i.e. it holds $\sqrt{L_\tau}[\boldsymbol{\ell}^{\text{sort}}]_\ell = [\boldsymbol{\ell}_k]_\tau$.

By assumption $[\boldsymbol{\ell}^{\text{sort}}]_\ell > \frac{\|\boldsymbol{c}_t\|_2^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1}$, thus $[\boldsymbol{\ell}^{\text{sort}}]_\ell \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell^2 > \|\boldsymbol{c}_t\|_2^2 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell^2$ and consequently $m_{t+1} = \frac{\|\boldsymbol{c}_t\|_2^2 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell} < [\boldsymbol{\ell}^{\text{sort}}]_\ell$. For to show $m_{t+1} > m_t$ we make use of the assumption $[\boldsymbol{\ell}^{\text{sort}}]_\ell > \frac{\|\boldsymbol{c}_t\|_2^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1}$ in a similar way. Clearly, $L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell^2 \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 > L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell \cdot \|\boldsymbol{c}_t\|_2^2$ and thus $\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 \cdot \|\boldsymbol{c}_t\|_2^2 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell^2 \cdot \|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 > \|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 \cdot \|\boldsymbol{c}_t\|_2^2 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell \cdot \|\boldsymbol{c}_t\|_2^2$ which implies $m_{t+1} = \frac{\|\boldsymbol{c}_t\|_2^2 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1 + L_\tau[\boldsymbol{\ell}^{\text{sort}}]_\ell} > \frac{\|\boldsymbol{c}_t\|_2^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}_t\|_1} = m_t$.

The inequality $m_{t+1} \leq [\boldsymbol{\ell}^{\text{sort}}]_\ell$ implies that the chosen update does not interfere with any previously made decisions regarding lower bounds, as $m_{t+1} \leq [\boldsymbol{\ell}^{\text{sort}}]_i$ for $i = \ell+1, \dots, n$ (with this notation, $n+1, \dots, n$ just denotes the empty set). The opposite inequality $m_{t+1} \geq m_t$ implies that the chosen update does not interfere with any previously made decisions regarding upper bounds, as $m_{t+1} \geq [\boldsymbol{u}^{\text{sort}}]_i$ for $i = 1, \dots, u-1$.

If line 6 is executed and the check is true, i.e. $[\boldsymbol{u}^{\text{sort}}]_u < m_t$, then it can be shown that $m_{t+1} \in [[\boldsymbol{u}^{\text{sort}}]_u, m_t]$ by analogous arguments. □

C.8.3 Competitive Ratio

Proof of Lemma C.3.2.3. The proof of this lemma is immediate from the definition:

$$\rho_k = \max_{\boldsymbol{c} \in C_k} \frac{V(\hat{\boldsymbol{p}}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \cdot \frac{\|\boldsymbol{c}\|_2^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2} \leq \max_{\boldsymbol{c} \in C_k} \frac{V(\hat{\boldsymbol{p}}, \boldsymbol{c})}{\|\boldsymbol{c}\|_2^2} \cdot \max_{\boldsymbol{c} \in C_k} \frac{\|\boldsymbol{c}\|_2^2}{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2} \leq \frac{v_k}{w_k}. \tag{C.44}$$

where $w_k := \min_{\boldsymbol{c} \in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2}$. The claimed upper bound $w_k \leq v_k$ follows by the observation $v_k \overset{(C.35)}{=} \max_{\boldsymbol{c} \in C_k} \frac{\|\sqrt{\mathbf{L}}\boldsymbol{c}\|_1^2}{\|\boldsymbol{c}\|_2^2}$. □

Proof of Lemma C.3.2.4. As we have relative accuracy, it holds $[C_k]_i \cap \gamma[C_k]_i = \emptyset$ and $\gamma^{-1}[C_k]_i \cap [C_k]_i = \emptyset$, for all $i \in [n]$. Let $\boldsymbol{c}^\star \in C_k$ denote the vector for which the maximum is attained and let $\hat{\boldsymbol{c}} \in C_k$ be such that $\hat{\boldsymbol{p}} = \frac{\sqrt{\mathbf{L}}\hat{\boldsymbol{c}}}{\|\sqrt{\mathbf{L}}\hat{\boldsymbol{c}}\|_1}$. It holds $V(\hat{\boldsymbol{p}}, \boldsymbol{c}^\star) \leq V(\hat{\boldsymbol{p}}, \boldsymbol{c})$ for all $\boldsymbol{c} \in \gamma C_k$ by monotonicity in each coordinate, especially $V(\hat{\boldsymbol{p}}, \boldsymbol{c}^\star) \leq V(\hat{\boldsymbol{p}}, \gamma\hat{\boldsymbol{c}})$. And similarly $\|\sqrt{\mathbf{L}}\boldsymbol{c}^\star\|_1^2 \geq \|\gamma^{-1}\sqrt{\mathbf{L}}\hat{\boldsymbol{c}}\|_1^2$. Thus

$$\rho_k \leq \frac{V(\hat{\boldsymbol{p}}, \gamma\hat{\boldsymbol{c}})}{\|\gamma^{-1}\sqrt{\mathbf{L}}\hat{\boldsymbol{c}}\|_1^2} = \frac{\gamma^2 V(\hat{\boldsymbol{p}}, \hat{\boldsymbol{c}})}{\gamma^{-2}\|\sqrt{\mathbf{L}}\hat{\boldsymbol{c}}\|_1^2} = \frac{\gamma^2}{\gamma^{-2}}. \tag{C.45}$$

which proves the claim. □

C.9 Safe Gradient Bounds in the Proximal Setting

Proof of Lemma C.4.0.1. Observe

$$\begin{aligned}
\nabla f_i(\boldsymbol{x}_{k+1}) - \nabla f_i(\boldsymbol{x}_k) &= \nabla_x h_i(\boldsymbol{a}_i^\top \boldsymbol{x}_{k+1}) - \nabla_x h_i(\boldsymbol{a}_i^\top \boldsymbol{x}_k) \\
&= \boldsymbol{a}_i\big(\nabla h_i(\boldsymbol{a}_i^\top \boldsymbol{x}_{k+1}) - \nabla h_i(\boldsymbol{a}_i^\top \boldsymbol{x}_k)\big) \\
&= \boldsymbol{a}_i\big(\boldsymbol{a}_i^\top \boldsymbol{x}_{k+1} - \boldsymbol{a}_i^\top \boldsymbol{x}_k\big)\nabla^2 h_i(\boldsymbol{a}_i^\top \tilde{\boldsymbol{x}}) \qquad \text{(C.46)} \\
&= \boldsymbol{a}_i\big(\boldsymbol{a}_i^\top (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)\nabla^2 h_i(\boldsymbol{a}_i^\top \tilde{\boldsymbol{x}})\big) \\
&= \boldsymbol{a}_i\big(\boldsymbol{a}_i^\top \gamma_k \boldsymbol{a}_{i_k}\nabla^2 h_i(\boldsymbol{a}_i^\top \tilde{\boldsymbol{x}})\big) \\
&= \gamma_k \nabla^2 h_i(\boldsymbol{a}_i^\top \tilde{\boldsymbol{x}})\langle \boldsymbol{a}_i, \boldsymbol{a}_{i_k}\rangle \boldsymbol{a}_i \;\; \forall\, i \neq i_k\,,
\end{aligned}$$

Equation (C.46) comes from the mean value theorem which says for continuous function f in closed intervals $[a,b]$ and differentiable on open intervals (a,b), there exists a point c in (a,b) such that :

$$f'(c) = \frac{f(b) - f(a)}{b - a}\,. \qquad \text{(C.47)}$$

□

In Section C.4 we have discussed practical safe upper and lower bounds $\boldsymbol{u}, \boldsymbol{\ell}$ that can be maintained efficiently during optimization, also for the SGD setting (finite sum objective). We now argue that such bounds can also be extended to proximal SGD settings.

We see from Lemma C.4.0.1 that tracking the norm of the gradient of each function can be done easily for simple updates as given in Lemma C.4.0.1. The approximate update of the component wise gradient norms for more composite problems can also be done by a little modification, but it is definitely not as trivial as in the case of coordinate descent. For example, consider a proximal type of update as $\boldsymbol{x}_{k+1} = prox_{\eta_k g}\big(\boldsymbol{x}_k - \eta_k \cdot \boldsymbol{a}_{i_k}\nabla f_{i_k}(\boldsymbol{a}_{i_k}^\top \boldsymbol{x}_k)\big)$ which implies that $\boldsymbol{x}_{k+1} \in \boldsymbol{x}_k - \eta_k \cdot \boldsymbol{a}_{i_k}\nabla f_{i_k}(\boldsymbol{a}_{i_k}^\top \boldsymbol{x}_k) - \eta_k \partial g(\boldsymbol{x}_{k+1})$ and thus $\boldsymbol{x}_{k+1} \in \boldsymbol{x}_k + \gamma_k \cdot \boldsymbol{a}_{i_k} - \eta_k \partial g(\boldsymbol{x}_{k+1})$. If we denote the progress made in the k-th iteration of the algorithm as δ_k then the progress equals $\delta_k = \gamma_k\, \boldsymbol{a}_{i_k} - \eta_k \boldsymbol{\alpha}_k$ where $\boldsymbol{\alpha}_k \in \partial g(\boldsymbol{x}_{k+1})$. To approximate the gradient we will need to compute two dot products. The first one is $\langle \boldsymbol{a}_i, \boldsymbol{a}_{i_k}\rangle$ and the second one is $\langle \boldsymbol{a}_{i_k}, \boldsymbol{\alpha}_k\rangle$. Since $\boldsymbol{\alpha}_k$ is usually small, hence even approximating $\langle \boldsymbol{a}_{i_k}, \boldsymbol{\alpha}_k\rangle$ with $\|\boldsymbol{a}_{i_k}\|\|\boldsymbol{\alpha}_k\|$ doesn't affect the upper and bounds too much and the main contribution in error comes from the approximation of the scalar product $\langle \boldsymbol{a}_i, \boldsymbol{a}_{i_k}\rangle$.

Appendix D

On Matching Pursuit and Coordinate Descent

Francesco Locatello[*1,3], Anant Raj[*1], Sai Praneeth Karimireddy[2], Gunnar Rätsch[3], Bernhard Schölkopf[1], Sebastian U. Stich[2], Martin Jaggi[2]

1 – MPI for Intelligent Systems, Tübingen

2 – EPFL

3 – ETH Zürich

* – denotes Equal Contribution

Abstract

Two popular examples of first-order optimization methods over linear spaces are coordinate descent and matching pursuit algorithms, with their randomized variants. While the former targets the optimization by moving along coordinates, the latter considers a generalized notion of directions. Exploiting the connection between the two algorithms, we present a unified analysis of both, providing affine invariant sublinear $\mathcal{O}(1/t)$ rates on smooth objectives and linear convergence on strongly convex objectives. As a byproduct of our affine invariant analysis of matching pursuit, our rates for steepest coordinate descent are the tightest known. Furthermore, we show the first accelerated convergence rate $\mathcal{O}(1/t^2)$ for matching pursuit and steepest coordinate descent on convex objectives.

D.1 Introduction

In this paper we address the following convex optimization problem:

$$\min_{\mathbf{x}\in\operatorname{lin}(\mathcal{A})} f(\mathbf{x})\,, \tag{D.1}$$

where f is a convex function. The minimization is over a linear space, which is parametrized as the set of linear combinations of elements from a given set $\mathcal{A}$. These elements of $\mathcal{A}$ are called *atoms*. In the most general setting, $\mathcal{A}$ is assumed to be a compact but not necessarily finite subset of a Hilbert space, i.e., a linear space equipped with an inner product, complete in the corresponding norm. Problems of the form (D.1) are tackled by a multitude of first-order optimization methods and are of paramount interest in the machine learning community [154, 155, 211, 212, 241].

Traditionally, matching pursuit (MP) algorithms were introduced to solve the inverse problem of representing a measured signal by a sparse combination of atoms from an over-complete basis [149]. In other words, the solution of the optimization problem (D.1) is formed as a linear combination of few of the elements of the atom set $\mathcal{A}$ – i.e. a sparse approximation. At each iteration, the MP algorithm picks a direction from $\mathcal{A}$ according to the gradient information, and takes a step. This procedure is not limited to atoms of fixed dimension. Indeed, $\langle(\rangle \mathcal{A})$ can be an arbitrary linear subspace of the ambient space and we are interested in finding the minimizer of f only on this domain, see e.g. [78]. Conceptually, MP stands in the middle between coordinate descent (CD) and gradient descent, as the algorithm is allowed to descend the function along a prescribed set of directions which does not necessarily correspond to coordinates. This is particularly important for machine learning applications as it translates to a sparse representation of the iterates in terms of the elements of $\mathcal{A}$ while maintaining the convergence guarantees [122, 142].

The first analysis of the MP algorithm in the optimization sense to solve the template (D.1) without incoherence assumptions was done by [143]. To prove convergence, they exploit the connection between MP and the Frank-Wolfe (FW) algorithm [68], a popular projection-free algorithm for the constrained optimization case. On the other hand, steepest coordinate descent is a special case of MP (when the atom set is the L1 ball). This is particularly important as the CD rates can be deduced from the MP rates. Furthermore, the literature on coordinate descent is currently much richer than the one on MP. Therefore, understanding the connection of the two classes of CD and MP-type algorithms is a main goal of this paper, and results in benefits for both sides of the spectrum. In particular, the contributions of this paper are:

- We present an affine invariant convergence analysis for Matching Pursuit algorithms solving (D.1). Our approach is tightly related to the analysis of coordinate descent and relies on the properties of the atomic norm in order to generalize from coordinates to atoms.

- Using our analysis, we present the tightest known linear and sublinear convergence rates for steepest coordinate descent, improving the constants in the rates of [181, 231].

- We discuss the convergence guarantees of Random Pursuit (RP) methods which we analyze through the lens of MP. In particular, we present a unified analysis of both

MP and RP which allows us to carefully trade off the use of (approximate) steepest directions over random ones.

- We prove the first known accelerated rate for MP, as well as for steepest coordinate descent. As a consequence, we also demonstrate an improvement upon the accelerated random CD rate by performing a steepest coordinate update instead.

Related Work: Matching Pursuit was introduced in the context of sparse recovery [149], and later, fully corrective variants similar to the one used in Frank-Wolfe [90, 112, 119] were introduced under the name of orthogonal matching pursuit [41, 245]. The classical literature for MP-type methods is typically focused on recovery guarantees for sparse signals and the convergence depends on very strong assumptions (from an optimization perspective), such as incoherence or restricted isometry properties of the atom set [52, 245]. Convergence rates with incoherent atom sets are predented in [82, 178, 235, 236]. Also boosting can be seen as a generalized coordinate descent method over a hypothesis class [154, 197].

The idea of following a prescribed set of directions also appears in the field of derivative free methods. For instance, the early method of Pattern-Search [55, 92, 244] explores the search space by probing function values along predescribed directions ("patterns" or atoms). This method is in some sense orthogonal to the approach here: by probing the function values along all atoms, one aims to find a direction along which the function decreases (and the absolute value of the scalar product with the gradient is potentially small). MP does not access the function value, but computes the gradient and then picks the atom with the smallest scalar product with the gradient, and then moves to a point where the function value decreases.

The description of random pursuit appears already in the work of Mutseniyeks and Rastrigin [160] and was first analyzed by Karmanov [109, 110], Zieliński and Neumann [274]. More recently random pursuit was revisited in [228, 229].

Acceleration of first-order methods was first developed in [168]. An accelerated CD method was described in [171]. The method was extended in [131] for non-uniform sampling, and later in [228] for optimization along arbitrary random directions. Recently, optimal rates have been obtained for accelerated CD [13, 176]. A close setup is the accelerated algorithm presented in [63], which minimizes a composite problem of a convex function on $\mathbb{R}^n$ with a non-smooth regularizer which acts as prior for the structure of the space. Contrary to our setting, the approach is restricted to the atoms being linearly independent. Simultaneously at ICML 2018, Lu *et al.* [145] propose an accelerated rate for the semi-greedy coordinate descent which is a special case of our accelerated MP algorithm.

Notation: Given a non-empty subset $\mathcal{A}$ of some Hilbert space, let $\operatorname{conv}(\mathcal{A})$ be the convex hull of $\mathcal{A}$, and let $\operatorname{lin}(\mathcal{A})$ denote its linear span. Given a closed set $\mathcal{A}$, we call its diameter $\operatorname{diam}(\mathcal{A}) = \max_{\mathbf{z}_1,\mathbf{z}_2\in\mathcal{A}} \|\mathbf{z}_1 - \mathbf{z}_2\|$ and its radius $\operatorname{radius}(\mathcal{A}) = \max_{\mathbf{z}\in\mathcal{A}} \|\mathbf{z}\|$.

$\|\mathbf{x}\|_{\mathcal{A}} := \inf\{c > 0 \colon \mathbf{x} \in c \cdot \operatorname{conv}(\mathcal{A})\}$ is the atomic norm of $\mathbf{x}$ over a set $\mathcal{A}$ (also known as the gauge function of $\operatorname{conv}(\mathcal{A})$). We call a subset $\mathcal{A}$ of a Hilbert space symmetric if it is closed under negation.

D.2 Revisiting Matching Pursuit

Let $\mathcal{H}$ be a Hilbert space with associated inner product $\langle \mathbf{x}, \mathbf{y} \rangle$, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{H}$. The inner product induces the norm $\|\mathbf{x}\|^2 := \langle \mathbf{x}, \mathbf{x} \rangle$, $\forall \mathbf{x} \in \mathcal{H}$. Let $\mathcal{A} \subset \mathcal{H}$ be a compact and symmetric set (the "set of atoms" or dictionary) and let $f \colon \mathcal{H} \to \mathbb{R}$ be convex and L-smooth (L-Lipschitz gradient in the finite dimensional case). If $\mathcal{H}$ is an infinite-dimensional Hilbert space, then f is assumed to be Fréchet differentiable. In each iteration, MP queries a linear

Algorithm 9 Generalized Matching Pursuit

init $\mathbf{x}_0 \in \operatorname{lin}(\mathcal{A})$
for $t = 0 \dots T$
  Find $\mathbf{z}_t := (\text{Approx-})\operatorname{LMO}_{\mathcal{A}}(\nabla f(\mathbf{x}_t))$
  $\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle}{L \|\mathbf{z}_t\|^2} \mathbf{z}_t$
end for

minimization oracle (LMO) to find the steepest descent direction among the set $\mathcal{A}$:

$$\operatorname{LMO}_{\mathcal{A}}(\mathbf{y}) := \arg\min_{\mathbf{z} \in \mathcal{A}} \langle \mathbf{y}, \mathbf{z} \rangle , \tag{D.2}$$

for a given query vector $\mathbf{y} \in \mathcal{H}$. This key subroutine is shared with the Frank-Wolfe method [68, 96] as well as steepest coordinate descent. Indeed, finding the steepest coordinate is equivalent to minimizing Equation D.2. The MP update step minimizes a quadratic upper bound $g_{\mathbf{x}_t}(\mathbf{x}) = f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{x}_t\|^2$ of f at $\mathbf{x}_t$ on the direction $\mathbf{z}$ returned by the LMO, where L is an upper bound on the smoothness constant of f with respect to the Hilbert norm $\|\cdot\|$. For $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|^2$, $\mathbf{y} \in \mathcal{H}$, Algorithm 9 recovers the classical MP algorithm [149].

The LMO. Greedy and projection-free optimization algorithms such as Frank-Wolfe and Matching Pursuit rely on the property that the result of the LMO is a descent direction, which is translated to an *alignment assumption* of the search direction returned by the LMO (i.e., $\mathbf{z}_t$ in Algorithm 9) and the gradient of the objective at the current iteration (see [142], [188, third premise] and [244, Lemma 12 and proof of Proposition 6.4]). Specifically, for Algorithm 9, a symmetric atom set $\mathcal{A}$ ensures that $\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle < 0$, as long as $\mathbf{x}_t$ is not optimal yet. Indeed, we then have that $\min_{\mathbf{z} \in \mathcal{A}} \langle \nabla f(\mathbf{x}_t), \mathbf{z} \rangle = \min_{\mathbf{z} \in \operatorname{conv}(\mathcal{A})} \langle \nabla f(\mathbf{x}_t), \mathbf{z} \rangle < 0$ where the inequality comes from symmetry as $\mathbf{z} = \mathbf{0} \in \operatorname{conv}(\mathcal{A})$. Note that an alternative sufficient condition instead of symmetry is that $\mathcal{A}$ is the atomic ball of a norm (the so called atomic norm [39]).

Steepest Coordinate Descent. In the case when $\mathcal{A}$ is the L1-ball, the MP algorithm becomes identical to steepest coordinate descent [171]. Indeed, due to symmetry of $\mathcal{A}$, one can rewrite the LMO problem as $i_t = \arg\max_i |\nabla_i f(x)|$, where ∇_i is the i-th component of the gradient, i.e. $\langle \nabla f(x), \mathbf{e}_i \rangle$ with $\mathbf{e}_i$ being one of the natural vectors. Then the update step can be written as:

$$\mathbf{x}_{t+1} := \mathbf{x}_{t+1} - \frac{1}{L} \nabla_{i_t} f(\mathbf{x}_t) \mathbf{e}_i \,.$$

Note that by assuming a symmetric atom set and solving the LMO problem as defined in (D.2) the steepest atom is aligned with the negative gradient, therefore the positive stepsize $-\frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle}{L}$ decreases the objective.

Approximate linear oracles. Exactly solving the LMO defined in (D.2) can be costly in practice, both in the MP and the CD setting, as $\mathcal{A}$ can contain (infinitely) many atoms. On the other hand, approximate versions can be much more efficient. Algorithm 9 allows for an *approximate* LMO. Different notions of such a LMO were explored for MP and OMP in [149] and [245], respectively, for the Frank-Wolfe framework in [96, 122] and for coordinate descent [231]. For given quality parameter $\delta \in (0, 1]$ and given direction $\mathbf{d} \in \mathcal{H}$, the approximate LMO for Algorithm 9 returns a vector $\tilde{\mathbf{z}} \in \mathcal{A}$ such that:

$$\langle \mathbf{d}, \tilde{\mathbf{z}} \rangle \leq \delta \langle \mathbf{d}, \mathbf{z} \rangle \,, \tag{D.3}$$

relative to $\mathbf{z} = \mathrm{LMO}_{\mathcal{A}}(\mathbf{d})$ being an exact solution.

D.2.1 Affine Invariant Algorithm

In this section, we will present our new affine invariant algorithm for the optimization problem (D.1). Hence, we first explain in Definition D.2.1.1 that what does it mean for an optimization algorithm to be affine invariant:

Definition D.2.1.1. An optimization method is called *affine invariant* if it is invariant under affine transformations of the input problem: If one chooses any re-parameterization of the domain $\mathcal{Q}$ by a *surjective* linear or affine map $\mathbf{M} : \hat{\mathcal{Q}} \to \mathcal{Q}$, then the "old" and "new" optimization problems $\min_{\mathbf{x} \in \mathcal{Q}} f(\mathbf{x})$ and $\min_{\hat{\mathbf{x}} \in \hat{\mathcal{Q}}} \hat{f}(\hat{\mathbf{x}})$ for $\hat{f}(\hat{\mathbf{x}}) := f(\mathbf{M}\hat{\mathbf{x}})$ look the same to the algorithm.

In other words, a step of the algorithm in the original optimization problem is the same as a step in the transformed problem. We will further demonstrate in the appendix that the proposed Algorithm 10 which we discuss later in detail is indeed an affine invariant algorithm. In order to obtain an affine invariant algorithm, we define an affine invariant notion of smoothness using the atomic norm. This notion is inspired by the curvature constant employed in FW and MP, see [96, 143]. We define:

$$L_{\mathcal{A}} := \sup_{\substack{\mathbf{x},\mathbf{y}\in\text{lin}(\mathcal{A}) \\ \mathbf{y}=\mathbf{x}+\gamma\mathbf{z} \\ \|\mathbf{z}\|_{\mathcal{A}}=1,\gamma\in\mathbb{R}_{>0}}} \frac{2}{\gamma^2}\left[f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle\right]. \tag{D.4}$$

This definition combines the complexity of the function f as well as the set $\mathcal{A}$ into a single number, and is affine invariant under transformations of our input problem (D.1). It yields the same upper bound to the function as the one given by the traditional smoothness definition, that is $L_{\mathcal{A}}$-smoothness with respect to the atomic norm $\|\cdot\|_{\mathcal{A}}$, when $\mathbf{x},\mathbf{y}$ are constrained to the set $\text{lin}(\mathcal{A})$:

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{L_{\mathcal{A}}}{2}\|\mathbf{y}-\mathbf{x}\|_{\mathcal{A}},$$

For example, if $\mathcal{A}$ is the L1-ball we obtain $f(\mathbf{x}+\gamma\mathbf{z}) \leq f(\mathbf{x}) + \gamma\langle \nabla f(\mathbf{x}), \mathbf{z}\rangle + \gamma^2\frac{L_1}{2}$ where $\|\mathbf{z}\|_1 = 1$. Based on the affine-invariant notion of smoothness defined above, we now present pseudocode of our affine-invariant method in Algorithm 10.

Algorithm 10 Affine Invariant Generalized Matching Pursuit

1: **init** $\mathbf{x}_0 \in \text{lin}(\mathcal{A})$
2: **for** $t = 0 \dots T$
3: Find $\mathbf{z}_t := (\text{Approx-})\text{LMO}_{\mathcal{A}}(\nabla f(\mathbf{x}_t))$
4: $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t\rangle}{L_{\mathcal{A}}}\mathbf{z}_t$
5: **end for**

The above algorithm looks very similar to the generalized MP (Algorithm 9), however, the main difference is that while the original algorithm is not affine invariant over the domain $\mathcal{Q} = \text{lin}(\mathcal{A})$ (Def D.2.1.1), the new Algorithm 10 is so, due to using the generalized smoothness constant $L_{\mathcal{A}}$.

Note. For the purpose of the analysis, we call $\mathbf{x}^\star$ the minimizer of problem (D.1). If the optimum is not unique, we pick the one with largest atomic norm as it represent the worst case for the analysis. All the proofs are deferred to the appendix.

New Affine Invariant Sublinear Rate

In this section, we will provide the theoretical justification of our proposed approach for smooth functions (sublinear rate) and its theoretical comparison with existing previous analysis for special cases. We define the level set radius measured with the atomic norm as:

$$R^2_{\mathcal{A}} := \max_{\substack{\mathbf{x}\in\text{lin}(\mathcal{A}) \\ f(\mathbf{x})\leq f(\mathbf{x}_0)}} \|\mathbf{x}-\mathbf{x}^\star\|^2_{\mathcal{A}}. \tag{D.5}$$

When we measure this radius with the $\|\cdot\|_2$ we call it R_2^2, and when we measure it with $\|\cdot\|_1$ we call it R_1^2. Note that measuring smoothness using the atomic norm guarantees that for the Lipschitz constant $L_{\mathcal{A}}$ the following holds:

Lemma D.2.1.1. *Assume f is L-smooth w.r.t. a given norm $\|\cdot\|$, over $\operatorname{lin}(\mathcal{A})$ where $\mathcal{A}$ is symmetric. Then,*

$$L_{\mathcal{A}} \leq L \operatorname{radius}_{\|\cdot\|}(\mathcal{A})^2 . \tag{D.6}$$

For example, in the coordinate descent setting we measure smoothness with the atomic norm being the L1-norm. Lemma D.2.1.1 implies that $L_{\mathcal{A}} \leq L_1 \leq L_2$ where L_2 is the smoothness constant measured with the L2-norm. Note that the radius of the L1-ball measured with $\|\cdot\|_1$ is 1. Therefore, we put ourselves in a more general setting than Algorithm 9, showing convergence of the affine invariant Algorithm 10

We are now ready to prove the convergence rate of Algorithm 10 for smooth functions.

Theorem D.2.1.2. *Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm over $\operatorname{lin}(\mathcal{A})$. Let f be convex and $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$ over $\operatorname{lin}(\mathcal{A})$, and let $R_{\mathcal{A}}$ be the radius of the level set of $\mathbf{x}_0$ measured with the atomic norm. Then, Algorithm 10 converges for $t \geq 0$ as*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star) \leq \frac{2L_{\mathcal{A}}R_{\mathcal{A}}^2}{\delta^2(t+2)} ,$$

where $\delta \in (0,1]$ is the relative accuracy parameter of the employed approximate LMO (D.3).

Discussion. The proof of Theorem D.2.1.2 extends the convergence analysis of steepest coordinate descent. As opposed to the classical proof in [171], the atoms are here not orthogonal to each other, do not have the same norm and do not correspond to the coordinates of the ambient space. Indeed, $\operatorname{lin}(\mathcal{A})$ could be a subset of the ambient space and the only assumptions on $\mathcal{A}$ are that is closed, bounded and $\|\cdot\|_{\mathcal{A}}$ is a norm over $\operatorname{lin}(\mathcal{A})$. We do not make any incoherence assumption. The key element of our proof is the definition of smoothness using the atomic norm. Furthermore, we use the properties of the atomic norm to obtain a proof which shares the spirit of the Nesterov's one without having to rely on strong assumptions on $\mathcal{A}$.

Relation to Previous MP Sublinear Rate. The sublinear convergence rate presented in Theorem D.2.1.2 is fundamentally different in spirit from the one proved in [143]. Indeed, their convergence analysis builds on top of the proof technique used for Frank-Wolfe in [96]. They introduce a dependency from the atomic norm of the iterates as a way to constrain the part of the space in which the optimization is taking place which artificially induce a notion of duality gap. They do so by defining $\rho := \max\{\|\mathbf{x}^\star\|_{\mathcal{A}}, \|\mathbf{x}_0\|_{\mathcal{A}} \ldots, \|\mathbf{x}_T\|_{\mathcal{A}}\} < \infty$. [143] also used an affine invariant notion of smoothness, thus obtaining an affine invariant rate. On the other hand, their notion of smoothness depends explicitly on ρ.

While this constant can be further upper bounded with the level set radius, it is not known a priori, which makes the estimation of the smoothness constant problematic as it is needed in the algorithm and the proof technique more involved. We propose a much more elegant solution, which uses a different affine invariant definition of smoothness which explicitly depend on the atomic norm. Furthermore, we managed to get rid of the dependency on the sequence of the iterates by using only properties of the atomic norm without any additional assumption (finiteness of ρ).

Relation to Steepest Coordinate Descent. From our analysis, we can readily recover existing rates for coordinate descent. Indeed, if $\mathcal{A}$ is the L1-ball in an n dimensional space, the rate of Theorem D.2.1.2 with exact oracle can be written as:

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star) \leq \frac{2L_1R_1^2}{t+2} \leq \frac{2L_2R_1^2}{t+2} \leq \frac{2L_2nR_2^2}{t+2},$$

where the first inequality is our rate, the second inequality is the rate of [231] and the last inequality is the rate given in [171], both with global Lipschitz constant. Therefore, by measuring smoothness with the atomic norm, we have shown a tighter dependency on the dimensionality of the space. Indeed, the atomic norm gives the tightest norm to measure the product between the smoothness of the function and the level set radius among the known rates. Therefore, our rate for steepest coordinate descent is the tightest known[1].

Coordinate Descent and Affine Transformations. But what does it mean to have an affine invariant rate for coordinate descent? By definition, it means that if one applies an affine transformation to the L1-ball, the coordinate descent algorithm in the natural basis and on the transformed domain $\hat{Q}$ are equivalent. Note that in the transformed problem, the coordinates do not corresponds to the natural coordinates anymore. Indeed, in the transformed domain the coordinates are $\hat{\mathbf{e}}_i = \mathbf{M}^{-1}\mathbf{e}_i$ where $\mathbf{M}^{-1}$ is the inverse of the affine map $\mathbf{M} : \hat{Q} \to Q$. If one would instead perform coordinate descent in the transformed space using the natural coordinates, one would obtain not only different atoms but also a different iterate sequence. In other words, while Matching Pursuit is fully affine invariant, the definition of CD is not, as the choice of the coordinates is not part of the definition of the optimization problem. The two algorithms do coincide for one particular choice of basis, the canonical coordinate basis for $\mathcal{A}$.

Sublinear Rate of Random Pursuit

There is a significant literature on optimization methods which do not require full gradient information. A notable example is random coordinate descent, where only a random

[1] Note that for coordinate-wise L our definition is equivalent to the classical one. $L_{\mathcal{A}} \leq L_2$ if the norm is defined over more than one dimension (i.e. blocks), otherwise there is equality. For the relationship of L_1-smoothness to coordinate-wise smoothness, see also [108, Theorem 4 in Appendix].

component of the gradient is known. As long as the direction that is selected by the LMO is not orthogonal to the gradient we have convergence guarantees due to the inexact oracle definition. We now abstract from the random coordinate descent setting and analyze a randomized variant of matching pursuit, the *random pursuit* algorithm, in which the atom $\mathbf{z}$ is randomly sampled from a distribution over $\mathcal{A}$, rather than picked by a linear minimization oracle. This approach is particularly interesting, as it is deeply connected to the random pursuit algorithm analyzed in [229]. For now we assume that we can compute the projection of the gradient onto a single atom $\langle \nabla f, \mathbf{z} \rangle$ efficiently. In order to present a general recipe for any atom set, we exploit the notion of inexact oracle and define the inexactness of the expectation of the sampled direction for a given sampling distribution:

$$\hat{\delta}^2 := \min_{\mathbf{d} \in \operatorname{lin}(\mathcal{A})} \frac{\mathbb{E}_{\mathbf{z} \in \mathcal{A}} \langle \mathbf{d}, \mathbf{z} \rangle^2}{\|\mathbf{d}\|^2_{\mathcal{A}*}}. \tag{D.7}$$

This constant was already used in [228] to measure the convergence of random pursuit (β^2 in his notation). Note that for uniform sampling from the corners of the L1-ball, we have $\hat{\delta}^2 = \frac{1}{n}$. Indeed, $\mathbb{E}_{\mathbf{z} \in \mathcal{A}} \langle \mathbf{d}, \mathbf{z} \rangle^2 = \frac{1}{n}$ for any $\mathbf{d}$. This definition holds for any sampling scheme as long as $\hat{\delta}^2 \neq 0$. Note that by using this quantity we do not get the tightest possible rate, as at each iteration, we consider how much worse a random update could be compared to the optimal (steepest) update.

We are now ready to present the sublinear convergence rate of random matching pursuit.

Theorem D.2.1.3. *Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm. Let f be convex and $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$ over $\operatorname{lin}(\mathcal{A})$ and let $R_{\mathcal{A}}$ be the radius of the level set of $\mathbf{x}_0$ measured with the atomic norm. Then, Algorithm 10 converges for $t \geq 0$ as*

$$\mathbb{E}_{\mathbf{z}}\big[f(\mathbf{x}_{t+1})\big] - f(\mathbf{x}^\star) \leq \frac{2L_{\mathcal{A}} R^2_{\mathcal{A}}}{\hat{\delta}^2 (t+2)},$$

when the LMO *is replaced with random sampling of $\mathbf{z}$ from a distribution over $\mathcal{A}$.*

Gradient-Free Variant. If is possible to obtain a fully gradient-free optimization scheme. In addition to having replaced the LMO in Algorithm 9 by the random sampling as above, as can additionally also replace the line search step on the quadratic upper bound given by smoothness, with instead an approximate line search on f. As long as the update scheme guarantees as much decrease as the above algorithm, the convergence rate of Theorem D.2.1.3 holds.

Discussion. This approach is very general, as it allows to guarantee convergence for *any* sampling scheme and *any* set $\mathcal{A}$ provided that $\hat{\delta}^2 \neq 0$. In the coordinate descent case we have that for the worst possible gradient for random has $\hat{\delta}^2 = \frac{1}{n}$. Therefore, the speed-up of steepest can be up to a factor equal to the number of dimensions in the best case. Similarly, if $\mathbf{z}$ is sampled from a spherical distribution, $\hat{\delta}^2 = \frac{1}{n}$ [229]. More

examples of computation of $\hat{\delta}^2$ can be found in [228, Section 4.2]. Last but not least, note that $\hat{\delta}^2$ is affine invariant as long as the sampling distribution over the atoms is preserved.

Strong Convexity and Affine Invariant Linear Rates

Similar to the affine invariant notion of smoothness, we here define the affine invariant notion of strong convexity.

$$\mu_{\mathcal{A}} := \inf_{\substack{\mathbf{x},\mathbf{y}\in\operatorname{lin}(\mathcal{A}) \\ \mathbf{x}\neq\mathbf{y}}} \frac{2}{\|\mathbf{y}-\mathbf{x}\|_{\mathcal{A}}^2} D(\mathbf{y},\mathbf{x})\,.$$

where $D(\mathbf{y},\mathbf{x}) := f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle$ We can now show the linear convergence rate of both the matching pursuit algorithm and its random pursuit variant.

Theorem D.2.1.4. *Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm. Let f be $\mu_{\mathcal{A}}$-strongly convex and $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$, both over* $\operatorname{lin}(\mathcal{A})$. *Then, Algorithm 10 converges for $t \geq 0$ as*

$$\varepsilon_{t+1} \leq \big(1 - \delta^2 \frac{\mu_{\mathcal{A}}}{L_{\mathcal{A}}}\big)\varepsilon_t\,.$$

where $\varepsilon_t := f(\mathbf{x}_t) - f(\mathbf{x}^\star)$. If the LMO direction is sampled randomly from $\mathcal{A}$, Algorithm 10 converges for $t \geq 0$ as

$$\mathbb{E}_{\mathbf{z}}\left[\varepsilon_{t+1} | \mathbf{x}_t\right] \leq \big(1 - \hat{\delta}^2 \frac{\mu_{\mathcal{A}}}{L_{\mathcal{A}}}\big)\varepsilon_t\,.$$

Relation to Previous MP Linear Rate. Again, the proof of Theorem D.2.1.4 extends the convergence analysis of steepest coordinate descent using solely the affine invariant definition of strong convexity and the properties of the atomic norm. Note that again we define the strong convexity constant without relying on $\rho = \max\{\|\mathbf{x}^\star\|_{\mathcal{A}}, \|\mathbf{x}_0\|_{\mathcal{A}} \ldots, \|\mathbf{x}_T\|_{\mathcal{A}}\} < \infty$ as in [143]. We now show that our choice of the strong convexity parameter is the tightest w.r.t. any choice of the norm and that we can precisely recover the non affine invariant rate of [143]. Let us recall their notion of *minimal directional width*, which is the crucial constant to measure the geometry of the atom set for a fixed norm:

$$\operatorname{mDW}(\mathcal{A}) := \min_{\substack{\mathbf{d}\in\operatorname{lin}(\mathcal{A}) \\ \mathbf{d}\neq 0}} \max_{z\in\mathcal{A}} \left\langle \frac{\mathbf{d}}{\|\mathbf{d}\|}, \mathbf{z} \right\rangle.$$

Note that for CD we have that $\operatorname{mDW}(\mathcal{A}) = \frac{1}{\sqrt{n}}$. Now, we relate the affine invariant notion of strong convexity with the minimal directional width and the strong convexity w.r.t. any chosen norm. This is important, as we want to make sure to perfectly recover the convergence rate given in [143].

Lemma D.2.1.5. *Assume f is μ-strongly convex w.r.t. a given norm $\|\cdot\|$ over $\operatorname{lin}(\mathcal{A})$ and $\mathcal{A}$ is symmetric. Then:*

$$\mu_{\mathcal{A}} \geq \operatorname{mDW}(\mathcal{A})^2 \mu \,.$$

We then recover their non-affine-invariant rate as:

$$\varepsilon_{t+1} \leq \left(1 - \delta^2 \frac{\mu \operatorname{mDW}(\mathcal{A})^2}{L \operatorname{radius}_{\|\cdot\|}(\mathcal{A})^2}\right) \varepsilon_t \,.$$

Relation to Coordinate Descent. When we fix $\mathcal{A}$ as the L1-ball and use an exact oracle our rate becomes:

$$\varepsilon_{t+1} \leq \left(1 - \frac{\mu_1}{L_1}\right) \varepsilon_t \leq \left(1 - \frac{\mu_1}{L}\right) \varepsilon_t \leq \left(1 - \frac{\mu}{nL}\right) \varepsilon_t \,,$$

where the first is our rate, the second is the rate of steepest CD [181] and the last is the one for randomized CD [171] (n is the dimension of the ambient space). Therefore, our linear rate for coordinate descent is the tightest known.

D.3 Accelerating Generalized Matching Pursuit

As we established in the previous sections, matching pursuit can be considered a generalized greedy coordinate descent where the allowed directions do not need to form an orthogonal basis. This insight allows us to generalize the analysis of accelerated coordinate descent methods and to accelerate matching pursuit [131, 176]. However it is not clear at the outset how to even accelerate greedy coordinate descent, let alone the matching pursuit method. Recently Song *et al.* [224] proposed an accelerated greedy coordinate descent method by using the linear coupling framework of [8]. However the updates they perform at each iteration are not guaranteed to be sparse which is critical for our application. We instead extend the acceleration technique in [229] which in turn is based on [131]. They allow the updates to the two sequences of iterates $\mathbf{x}$ and $\boldsymbol{b}$ to be chosen from any distribution. If this distribution is chosen to be over coordinate directions, we get the familiar accelerated coordinate descent, and if we instead chose the distribution to be over the set of atoms, we would get an accelerated random pursuit algorithm. To obtain an accelerated *matching* pursuit algorithm, we need to additionally *decouple* the updates for $\mathbf{x}$ and $\boldsymbol{b}$ and allow them to be chosen from different distributions. We will update $\mathbf{x}$ using the greedy coordinate update (or the matching pursuit update), and use a random coordinate (or atom) direction to update $\boldsymbol{b}$.

The possibility of decoupling the updates was noted in [228, Corollary 6.4] though its implications for accelerating greedy coordinate descent or matching pursuit were not explored. From here on out, we shall assume that the linear space spanned by the atoms $\mathcal{A}$ is finite dimensional. This was not necessary for the non-accelerated matching pursuit

and it remains open if it is necessary for accelerated MP. When sampling, we consider only a non-symmetric version of the set $\mathcal{A}$ with all the atoms in the same half space. Line search ensures that sampling either $\mathbf{z}$ or $-\mathbf{z}$ yields the same update. For simplicity, we focus on an exact LMO.

D.3.1 From Coordinates to Atoms

For the acceleration of MP we make some stronger assumption w.r.t. the rates in the previous section. In particular, we will not obtain an affine invariant rate which remains an open problem. The key challenges for an affine invariant accelerated rate are strong convexity of the model, which can be solved using arguments similar to [51] and the fact that our proof relies on defining a new norm which deform the space in order to obtain favorable sampling properties as we will explain in this section. The main difference between working with atoms and working with coordinates is that projection along coordinate basis vectors is 'unbiased'. Let $\mathbf{e}_i$ represent the ith coordinate basis vector. Then for some vector $\mathbf{d}$, if we project along a random basis vector $\mathbf{e}_i$,

$$\mathbb{E}_{i\in[n]}[\langle \mathbf{e}_i, \mathbf{d}\rangle \mathbf{e}_i] = \frac{1}{n}\mathbf{d}\,.$$

However if instead of coordinate basis, we choose from a set of atoms $\mathcal{A}$, then this is no longer true. We can correct for this by morphing the geometry of the space. Suppose we sample the atoms from a distribution $\mathcal{Z}$ defined over $\mathcal{A}$. Let us define

$$\tilde{\mathbf{P}} := \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}}[\mathbf{z}\mathbf{z}^\top]\,.$$

We assume that the distribution $\mathcal{Z}$ is such that $\operatorname{lin}(\mathcal{A}) \subseteq \operatorname{range}(\tilde{\mathbf{P}})$. This intuitively corresponds to assuming that there is a non-zero probability that the sampled $\mathbf{z} \sim \mathcal{Z}$ is along the direction of every atom $\mathbf{z}_t \in \mathcal{A}$ i.e.

$$\mathbb{P}_{\mathbf{z}\sim\mathcal{Z}}[\langle \mathbf{z}, \mathbf{z}_t\rangle > 0] > 0,\ \forall \mathbf{z}_t \in \mathcal{A}\,.$$

Further let $\mathbf{P} = \tilde{\mathbf{P}}^\dagger$ be the pseudo-inverse of $\tilde{\mathbf{P}}$. Note that both $\mathbf{P}$ and $\tilde{\mathbf{P}}$ are positive semi-definite matrices. We can equip our space with a new inner product $\langle \cdot, \mathbf{P}\cdot\rangle$ and the resulting norm $\|\cdot\|_{\mathbf{P}}$. With this new dot product,

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{Z}}[\langle \mathbf{z}, \mathbf{P}\mathbf{d}\rangle \mathbf{z}] = \mathbb{E}_{\mathbf{z}\sim\mathcal{Z}}[\mathbf{z}\mathbf{z}^\top]\mathbf{P}\mathbf{d} = \mathbf{P}^\dagger\mathbf{P}\mathbf{d} = \mathbf{d}\,.$$

The last equality follows from our assumption that $\operatorname{lin}(\mathcal{A}) \subseteq \operatorname{range}(\tilde{\mathbf{P}})$.

D.3.2 Analysis

Modeling explicitly the dependency on the structure of the set is crucial to accelerate MP. Indeed, acceleration works by defining two different quadratic subproblems, one upper

Algorithm 11 Accelerated Random Pursuit

init $\mathbf{x}_0 = \boldsymbol{b}_0 = \mathbf{y}_0$, $\beta_0 = 0$, and v'
for $t = 0, 1 \dots T$
  Solve $\alpha_{t+1}^2 L v' = \beta_t + \alpha_{t+1}$
  $\beta_{t+1} := \beta_t + \alpha_{t+1}$
  $\tau_t := \frac{\alpha_{t+1}}{\beta_{t+1}}$
  Compute $\mathbf{y}_t := (1-\tau_t)\mathbf{x}_t + \tau_t \boldsymbol{b}_t$
  Sample $\mathbf{z}_t \sim \mathcal{Z}$
  $\mathbf{x}_{t+1} := \mathbf{y}_t - \frac{\langle \nabla f(\mathbf{y}_t), \mathbf{z}_t \rangle}{L \|\mathbf{z}_t\|_2^2} \mathbf{z}_t$
  $\boldsymbol{b}_{t+1} := \boldsymbol{b}_t - \alpha_{t+1} \langle \nabla f(\mathbf{y}_t), \mathbf{z}_t \rangle \mathbf{z}_t$
end for

Algorithm 12 Accelerated Matching Pursuit

init $\mathbf{x}_0 = \boldsymbol{b}_0 = \mathbf{y}_0$, $\beta_0 = 0$, and v
for $t = 0, 1 \dots T$
  Solve $\alpha_{t+1}^2 L v = \beta_t + \alpha_{t+1}$
  $\beta_{t+1} := \beta_t + \alpha_{t+1}$
  $\tau_t := \frac{\alpha_{t+1}}{\beta_{t+1}}$
  Compute $\mathbf{y}_t := (1-\tau_t)\mathbf{x}_t + \tau_t \boldsymbol{b}_t$
  Find $\mathbf{z}_t := \mathrm{LMO}_{\mathcal{A}}(\nabla f(\mathbf{y}_t))$
  $\mathbf{x}_{t+1} := \mathbf{y}_t - \frac{\langle \nabla f(\mathbf{y}_t), \mathbf{z}_t \rangle}{L \|\mathbf{z}_t\|_2^2} \mathbf{z}_t$
  Sample $\tilde{\mathbf{z}}_t \sim \mathcal{Z}$
  $\boldsymbol{b}_{t+1} := \boldsymbol{b}_t - \alpha_{t+1} \langle \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t \rangle \tilde{\mathbf{z}}_t$
end for

bound given by smoothness, and one lower bound given by a model of the function. The constraints on the set of possible descent direction implicitly used in MP influence both these subproblems. While the smoothness quadratic upper bound contains information about $\mathcal{A}$ in its definition ($\mathbf{y} = \mathbf{x} + \gamma \mathbf{z}$ and $\|\mathbf{z}\|_{\mathcal{A}} = 1$), the model of the function needs explicit modeling of $\mathcal{A}$. This is particularly crucial when sampling a direction in the model update, which can be thought as a sort of exploration part of the algorithm. In both the algorithms, the update of the parameter $\boldsymbol{b}$ corresponds to optimizing the modeling function ψ which can be given as :

$$\psi_{t+1}(\mathbf{x}) = \psi_t(\mathbf{x}) + \alpha_{t+1} \left(f(\mathbf{y}_t) + \langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P} (\mathbf{x} - \mathbf{y}_t) \rangle \right), \tag{D.8}$$

where $\psi_0(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} - \mathbf{x}_0\|_{\mathbf{P}}^2$.

Lemma D.3.2.1. *The update of $\boldsymbol{b}$ in Algorithm 11 and 12 minimizes the model*

$$\boldsymbol{b}_t \in \arg\min_{\mathbf{x}} \psi_t(\mathbf{x}) .$$

We will be first discussing the theory for the *greedy* accelerated method in detail. As evident from the algorithm 12, another important constant which is required for both the analysis and to actually run the algorithm is ν for which:

$$\nu \leq \max_{\mathbf{d} \in \operatorname{lin}(\mathcal{A})} \frac{\mathbb{E}\left[(\tilde{\mathbf{z}}_t^\top \mathbf{d})^2 \|\tilde{\mathbf{z}}_t\|_{\mathbf{P}}^2\right] \|\mathbf{z}(\mathbf{d})\|_2^2}{(\mathbf{z}(\mathbf{d})^\top \mathbf{d})^2} ,$$

where $\mathbf{z}(\mathbf{d})$ is defined to be

$$\mathbf{z}(\mathbf{d}) = \mathrm{LMO}_{\mathcal{A}}(-\mathbf{d}) .$$

The quantity ν relates the geometry of the atom set with the sampling procedure in a similar way as $\hat{\delta}^2$ in Equation (D.7) but instead of measuring how much worse a random update is when compared to a steepest update.

Theorem D.3.2.2. *Let f be a convex function and $\mathcal{A}$ be a symmetric compact set. Then the output of algorithm 12 for any $t \geq 1$ converges with the following rate:*

$$\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}^\star) \leq \frac{2L\nu}{t(t+1)} \|\mathbf{x}^\star - \mathbf{x}_0\|_{\mathbf{P}}^2 .$$

Proof. We extend the proof technique of [131, 229] to allow for general atomic updates. The analysis can be found in Appendix D.8.1 □

Once we understand the convergence of the greedy approach, the analysis of accelerated random pursuit can be derived easily. Here, we state the rate of convergence for accelerated random pursuit:

Theorem D.3.2.3. *Let f be a convex function and $\mathcal{A}$ be a symmetric set. Then the output of the algorithm 11 for any $t \geq 1$ converges with the following rate:*

$$\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}^\star) \leq \frac{2L\nu'}{t(t+1)} \|\mathbf{x}^\star - \mathbf{x}_0\|_{\mathbf{P}}^2 ,$$

where

$$\nu' \leq \max_{\mathbf{d} \in \operatorname{lin}(\mathcal{A})} \frac{\mathbb{E}\left[(\mathbf{z}_t^\top \mathbf{d})^2 \|\mathbf{z}_t\|_{\mathbf{P}}^2\right]}{\mathbb{E}\left[(\mathbf{z}_t^\top \mathbf{d})^2 / \|\mathbf{z}_t\|_2^2\right]} .$$

Discussion on Greedy Accelerated Coordinate Descent. The convergence rate for greedy accelerated coordinate descent can directly be obtained from the rate from accelerated matching pursuit. Let the atom set $\mathcal{A}$ consist of the standard basis vectors $\{\mathbf{e}_i, i \in [n]\}$ and $\mathcal{Z}$ be a uniform distribution over this set. Then algorithm 11 reduces to the accelerated randomized coordinate method (ACDM) of [131, 176] and we recover their rates. Instead

if we use algorithm 12, we obtain a novel accelerated greedy coordinate method with a (potentially) better convergence rate.[2]

Lemma D.3.2.4. *When $\mathcal{A} = \{\mathbf{e}_i, i \in [n]\}$ and $\mathcal{Z}$ is a uniform distribution over $\mathcal{A}$, then $\mathbf{P} = n\mathbf{l}$, $\nu' = n$ and $\nu \in [1, n]$.*

D.4 Empirical Evaluation

In this section we aim at empirically validate our theoretical findings. In both experiments we use 1 and the intrinsic dimensionality of $\text{lin}(\mathcal{A})$ as ν and ν' respectively. Note that a value of ν smaller than ν' represents the best case for the steepest update. We implicitly assume that the worst case in which a random update is as good as the steepest one never happens.

Toy Data: First, we report the function value while minimizing the squared distance between the a random 100 dimensional signal with both positive and negative entries and its sparse representation in terms of atoms. We sample a random dictionary containing 200 atoms which we then make symmetric. The result is depicted in Figure D.1. As anticipated from our analysis, the accelerated schemes converge much faster than the non-accelerated variants. Furthermore, in both cases the steepest update converge faster than the random one, due to a better dependency on the dimensionality of the space.

Real Data: We use the under-sampled Urban HDI Dataset from which we extract the dictionary of atoms using the hierarchical clustering approached of [79]. This dataset contains 5'929 pixels, each associated with 162 hyperspectral features. The number of dictionary elements is 6, motivated by the fact that 6 different physical materials are depicted in this HSI data [78]. We approximate each pixel with a linear combination of the dictionary elements by minimizing the square distance between the observed pixel and our approximation. We report in Figure D.2 the loss as an average across all the pixels:

$$\min_{\mathbf{x}_i \in \text{lin}(\mathcal{A})} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{b}_i\|^2$$

We notice that as expected, the steepest matching pursuit converges faster than the random pursuit, but as expected both of them converge at the same regime. On the other hand, the accelerated scheme converge much faster than the non-accelerated variants. Note that the acceleration kicks in only after a few iterations as the accelerated rate has a worse dependency on the intrinsic dimensionality of the linear span than the non accelerated algorithms. We notice that the speedup of steepest MP is much more evident in the synthetic data. The reason is that this experiment is much more high dimensional than the hyperspectral data. Indeed, the span of the dictionary is a 6 dimensional manifold in the latter and the full ambient space in the former and the steepest update yields a better dependency on the dimensionality.

[2] Simultaneously (and independently) [145] derived the same accelerated greedy coordinate algorithm.

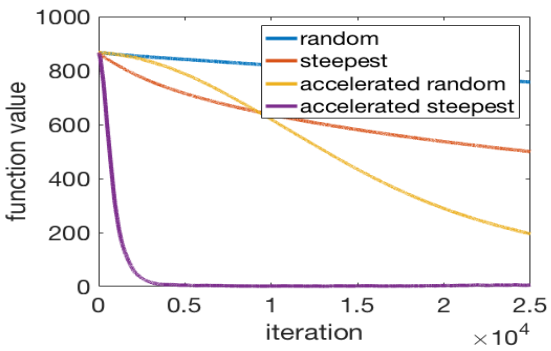


Figure D.1: loss for synthetic data

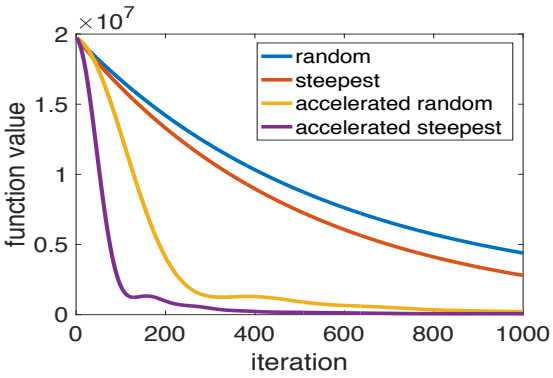


Figure D.2: loss for hyperspectral data

D.5 Conclusions

In this paper we presented a unified analysis of matching pursuit and coordinate descent algorithms. As a consequence, we exploit the similarity between the two to obtain the best of both worlds: tight sublinear and linear rates for steepest coordinate descent and the first accelerated rate for matching pursuit and steepest coordinate descent. Furthermore, we discussed the relation between the steepest and the random directions by viewing the latter as an approximate version of the former. An affine invariant accelerated proof remains an open problem.

Proofs for Main Results

D.6 Sublinear Rates

Theorem' D.2.1.1. *Assume f is L-smooth w.r.t. a given norm $\|\cdot\|$, over $\langle(\rangle \mathcal{A})$ where $\mathcal{A}$ is symmetric. Then,*

$$L_{\mathcal{A}} \leq L \operatorname{radius}_{\|\cdot\|}(\mathcal{A})^2 . \tag{D.9}$$

Proof. Let $D(\mathbf{y},\mathbf{x}) := f(\mathbf{y}) - f(\mathbf{x}) + \gamma\langle \nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle$ By the definition of smoothness of f w.r.t. $\|\cdot\|$,

$$D(\mathbf{y},\mathbf{x}) \leq \frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2 .$$

Hence, from the definition of $L_{\mathcal{A}}$,

$$\begin{aligned} L_{\mathcal{A}} &\leq \sup_{\substack{\mathbf{x},\mathbf{y}\in\langle(\rangle\mathcal{A}) \\ \mathbf{y}=\mathbf{x}+\gamma\mathbf{z} \\ \|\mathbf{z}\|_{\mathcal{A}}=1,\gamma\in\mathbb{R}_{>0}}} \frac{2}{\gamma^2}\frac{L}{2}\|\mathbf{y}-\mathbf{x}\|^2 \\ &= L \sup_{\mathbf{z}\ s.t.\|\mathbf{z}\|_{\mathcal{A}}=1} \|\mathbf{s}\|^2 \\ &= L \operatorname{radius}_{\|\cdot\|}(\mathcal{A})^2 . \end{aligned}$$

□

The definition of the smoothness constant w.r.t. the atomic norm yields the following quadratic upper bound:

$$L_{\mathcal{A}} = \sup_{\substack{\mathbf{x},\mathbf{y}\in\langle(\rangle\mathcal{A}) \\ \mathbf{y}=\mathbf{x}+\gamma\mathbf{z} \\ \|\mathbf{z}\|_{\mathcal{A}}=1,\gamma\in\mathbb{R}_{>0}}} \frac{2}{\gamma^2}\left[f(\mathbf{y}) - f(\mathbf{x}) + \langle\nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle\right] . \tag{D.10}$$

Furthermore, let:

$$R_{\mathcal{A}}^2 = \max_{\substack{\mathbf{x}\in\langle(\rangle\mathcal{A}) \\ f(\mathbf{x})\leq f(\mathbf{x}_0)}} \|\mathbf{x}-\mathbf{x}^\star\|_{\mathcal{A}}^2 . \tag{D.11}$$

Now, we show that the algorithm we presented is affine invariant. An optimization method is called *affine invariant* if it is invariant under affine transformations of the input problem: If one chooses any re-parameterization of the domain $\mathcal{C}$ by a *surjective* linear or affine map $\mathbf{M} : \hat{\mathcal{C}} \to \mathcal{C}$, then the "old" and "new" optimization problems $\min_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x})$ and $\min_{\hat{\mathbf{x}}\in\hat{\mathcal{C}}} \hat{f}(\hat{\mathbf{x}})$ for $\hat{f}(\hat{\mathbf{x}}) := f(\mathbf{M}\hat{\mathbf{x}})$ look the same to the algorithm. Note that $\nabla\hat{f} = \mathbf{M}^T\nabla f$.

First of all, let us note that $L_{\mathcal{A}}$ is affine invariant as it does not depend on any norm. Now:

$$\begin{aligned}
\mathbf{M}\hat{\mathbf{x}}_{t+1} &= \mathbf{M}\left(\hat{\mathbf{x}}_t + \frac{\langle \nabla \hat{f}(\hat{\mathbf{x}}_t), \hat{\mathbf{z}}_t \rangle}{L_{\mathcal{A}}} \hat{\mathbf{z}}_t\right) \\
&= \mathbf{M}\hat{\mathbf{x}}_t + \frac{\langle \nabla \hat{f}(\hat{\mathbf{x}}_t), \hat{\mathbf{z}}_t \rangle}{L_{\mathcal{A}}} \mathbf{M}\hat{\mathbf{z}}_t \\
&= \mathbf{x}_t + \frac{\langle \nabla \hat{f}(\hat{\mathbf{x}}_t), \hat{\mathbf{z}}_t \rangle}{L_{\mathcal{A}}} \mathbf{z}_t \\
&= \mathbf{x}_t + \frac{\langle \mathbf{M}^T \nabla f(\mathbf{x}_t), \hat{\mathbf{z}}_t \rangle}{L_{\mathcal{A}}} \mathbf{z}_t \\
&= \mathbf{x}_t + \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{M}\hat{\mathbf{z}}_t \rangle}{L_{\mathcal{A}}} \mathbf{z}_t \\
&= \mathbf{x}_t + \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{z}_t \rangle}{L_{\mathcal{A}}} \mathbf{z}_t \\
&= \mathbf{x}_{t+1} \,.
\end{aligned}$$

Therefore the algorithm is affine invariant.

D.6.1 Affine Invariant Sublinear Rate

Theorem' D.2.1.2. *Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm over $\langle() \mathcal{A})$. Let f be convex and $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$ over $\langle() \mathcal{A})$, and let $R_{\mathcal{A}}$ be the radius of the level set of $\mathbf{x}_0$ measured with the atomic norm. Then, Algorithm 10 converges for $t \geq 0$ as*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star) \leq \frac{2L_{\mathcal{A}}R_{\mathcal{A}}^2}{\delta^2(t+2)} \,,$$

where $\delta \in (0,1]$ is the relative accuracy parameter of the employed approximate LMO (D.3).

Proof. Recall that $\tilde{\mathbf{z}}_t$ is the atom selected in iteration t by the approximate LMO defined in (D.3). We start by upper-bounding f using the definition of $L_{\mathcal{A}}$ as follows:

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq \min_{\gamma \in \mathbb{R}} f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle + \frac{\gamma^2}{2} L_{\mathcal{A}} \|\mathbf{z}\|_{\mathcal{A}}^2 \\
&= \min_{\gamma \in \mathbb{R}} f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle + \frac{\gamma^2}{2} L_{\mathcal{A}} \\
&\leq f(\mathbf{x}_t) - \frac{\langle \nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle^2}{2L_{\mathcal{A}}}
\end{aligned}$$

$$
\begin{aligned}
&= f(\mathbf{x}_t) - \frac{\langle \nabla_{\|} f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle^2}{2L_{\mathcal{A}}} \\
&\leq f(\mathbf{x}_t) - \delta^2 \frac{\langle \nabla_{\|} f(\mathbf{x}_t), \mathbf{z}_t \rangle^2}{2L_{\mathcal{A}}} .
\end{aligned}
$$

Where $\nabla_{\|} f$ is the parallel component of the gradient wrt the linear span of $\mathcal{A}$. Note that $\|\mathbf{d}\|_{\mathcal{A}*} := \sup\{\langle \mathbf{z}, \mathbf{d} \rangle, \mathbf{z} \in \mathcal{A}\}$ is the dual of the atomic norm. Therefore, by definition:

$$
\langle \nabla_{\|} f(\mathbf{x}_t), \mathbf{z}_t \rangle^2 = \| - \nabla_{\|} f(\mathbf{x}_t) \|^2_{\mathcal{A}*} ,
$$

which gives:

$$
\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) - \delta^2 \frac{1}{2L_{\mathcal{A}}} \|\nabla_{\|} f(\mathbf{x}_t)\|^2_{\mathcal{A}*} \\
&\leq f(\mathbf{x}_t) - \delta^2 \frac{1}{2L_{\mathcal{A}}} \frac{\left(-\langle \nabla_{\|} f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle\right)^2}{R^2_{\mathcal{A}}} \\
&= f(\mathbf{x}_t) - \delta^2 \frac{1}{2L_{\mathcal{A}}} \frac{\left(\langle \nabla_{\|} f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle\right)^2}{R^2_{\mathcal{A}}} \\
&\leq f(\mathbf{x}_t) - \delta^2 \frac{1}{2L_{\mathcal{A}}} \frac{\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right)^2}{R^2_{\mathcal{A}}} ,
\end{aligned}
$$

where the second inequality is Cauchy-Schwarz and the third one is convexity. Which gives:

$$
\varepsilon_{t+1} \leq \frac{2L_{\mathcal{A}} R^2_{\mathcal{A}}}{\delta^2 (t+2)} . \qquad \square
$$

D.6.2 Randomized Affine Invariant Sublinear Rate

For random sampling of $\mathbf{z}$ from a distribution over $\mathcal{A}$, let

$$
\hat{\delta}^2 := \min_{\mathbf{d} \in \langle \mathcal{A} \rangle} \frac{\mathbb{E}_{\mathbf{z} \in \mathcal{A}} \langle \mathbf{d}, \mathbf{z} \rangle^2}{\|\mathbf{d}\|^2_{\mathcal{A}*}} . \tag{D.12}
$$

Theorem' D.2.1.3. *Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm. Let f be convex and $L_{\mathcal{A}}$ smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$ over $\langle() \mathcal{A})$ and let $R_{\mathcal{A}}$ be the radius of the level set of $\mathbf{x}_0$ measured with the atomic norm. Then, Algorithm 10 converges for $t \geq 0$ as*

$$
\mathbb{E}_{\mathbf{z}}\left[f(\mathbf{x}_{t+1})\right] - f(\mathbf{x}^\star) \leq \frac{2L_{\mathcal{A}} R^2_{\mathcal{A}}}{\hat{\delta}^2 (t+2)} ,
$$

when the LMO *is replaced with random sampling of* $\mathbf{z}$ *from a distribution over* $\mathcal{A}$.

Proof. Recall that $\tilde{\mathbf{z}}_t$ is the atom selected in iteration t by the approximate LMO defined in (D.3). We start by upper-bounding f using the definition of $L_{\mathcal{A}}$ as follows

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}} f(\mathbf{x}_{t+1}) &\leq \mathbb{E}_{\mathbf{z}}\left[\min_{\gamma\in\mathbb{R}} f(\mathbf{x}_t) + \gamma\langle\nabla f(\mathbf{x}_t),\mathbf{z}\rangle + \frac{\gamma^2}{2}L_{\mathcal{A}}\|\mathbf{z}\|_{\mathcal{A}}^2\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\min_{\gamma\in\mathbb{R}} f(\mathbf{x}_t) + \gamma\langle\nabla f(\mathbf{x}_t),\mathbf{z}\rangle + \frac{\gamma^2}{2}L_{\mathcal{A}}\right] \\
&\leq f(\mathbf{x}_t) - \frac{\mathbb{E}_{\mathbf{z}}\left[\langle\nabla f(\mathbf{x}_t),\mathbf{z}\rangle^2\right]}{2L_{\mathcal{A}}} \\
&= f(\mathbf{x}_t) - \frac{\mathbb{E}_{\mathbf{z}}\left[\langle\nabla_{\|} f(\mathbf{x}_t),\mathbf{z}\rangle^2\right]}{2L_{\mathcal{A}}} \\
&\leq f(\mathbf{x}_t) - \hat{\delta}^2\frac{\langle\nabla_{\|} f(\mathbf{x}_t),\mathbf{z}_t\rangle^2}{2L_{\mathcal{A}}}\,.
\end{aligned}$$

The rest of the proof proceeds as in Theorem D.2.1.2. □

D.7 Linear Rates

D.7.1 Affine Invariant Linear Rate

Let us first the fine the affine invariant notion of strong convexity based on the atomic norm:

$$\mu_{\mathcal{A}} := \inf_{\substack{\mathbf{x},\mathbf{y}\in\langle\mathcal{A}\rangle \\ \mathbf{x}\neq\mathbf{y}}} \frac{2}{\|\mathbf{y}-\mathbf{x}\|_{\mathcal{A}}^2}\left[f(\mathbf{y}) - f(\mathbf{x}) - \langle\nabla f(\mathbf{x}),\mathbf{y}-\mathbf{x}\rangle\right].$$

Let us recall the definition of *minimal directional width* from [143]:

$$\mathrm{mDW}(\mathcal{A}) := \min_{\substack{\mathbf{d}\in\langle()\mathcal{A}) \\ \mathbf{d}\neq 0}} \max_{z\in\mathcal{A}} \langle\frac{\mathbf{d}}{\|\mathbf{d}\|},\mathbf{z}\rangle\,.$$

Then, we can relate our new definition of strong convexity with the $\mathrm{mDW}(\mathcal{A})$ as follows.

Theorem' D.2.1.5. *Assume* f *is* μ *strongly convex wrt a given norm* $\|\cdot\|$ *over* $\langle()\,\mathcal{A})$ *and* $\mathcal{A}$ *is symmetric. Then:*

$$\mu_{\mathcal{A}} \geq \mathrm{mDW}(\mathcal{A})^2\,\mu\,.$$

Proof. First of all, note that for any $\mathbf{x}, \mathbf{y} \in \langle () \, \mathcal{A}) \rangle$ with $\mathbf{x} \neq \mathbf{y}$ we have that:

$$\langle \nabla f(x), x - y \rangle^2 \leq \|\nabla f(\mathbf{x})\|_{\mathcal{A}*}^2 \|\mathbf{x} - \mathbf{y}\|_{\mathcal{A}}^2 .$$

Therefore:

$$\begin{aligned}
\mu_{\mathcal{A}} &= \inf_{\substack{\mathbf{x},\mathbf{y} \in \langle \mathcal{A} \rangle \\ \mathbf{x} \neq \mathbf{y}}} \frac{2}{\|\mathbf{y} - \mathbf{x}\|_{\mathcal{A}}^2} D(\mathbf{x}, \mathbf{y}) \\
&\geq \inf_{\substack{\mathbf{x},\mathbf{y},\mathbf{d} \in \langle \mathcal{A} \rangle \\ \mathbf{x} \neq \mathbf{y}, \mathbf{d} \neq 0}} \frac{\|\mathbf{d}\|_{\mathcal{A}*}^2}{\langle \mathbf{d}, \mathbf{x} - \mathbf{y} \rangle^2} 2D(\mathbf{x}, \mathbf{y}) \\
&\geq \inf_{\substack{\mathbf{x},\mathbf{y},\mathbf{d} \in \langle \mathcal{A} \rangle \\ \mathbf{x} \neq \mathbf{y}, \mathbf{d} \neq 0}} \frac{\|\mathbf{d}\|_{\mathcal{A}*}^2}{\langle \mathbf{d}, \mathbf{x} - \mathbf{y} \rangle^2} \mu \|\mathbf{x} - \mathbf{y}\|^2 \\
&\geq \inf_{\substack{\mathbf{x},\mathbf{y},\mathbf{d} \in \langle \mathcal{A} \rangle \\ \mathbf{x} \neq \mathbf{y}, \mathbf{d} \neq 0}} \frac{\|\mathbf{d}\|_{\mathcal{A}*}^2}{\langle \mathbf{d}, \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|} \rangle^2} \mu \\
&\geq \inf_{\substack{\mathbf{x},\mathbf{y},\mathbf{d} \in \langle \mathcal{A} \rangle \\ \mathbf{x} \neq \mathbf{y}, \mathbf{d} \neq 0}} \frac{\|\mathbf{d}\|_{\mathcal{A}*}^2}{\|\mathbf{d}\|^2} \mu \\
&\geq \inf_{\substack{\mathbf{d} \in \langle \mathcal{A} \rangle \\ \mathbf{d} \neq 0}} \max_{z} \frac{\langle \mathbf{d}, \mathbf{z} \rangle^2}{\|\mathbf{d}\|^2} \mu \\
&= \mathrm{mDW}(\mathcal{A})^2 \, \mu \, .
\end{aligned}$$

□

Theorem' D.2.1.4. *(Part 1). Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm. Let f be $\mu_{\mathcal{A}}$-strongly convex and $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$, both over $\langle () \, \mathcal{A}) \rangle$. Then, Algorithm 10 converges for $t \geq 0$ as*

$$\varepsilon_{t+1} \leq \left(1 - \delta^2 \frac{\mu_{\mathcal{A}}}{L_{\mathcal{A}}}\right) \varepsilon_t \, .$$

where $\varepsilon_t := f(\mathbf{x}_t) - f(\mathbf{x}^\star)$.

Proof. Recall that $\tilde{\mathbf{z}}_t$ is the atom selected in iteration t by the approximate LMO defined in (D.3). We start by upper-bounding f using the definition of $L_{\mathcal{A}}$ as follows

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq \min_{\gamma \in \mathbb{R}} f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle + \frac{\gamma^2}{2} L_{\mathcal{A}} \|\mathbf{z}\|_{\mathcal{A}}^2 \\
&= \min_{\gamma \in \mathbb{R}} f(\mathbf{x}_t) + \gamma \langle \nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle + \frac{\gamma^2}{2} L_{\mathcal{A}}
\end{aligned}$$

$$
\begin{aligned}
&\leq f(\mathbf{x}_t) - \frac{\langle \nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle^2}{2L_{\mathcal{A}}} \\
&= f(\mathbf{x}_t) - \frac{\langle \nabla_{\|} f(\mathbf{x}_t), \tilde{\mathbf{z}}_t \rangle^2}{2L_{\mathcal{A}}} \\
&\leq f(\mathbf{x}_t) - \delta^2 \frac{\langle \nabla_{\|} f(\mathbf{x}_t), \mathbf{z}_t \rangle^2}{2L_{\mathcal{A}}} \\
&= f(\mathbf{x}_t) - \delta^2 \frac{\langle -\nabla_{\|} f(\mathbf{x}_t), \mathbf{z}_t \rangle^2}{2L_{\mathcal{A}}} .
\end{aligned}
$$

Where $\|\mathbf{d}\|_{\mathcal{A}^*} := \sup\{\langle \mathbf{z}, \mathbf{d} \rangle, \mathbf{z} \in \mathcal{A}\}$ is the dual of the atomic norm. Therefore, by definition:

$$
\langle -\nabla_{\|} f(\mathbf{x}_t), \mathbf{z}_t \rangle^2 = \|\nabla_{\|} f(\mathbf{x}_t)\|^2_{\mathcal{A}^*} ,
$$

which gives:

$$
f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \delta^2 \frac{1}{2L_{\mathcal{A}}} \|\nabla_{\|} f(\mathbf{x}_t)\|^2_{\mathcal{A}^*} .
$$

From strong convexity we have that:

$$
f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu_{\mathcal{A}}}{2} \|\mathbf{y} - \mathbf{x}\|^2_{\mathcal{A}} .
$$

Fixing $\mathbf{y} = \mathbf{x}_t + \gamma(\mathbf{x}^\star - \mathbf{x}_t)$ and $\gamma = 1$ in the LHS and minimizing the RHS we obtain:

$$
\begin{aligned}
f(\mathbf{x}^\star) &\geq f(\mathbf{x}_t) - \frac{1}{2\mu_{\mathcal{A}}} \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle}{\|\mathbf{x}^\star - \mathbf{x}_t\|^2_{\mathcal{A}}} \\
&\geq f(\mathbf{x}_t) - \frac{1}{2\mu_{\mathcal{A}}} \|\nabla_{\|} f(\mathbf{x}_t)\|^2_{\mathcal{A}^*} ,
\end{aligned}
$$

where the last inequality is obtained by the fact that $\langle \nabla f(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle = \langle \nabla_{\|} f(\mathbf{x}_t), \mathbf{x}^\star - \mathbf{x}_t \rangle$and Cauchy-Schwartz. Therefore:

$$
\|\nabla f(\mathbf{x}_t)\|_{\mathcal{A}^*} \geq 2\varepsilon_t \mu_{\mathcal{A}} ,
$$

which yields:

$$
\varepsilon_{t+1} \leq \varepsilon_t - \delta^2 \frac{\mu_{\mathcal{A}}}{L_{\mathcal{A}}} \varepsilon_t . \qquad \square
$$

D.7.2 Randomized Affine Invariant Linear Rate

Theorem' D.2.1.4. *(Part 2). Let $\mathcal{A} \subset \mathcal{H}$ be a closed and bounded set. We assume that $\|\cdot\|_{\mathcal{A}}$ is a norm. Let f be $\mu_{\mathcal{A}}$-strongly convex and $L_{\mathcal{A}}$-smooth w.r.t. the norm $\|\cdot\|_{\mathcal{A}}$, both over $\langle() \mathcal{A})$. Then, Algorithm 10 converges for $t \geq 0$ as*

$$\mathbb{E}_{\mathbf{z}}[\varepsilon_{t+1}|\mathbf{x}_t] \leq \left(1 - \hat{\delta}^2 \frac{\mu_{\mathcal{A}}}{L_{\mathcal{A}}}\right)\varepsilon_t \,,$$

where $\varepsilon_t := f(\mathbf{x}_t) - f(\mathbf{x}^\star)$, and the LMO direction $\mathbf{z}$ is sampled randomly from $\mathcal{A}$, from the same distribution as used in the definition of $\hat{\delta}$.

Proof. We start by upper-bounding f using the definition of $L_{\mathcal{A}}$ as follows

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}}[f(\mathbf{x}_{t+1})] &\leq \mathbb{E}_{\mathbf{z}}\left[\min_{\gamma\in\mathbb{R}} f(\mathbf{x}_t) + \gamma\langle\nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t\rangle + \frac{\gamma^2}{2} L_{\mathcal{A}} \|\mathbf{z}\|_{\mathcal{A}}^2\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\min_{\gamma\in\mathbb{R}} f(\mathbf{x}_t) + \gamma\langle\nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t\rangle + \frac{\gamma^2}{2} L_{\mathcal{A}}\right] \\
&\leq f(\mathbf{x}_t) - \mathbb{E}_{\mathbf{z}}\left[\frac{\langle\nabla f(\mathbf{x}_t), \tilde{\mathbf{z}}_t\rangle^2}{2L_{\mathcal{A}}}\right] \\
&\leq f(\mathbf{x}_t) - \hat{\delta}^2 \frac{\langle\nabla f(\mathbf{x}_t), \mathbf{z}_t\rangle^2}{2L_{\mathcal{A}}} \\
&= f(\mathbf{x}_t) - \hat{\delta}^2 \frac{\langle\nabla_{\|} f(\mathbf{x}_t), \tilde{\mathbf{z}}_t\rangle^2}{2L_{\mathcal{A}}} \\
&= f(\mathbf{x}_t) - \hat{\delta}^2 \frac{\langle -\nabla_{\|} f(\mathbf{x}_t), \tilde{\mathbf{z}}_t\rangle^2}{2L_{\mathcal{A}}} \,.
\end{aligned}$$

The rest of the proof proceeds as in Part 1 of the proof of Theorem D.2.1.4. □

D.8 Accelerated Matching Pursuit

Our proof follows the technique for acceleration given in [131, 169, 176, 229]

D.8.1 Proof of Convergence

We define $\|\mathbf{x}\|_{\mathbf{P}}^2 = \mathbf{x}^\top \mathbf{P}\mathbf{x}$. We start our proof by first defining the model function ψ_t. For $t = 0$, we define :

$$\psi_0(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_0\|_{\mathbf{P}}^2 \,.$$

Algorithm 13 Accelerated Matching Pursuit

init $\mathbf{x}_0 = \boldsymbol{b}_0 = \mathbf{y}_0$, $\beta_0 = 0$, and v
for $t = 0, 1 \dots T$
  Solve $\alpha_{t+1}^2 Lv = \beta_t + \alpha_{t+1}$
  $\beta_{t+1} = \beta_t + \alpha_{t+1}$
  $\tau_t = \frac{\alpha_{t+1}}{\beta_{t+1}}$
  Compute $\mathbf{y}_t = (1-\tau_t)\mathbf{x}_t + \tau_t \boldsymbol{b}_t$
  Find $\mathbf{z}_t := \mathrm{LMO}_{\mathcal{A}}(\nabla f(\mathbf{y}_t))$
  $\mathbf{x}_{t+1} = \mathbf{y}_t - \frac{\langle \nabla f(\mathbf{y}_t), \mathbf{z}_t \rangle}{L\|\mathbf{z}_t\|_2^2}\mathbf{z}_t$
  sample $\tilde{\mathbf{z}}_t \sim \mathcal{Z}$
  $\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \alpha_{t+1}\langle \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t \rangle \tilde{\mathbf{z}}_t$
end for

Then for $t > 1$, ψ_t is inductively defined as

$$\psi_{t+1}(\mathbf{x}) = \psi_t(\mathbf{x}) + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big). \tag{D.13}$$

Proof of Lemma D.3.2.1. We will prove the statement inductively. For $t = 0$, $\psi_0(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_0\|_{\mathbf{P}}^2$ and so the statement holds. Suppose it holds for some $t \geq 0$. Observe that the function $\psi_t(\mathbf{x})$ is a quadratic with Hessian $\mathbf{P}$. This means that we can reformulate $\psi_t(\mathbf{x})$ with minima at $\boldsymbol{b}_t$ as

$$\psi_t(\mathbf{x}) = \psi_t(\boldsymbol{b}_t) + \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2.$$

Using this reformulation,

$$\begin{aligned}
\arg\min_{\mathbf{x}} \psi_{t+1}(\mathbf{x}) &= \arg\min_{\mathbf{x}} \Big\{ \psi_t(\mathbf{x}) + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big) \Big\} \\
&= \arg\min_{\mathbf{x}} \Big\{ \psi_t(\boldsymbol{b}_t) + \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2 + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big) \Big\} \\
&= \arg\min_{\mathbf{x}} \Big\{ \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2 + \alpha_{t+1}\langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \boldsymbol{b}_t)\rangle \Big\} \\
&= \boldsymbol{b}_t - \alpha_{t+1}\langle \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t \rangle \tilde{\mathbf{z}}_t \\
&= \boldsymbol{b}_{t+1}.
\end{aligned}$$

□

Lemma D.8.1.1 (Upper bound on $\psi_t(\mathbf{x})$)**.**

$$\mathbb{E}[\psi_t(\mathbf{x})] \leq \beta_t f(\mathbf{x}) + \psi_0(\mathbf{x}).$$

Proof. We will also show this through induction. The statement is trivially true for $t = 0$

since $\beta_0 = 0$. Assuming the statement holds for some $t \geq 0$,

$$\begin{aligned}
\mathbb{E}[\psi_{t+1}(\mathbf{x})] &= \mathbb{E}\Big[\psi_t(\mathbf{x}) + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big)\Big] \\
&= \mathbb{E}\Big[\psi_t(\mathbf{x})\Big] + \alpha_{t+1}\mathbb{E}\Big[\Big(f(\mathbf{y}_t) + \langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big)\Big] \\
&\leq \beta_t f(\mathbf{x}) + \psi_0(\mathbf{x}) + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \nabla f(\mathbf{y}_t)^\top \mathbb{E}\Big[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top\Big]\mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big) \\
&= \beta_t f(\mathbf{x}) + \psi_0(\mathbf{x}) + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \nabla f(\mathbf{y}_t)^\top \mathbf{P}^{-1}\mathbf{P}(\mathbf{x} - \mathbf{y}_t)\rangle\Big) \\
&= \beta_t f(\mathbf{x}) + \psi_0(\mathbf{x}) + \alpha_{t+1}\Big(f(\mathbf{y}_t) + \nabla f(\mathbf{y}_t)^\top (\mathbf{x} - \mathbf{y}_t)\rangle\Big) \\
&\leq \beta_t f(\mathbf{x}) + \psi_0(\mathbf{x}) + \alpha_{t+1} f(\mathbf{x})\,.
\end{aligned}$$

In the above, we used the convexity of the function $f(\mathbf{x})$ and the definition of $\mathbf{P}$. □

Lemma D.8.1.2 (Bound on progress)**.** *For any $t \geq 0$ of algorithm 13,*

$$f(\mathbf{x}_{t+1}) - f(\mathbf{y}_t) \leq -\frac{1}{2L\|\mathbf{z}_t\|_2^2}\nabla f(\mathbf{y}_t)^\top \Big[\mathbf{z}_t \mathbf{z}_t^\top\Big]\nabla f(\mathbf{y}_t)\,.$$

Proof. The update $\mathbf{x}_{t+1}$ along with the smoothness of $f(\mathbf{x})$ guarantees that for $\gamma_{t+1} = \frac{\langle \nabla f(\mathbf{y}_t), \mathbf{z}_t\rangle}{L\|\mathbf{z}_t\|^2}$,

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &= f(\mathbf{y}_t + \gamma_{t+1}\mathbf{z}_t) \\
&\leq f(\mathbf{y}_t) + \gamma_{t+1}\langle \nabla f(\mathbf{y}_t), \mathbf{z}_t\rangle + \frac{L\gamma_{t+1}^2}{2}\|\mathbf{z}_t\|^2 \\
&= f(\mathbf{y}_t) - \frac{1}{2L\|\mathbf{z}_t\|_2^2}\nabla f(\mathbf{y}_t)^\top \Big[\mathbf{z}_t \mathbf{z}_t^\top\Big]\nabla f(\mathbf{y}_t)\,.
\end{aligned}$$

□

Lemma D.8.1.3 (Lower bound on $\psi_t(\mathbf{x})$)**.** *Given a filtration $\mathcal{F}_t$ upto time step t,*

$$\mathbb{E}[\min_{\mathbf{x}} \psi_t(\mathbf{x}) | \mathcal{F}_t] \geq \beta_t f(\mathbf{x}_t)\,.$$

Proof. This too we will show inductively. For $t = 0$, $\psi_t(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_0\|_{\mathbf{P}}^2 \geq 0$ with $\beta_0 = 0$. Assume the statement holds for some $t \geq 0$. Recall that $\psi_t(\mathbf{x})$ has a minima at $\boldsymbol{b}_t$ and can be alternatively formulated as $\psi_t(\boldsymbol{b}_t) + \frac{1}{2}\|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2$. Using this,

$$\begin{aligned}
\psi_{t+1}^\star &= \min_{\mathbf{x}}\Big[\psi_t(\mathbf{x}) + \alpha_{t+1}\Big(\Big\langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\Big\rangle + f(\mathbf{y}_t)\Big)\Big] \\
&= \min_{\mathbf{x}}\Big[\psi_t(\boldsymbol{b}_t) + \alpha_{t+1}\Big(\Big\langle \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \tilde{\mathbf{z}}_t^\top \mathbf{P}(\mathbf{x} - \mathbf{y}_t)\Big\rangle + \frac{1}{2\alpha_{t+1}}\|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2 + f(\mathbf{y}_t)\Big)\Big]
\end{aligned}$$

$$= \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \alpha_{t+1} \min_{\mathbf{x}} \left[\left\langle \mathbf{P}\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \mathbf{x} - \mathbf{y}_t \right\rangle + \frac{1}{2\alpha_{t+1}} \|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2 \right].$$

Since we defined $\mathbf{y}_t = (1-\tau_t)\mathbf{x}_t + \tau_t \boldsymbol{b}_t$, rearranging the terms gives us that

$$\mathbf{y}_t - \boldsymbol{b}_t = \frac{1-\tau_t}{\tau_t}(\mathbf{x}_t - \mathbf{y}_t).$$

Let us take now compute $\mathbb{E}[\psi_{t+1}^\star | \mathcal{F}_t]$ by combining the above two equations:

$$\begin{aligned}
\mathbb{E}[\psi_{t+1}^\star | \mathcal{F}_t] &= \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \left\langle \mathbf{P}\mathbb{E}_t[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top] \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \right\rangle \\
&\qquad + \alpha_{t+1} \mathbb{E}_t \min_{\mathbf{x}} \left[\left\langle \mathbf{P}\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \mathbf{x} - \boldsymbol{b}_t \right\rangle + \frac{1}{2\alpha_{t+1}} \|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2 \right] \\
&= \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \langle \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \\
&\qquad + \alpha_{t+1} \mathbb{E}_t \min_{\mathbf{x}} \left[\left\langle \mathbf{P}\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \nabla f(\mathbf{y}_t), \mathbf{x} - \boldsymbol{b}_t \right\rangle + \frac{1}{2\alpha_{t+1}} \|\mathbf{x} - \boldsymbol{b}_t\|_{\mathbf{P}}^2 \right] \\
&= \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \langle \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \\
&\qquad - \frac{\alpha_{t+1}^2}{2} \nabla f(\mathbf{y}_t)^\top \mathbb{E}_t \left[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \mathbf{P}\mathbf{P}^{-1} \mathbf{P}\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \right] \nabla f(\mathbf{y}_t) \\
&= \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \langle \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \\
&\qquad - \frac{\alpha_{t+1}^2}{2} \nabla f(\mathbf{y}_t)^\top \mathbb{E}_t \left[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \mathbf{P}\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \right] \nabla f(\mathbf{y}_t).
\end{aligned}$$

Let us define a constant $\nu \geq 0$ such that it is the smallest number for which the below inequality holds for all t,

$$\nu \nabla f(\mathbf{y}_t)^\top \frac{\left[\mathbf{z}_t \mathbf{z}_t^\top\right]}{2L\|\mathbf{z}_t\|_2^2} \nabla f(\mathbf{y}_t) \geq \nabla f(\mathbf{y}_t)^\top \mathbb{E}\left[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \mathbf{P}\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \right] \nabla f(\mathbf{y}_t).$$

Also recall from Lemma D.8.1.2 that

$$f(\mathbf{x}_{t+1}) - f(\mathbf{y}_t) \leq -\frac{1}{2L\|\mathbf{z}_t\|_2^2} \nabla f(\mathbf{y}_t)^\top \left[\mathbf{z}_t \mathbf{z}_t^\top \right] \nabla f(\mathbf{y}_t).$$

Using the above two statements in our computation of $\psi_{t+1}^\star$, we get

$$\mathbb{E}[\psi_{t+1}^\star | \mathcal{F}_t] = \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \langle \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle$$

$$
\begin{aligned}
&\quad - \frac{\alpha_{t+1}^2}{2} \nabla f(\mathbf{y}_t)^\top \mathbb{E}_t \left[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \mathbf{P} \tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \right] \nabla f(\mathbf{y}_t) \\
&\geq \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \langle \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \\
&\quad - \frac{\alpha_{t+1}^2 \nu}{2} \nabla f(\mathbf{y}_t)^\top \left[\mathbf{z}_t \mathbf{z}_t^\top \right] \nabla f(\mathbf{y}_t) \\
&\geq \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} \langle \nabla f(\mathbf{y}_t), \mathbf{y}_t - \mathbf{x}_t \rangle \\
&\quad + \alpha_{t+1}^2 L \nu (f(\mathbf{x}_{t+1}) - f(\mathbf{y}_t)) \\
&\geq \psi_t^\star + \alpha_{t+1} f(\mathbf{y}_t) + \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} (f(\mathbf{y}_t) - f(\mathbf{x}_t)) \\
&\quad + \alpha_{t+1}^2 L \nu (f(\mathbf{x}_{t+1}) - f(\mathbf{y}_t)) \,.
\end{aligned}
$$

Let us pick α_{t+1} such that it satisfies $\alpha_{t+1}^2 \nu L = \beta_{t+1}$. Then the above equation simplifies to

$$
\begin{aligned}
\mathbb{E}[\psi_{t+1}^\star | \mathcal{F}_t] &\geq \psi_t^\star + \frac{\alpha_{t+1}}{\tau_t} f(\mathbf{y}_t) - \frac{\alpha_{t+1}(1-\tau_t)}{\tau_t} f(\mathbf{x}_t) + \beta_{t+1}(f(\mathbf{x}_{t+1}) - f(\mathbf{y}_t)) \\
&= \psi_t^\star - \beta_t f(\mathbf{x}_t) + \beta_{t+1} f(\mathbf{y}_t) - \beta_{t+1} f(\mathbf{y}_t) + \beta_{t+1} f(\mathbf{x}_{t+1}) \\
&= \psi_t^\star - \beta_t f(\mathbf{x}_t) + \beta_{t+1} f(\mathbf{x}_{t+1}) \,.
\end{aligned}
$$

We used that $\tau_t = \alpha_{t+1}/\beta_{t+1}$. Finally we use the inductive hypothesis to conclude that

$$
\mathbb{E}[\psi_{t+1}^\star | \mathcal{F}_t] \geq \psi_t^\star - \beta_t f(\mathbf{x}_t) + \beta_{t+1} f(\mathbf{x}_{t+1}) \geq \beta_{t+1} f(\mathbf{x}_{t+1}) \,. \qquad \square
$$

Lemma D.8.1.4 (Final convergence rate)**.** *For any $t \geq 1$ the output of algorithm 13 satisfies:*

$$
\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}^\star) \leq \frac{2L\nu}{t(t+1)} \|\mathbf{x}^\star - \mathbf{x}_0\|_{\mathbf{P}}^2 \,.
$$

Proof. Putting together Lemmas D.8.1.1 and D.8.1.3, we have that

$$
\beta_t \mathbb{E}[f(\mathbf{x}_t)] \leq \mathbb{E}[\psi_t^\star] \leq \mathbb{E}[\psi_t(\mathbf{x}^\star)] \leq \beta_t f(\mathbf{x}^\star) + \psi_0(\mathbf{x}^\star) \,.
$$

Rearranging the terms we get

$$
\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}^\star) \leq \frac{1}{2\beta_t} \|\mathbf{x}^\star - \mathbf{x}_0\|_{\mathbf{P}}^2 \,.
$$

To finish the proof of the theorem, we only have to compute the value of β_t. Recall that

$$
\alpha_{t+1}^2 L \nu = \beta_t + \alpha_{t+1} \,.
$$

We will inductively show that $\alpha_t \geq \frac{t}{2L\nu}$. For $t = 0$, $\beta_0 = 0$ and $\alpha_1 = \frac{1}{2L\nu}$ which satisfies the condition. Suppose that for some $t \geq 0$, the inequality holds for all iterations $i \leq t$. Recall that $\beta_t = \sum_{i=1}^{t} \alpha_i$ i.e. $\beta_t \geq \frac{t(t+1)}{4L\nu}$. Then

$$(\alpha_{t+1}L\nu)^2 - \alpha_{t+1}L\nu = \beta_t L\nu \geq \frac{t(t+1)}{4}.$$

The positive root of the quadratic $x^2 - x - c = 0$ for $c \geq 0$ is $x = \frac{1}{2}\left(1 + \sqrt{4c+1}\right)$. Thus

$$\alpha_{t+1}L\nu \geq \frac{1}{2}\left(1 + \sqrt{t(t+1)+1}\right) \geq \frac{t+1}{2}.$$

This finishes our induction and proves the final rate of convergence. □

Lemma D.8.1.5 (Understanding ν)**.**

$$\nu \leq \max_{\mathbf{d} \in \langle () \mathcal{A}} \frac{\mathbb{E}\left[(\tilde{\mathbf{z}}_t^\top \mathbf{d})^2 \|\tilde{\mathbf{z}}_t\|_{\tilde{\mathbf{P}}}^2\right] \|\mathbf{z}(\mathbf{d})\|_2^2}{(\mathbf{z}(\mathbf{d})^\top \mathbf{d})^2},$$

Proof. Recall the definition of ν as a constant which satisfies the following inequality for all iterations t

$$\nu \nabla f(\mathbf{y}_t)^\top \frac{\left[\mathbf{z}_t \mathbf{z}_t^\top\right]}{2L\|\mathbf{z}_t\|_2^2} \nabla f(\mathbf{y}_t) \geq \nabla f(\mathbf{y}_t)^\top \mathbb{E}\left[\tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top \mathbf{P} \tilde{\mathbf{z}}_t \tilde{\mathbf{z}}_t^\top\right] \nabla f(\mathbf{y}_t).$$

which then yields the following sufficient condition for ν:

$$\nu \leq \max_{\mathbf{d} \in \langle () \mathcal{A}} \frac{\mathbb{E}\left[(\tilde{\mathbf{z}}_t^\top \mathbf{d})^2 \|\tilde{\mathbf{z}}_t\|_{\tilde{\mathbf{P}}}^2\right] \|\mathbf{z}(\mathbf{d})\|_2^2}{(\mathbf{z}(\mathbf{d})^\top \mathbf{d})^2},$$

where $\mathbf{z}(\mathbf{d})$ is defined to be

$$\mathbf{z}(\mathbf{d}) = \mathrm{LMO}_{\mathcal{A}}(-\mathbf{d}).$$

□

Proof of Theorem D.3.2.3. The proof of Theorem D.3.2.3 is exactly the same as that of the previous except that now the update to $\boldsymbol{b}_t$ is also a random variable. The only change needed is the definition of ν' where we need the following to hold:

$$\nu' \nabla f(\mathbf{y}_t)^\top \frac{1}{2L} \mathbb{E}_t\left[\mathbf{z}_t \mathbf{z}_t^\top / \|\mathbf{z}_t\|_2^2\right] \nabla f(\mathbf{y}_t) \geq \nabla f(\mathbf{y}_t)^\top \mathbb{E}\left[\mathbf{z}_t \mathbf{z}_t^\top \mathbf{P} \mathbf{z}_t \mathbf{z}_t^\top\right] \nabla f(\mathbf{y}_t).$$

Proof of Lemma D.3.2.4. When $\mathcal{A} = \{\mathbf{e}_i, i \in [n]\}$ and $\mathcal{Z}$ is a uniform distribution over $\mathcal{A}$, then $\tilde{\mathbf{P}} = 1/n\mathbf{I}$ and $\mathbf{P} = n\mathbf{I}$. A simple computation shows that $\nu' = n$ and $\nu \in [1, n]$. Note that here ν could be upto n times smaller than ν' meaning that our accelerated

greedy coordinate descent algorithm could be $\sqrt{n}$ times faster than the accelerated random coordinate descent. In the worst case $\nu = \nu'$, but in practice one can pick a smaller ν compared to ν' as the worst case gradient rarely happen. It is possible to tune ν and ν' empirically but we do not explore this directio

Appendix E

k-SVRG: Variance Reduction for Large Scale Optimization

Anant Raj[1], Sebastian U. Stich[2]

1 – MPI for Intelligent Systems, Tübingen

2 – EPFL, Lausanne

Abstract

Variance reduced stochastic gradient (SGD) methods converge significantly faster than the vanilla SGD counterpart. However, these methods are not very practical on large scale problems, as they either i) require frequent passes over the full data to recompute gradients—without making any progress during this time (like for SVRG), or ii) they require additional memory that can surpass the size of the input problem (like for SAGA).

In this work, we propose k-SVRG that addresses these issues by making best use of the *available* memory and minimizes the stalling phases without progress. We prove linear convergence of k-SVRG on strongly convex problems and convergence to stationary points on non-convex problems. Numerical experiments show the effectiveness of our method.

E.1 Introduction

We study optimization algorithms for empirical risk minimization problems $f\colon \mathbb{R}^d \to \mathbb{R}$ of the form

$$x^\star := \arg\min_x f(x)\,, \qquad \text{with} \qquad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\,, \tag{E.1}$$

where each $f_i\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth.

Problems with this structure are omnipresent in machine learning, especially in supervised learning applications [26].

Stochastic gradient descent (SGD) [205] is frequently used to solve optimization problems in machine learning. One drawback of SGD is that it does not converge at the optimal rate on many problem classes (cf. [121, 165]). *Variance reduced* methods have been introduced to overcome this challenge. Among the first of these methods were SAG [127], SVRG [98], SDCA [217] and SAGA [54]. The variance reduced methods can roughly be divided in two classes, namely i) methods that achieve variance reduction by computing (non-stochastic) gradients of f from time to time, as for example done SVRG, and ii) methods that maintain a table of previously computed stochastic gradients, such as done in SAGA.

Whilst these technologies allow the variance reduced methods to converge at a faster rate than vanilla SGD, they do not scale well to problems of very large scale. The reasons are simple: i) not only is computing a full batch gradient $\nabla f(x)$ almost inadmissible when the number of samples n is large, the optimization progress of SVRG *completely stalls* while this expensive computation takes place. This is avoided in SAGA, but ii) at the cost of $\mathcal{O}(dn)$ *additional* memory. When the data is sparse and the stochastic gradients $\nabla f_i(x)$ are not, the memory requirements can thus surpass the size of the dataset by orders of magnitude.

In this work we address these issues and propose a class of variance reduced methods that have i) shorter stalling phases of only order $\mathcal{O}(n/k)$ at the expense of only $\tilde{\mathcal{O}}(kd)$ additional memory. Here k is a parameter that can be set freely by the user. To get short stalling phases, it is advisable to set k such as to fit the capacity of the fast memory of the system. We show that the new methods converge as fast as SVRG and SAGA on convex and non-convex problems, but are more practical for large n. As a side-product of our analysis, we also crucially refine the previous theoretical analysis of SVRG, as we will outline in Section E.1.2 below.

E.1.1 SVRG, SAGA and k-SVRG

SVRG is an iterative algorithm, where in each each iteration only stochastic gradients, i.e. $\nabla f_i(x)$ for a random index $i \in [n]$, are computed, much like in SGD. In order to attain variance reduction a full gradient $\nabla f(x)$ is computed at a *snapshot* point in every few epochs. There are three issues with SVRG: i) the computation of the full gradient requires a full pass over the dataset. No progress (towards the optimal solution) is made during this time (see illustration in Figure E.1). On large scale problems, where one pass over the data might take several hours, this can yield to wasteful use of resources; ii) the theory requires the algorithm to restart at every snapshot point, resulting in discontinuous behaviour (see Fig. E.1) and iii) on strongly convex problems, the snapshot point can only be updated every $\Omega(\kappa)$ iterations (cf. [34, 98]), where $\kappa = L/\mu$ denotes the condition number (see (E.9)). When the condition number is large, this means that the algorithm

method	complexity	additional memory	*in situ* ∇f_i comp.	no full pass
Gradient Descent	$\mathcal{O}(n\kappa \log \frac{1}{\varepsilon})$	$\mathcal{O}(d)$	$\mathcal{O}(n)$	✗
SAGA	$\mathcal{O}((n+\kappa)\log \frac{1}{\varepsilon})$	$\mathcal{O}(dn)$	$\mathcal{O}(1)$	✓
SVRG	$\mathcal{O}((n+\kappa)\log \frac{1}{\varepsilon})$	$\mathcal{O}(d)$	$\mathcal{O}(n)$	✗
SCSG	$\mathcal{O}((\frac{\kappa}{\varepsilon} \wedge n+\kappa)\log \frac{1}{\varepsilon})$	$\mathcal{O}(d)$	$< n$	✓
k-SVRG	$\mathcal{O}((n+\kappa)\log \frac{1}{\varepsilon})$	$\mathcal{O}((dk+n)\log k)$	$\mathcal{O}(\frac{n}{k})$	✓

Table E.1: Comparison of running times and (additional) storage requirement for different algorithms on strongly convex functions, where $\kappa = L/\mu$ denotes the condition number. Most algorithms require *in situ* computations of many $\nabla f_i(x)$ for the same x without making progress. The longest such stalling phase is indicated, sometimes amounting to a full pass over the data (also indicated).

relies for a long time on "outdated" deterministic information. In practice—as suggested in the original paper by Johnson and Zhang [98]—the update interval is often set to $O(n)$, without theoretical justification.

SAGA circumvents the stalling phases by treating every iterate as a *partial snapshot* point. That is, for each index $i \in [n]$ a full dimensional vector is kept in memory and updated with the current value $\nabla f_i(x)$ if index i is picked in the current iteration. Hence, intuitively, in SAGA the gradient information at partial snapshot point does have more recent information about the gradient as compared to SVRG.

A big drawback of this method is the memory consumption: unless there are specific assumptions on the structure[1] of f, this requires $O(dn)$ memory (sparsity of the data does not necessarily imply sparsity of the gradients). For large scale problems it is impossible to keep all data available in fast memory (i.e. cache or RAM) which means we can not run SAGA on large scale problems which do not have GLM structure. Although SAGA can sometimes converge faster than SVRG (but not always, cf. [54]), the high memory requirements prohibit it's use. One main advantage of this algorithm is that the convergence can be proven for every single iterate[2]—thus justifying stopping the algorithm at any arbitrary time—whereas for SVRG convergence can only be proven for the snapshot points.

We propose ***k*-SVRG**, a class of algorithms that addresses the limitations of both, SAGA and SVRG. Compared to SVRG the proposed schmes have a reduced memory footprint of only $\tilde{\mathcal{O}}(kd)$ and therefore allow to optimally use the available (fast) memory. Compared to SVRG the schemes avoid long stalling phases on large scale applications (see Fig. E.1). The methods do not require restarts and show smoother convergence than SVRG (see Fig. E.1). As for SVRG, the convergence can only be guaranteed for

[1] Cf. the discussion in [54, Sec. 4].

[2] More precisely, convergence is not directly shown on the iterates, but in terms of an auxilarly Lyapunov function.

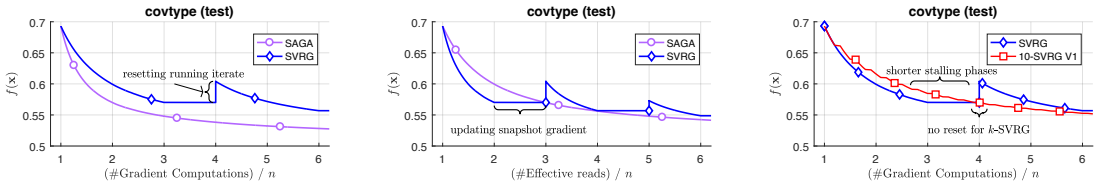


Figure E.1: Convergence behavior of SAGA, SVRG and k-SVRG. Left & Middle: SVRG recomputes the gradient at the snapshot point which yields to stalling for a full epoch both with respect to computation (left) and memory access (middle). SAGA requires only one stochastic gradient computation per iteration (left), but also one memory access (middle: roughly the identical performance as SVRG w.r.t. memory access). Right: k-SVRG does not reset the iterates at a snapshot point and equally distributes the stalling phases.

snapshot points. However, unlike as in the original SVRG, the proposed 1-SVRG updates the snapshot point every single epoch (n iterations) and thus provides more fine grained performance guarantees than the original SVRG with $\Omega(\kappa)$ iterations between snapshot points.

E.1.2 Contributions

We present k-SVRG, a limited memory variance reduced optimization algorithm that combines several good properties of SVRG as well as of SAGA. We propose two variants of k-SVRG that require to store $\tilde{\mathcal{O}}(k)$ vectors and enjoy the theoretical convergence guarantees, and one (more practical) variant that requires only $2k$ additional vectors in memory. Some key properties of our proposed approaches are:

- Low memory requirements (like SVRG, unlike SAGA): We break the memory barrier of SAGA. The required additional memory can freely be chosen by the user (parameter k) and thus all available fast memory (but not more!) can be used by the algorithm.
- Avoiding long stalling phases (like SAGA, unlike SVRG): This is in particular useful in large scale applications.
- Refinement of the SVRG analysis. To the best of our knowledge we present the first analysis that allows arbitrary sizes of inner loops, not only $\Omega(\kappa)$ as was supported by previous results.
- Linear convergence on strongly-convex problems (like SVRG, SAGA), cf. Table E.1.
- Convergence on non-convex problems (like SVRG, SAGA).

Outline. We informally introduce k-SVRG in Section E.2 and give the full details in Section E.3. All theoretical results are presented in Section E.4, the proofs can be found in Appendix E.9 and E.10. We discuss the empirical performance in Section H.4.

E.1.3 Related Work

Variance reduction alone is not sufficient to obtain the optimal convergence rate on problem (E.1). Accelerated schemes that combine the variance reduction with momentum as in Nesterov's acceleration technique [168] achieve optimal convergence rate [6, 137]. We do not discuss accelerated methods in this paper, however, we assume that it should be possible to accelerate the presented algorithm with the usual techniques.

There have also been significant efforts in developing stochastic variance reduced methods for non-convex problems [7, 10, 186, 199, 200, 213]. We will especially build on the technique proposed in [200] to derive the convergence analysis in the non-convex setting.

Recent work has also addressed the issue of making the stalling phase of SVRG shorter. In [132, 133] the authors propose SCSG, a method that makes only a batch gradient update instead of a full gradient update. However, this gives a slower rate of convergence (cf. Table E.1). In another line of work, there was an effort to combine the SVRG and SAGA approach in an asynchronus optimization setting [199] (HSAG) to run different updates in parallel. HSAG interpolates between SAGA and SVRG "per datapoint" which means snapshot points corresponding to indices in a (fixed) set S are updated like in SAGA, whereas all other snapshot points are updated after each epoch. This is orthogonal to our approach: we treat all datapoints "equally". All snapshot points are updated in the same, block-wise fashion. Also, convergence of HSAG is not guaranteed for every value of k. In another line of work Hofmann *et al.* [89] studied a version of SAGA with more than one update per iteration.

E.2 k-SVRG: A Limited Memory Approach

In this section, we informally introduce our proposed limited memory algorithm k-SVRG. For this, we will first present a unified framework that allows us to describe the algorithms SVRG and SAGA in concise notation. Let $x_0, x_1, \ldots, x_T$ denote the iterates of the algorithm, where $x_0 \in \mathbb{R}^d$ is the starting point. For each component f_i, $i \in [n]$, of the objective function (E.1) we denote by $\theta_i \in \mathbb{R}^d$ the corresponding snapshot point. The updates of the algorithms take the form

$$\begin{aligned} x_{t+1} &= x_t - \eta g_{i_t}(x_t), \qquad \text{with} \\ g_{i_t}(x_t) &:= \nabla f_{i_t}(x_t) - \nabla f_{i_t}(\theta_{i_t}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\theta_i), \end{aligned} \tag{E.2}$$

where $\eta > 0$ denotes the stepsize, and $i_t \in [n]$ an index (typically selected uniformly at random from the set $[n]$). The updates of SVRG and SAGA can both be written in this general form, as we will review now.

SVRG As mentioned before, SVRG maintains only one active snapshot point x, i.e.

$\theta_i = x$ for all $i \in [n]$. Instead of storing all components $\nabla f_i(x)$ separately, it suffices to store one single snapshot point x as well as $\nabla f(x)$ in memory, as all components of the gradient $\nabla f_i(x)$ can be recomputed when applying the update (E.2). This results in a slight increase in the computation cost, but in drastic reduction in the memory footprint.

SAGA The update of SAGA takes exactly the form (E.2). In general $\theta_i \neq \theta_j$ for $i \neq j$. Thus all θ_i parameters need to be kept in memory. In practice often $\nabla f_i(\theta_i)$ is stored instead, as this avoids recomputation of $\nabla f_i(\theta_i)$.

k-SVRG As a natural interpolation between those two algorithms we propose the following: instead of maintaining just one single snapshot point or n of them, just maintain *a few*. Precisely, the proposed algorithm maintains a set of snapshot points $\Theta \subset \mathbb{R}^d$ of cardinality $\tilde{\mathcal{O}}(k \log k)$, with the property $\theta_i \in \Theta$ for each $i \in [n]$. Therefore, it suffices to store only Θ in the memory, and a mapping from each index i to its corresponding element in Θ. This needs $\tilde{\mathcal{O}}((dk+n)\log k)$ memory. Opposed to SAGA, it is not adviced to store $\nabla f_i(\theta_i)$ directly, as this would require $O(dn)$ memory.

k_2-SVRG We also propose a heuristic variant of k-SVRG that maintains at most $2k$ snapshot points. This method comes without theoretical convergence rates, however, it shows quite good performance in practice.

We will give a formal definition of the algorithm in the next Section E.3. Below we introduce some notation that will be needed later.

E.2.1 Notation

Our algorithm consists of updates of two types: updates of the iterates as in (E.2), performed in the *inner loop* and the updates of the snapshot points at the end of the inner loops (thus constituting the *outer loop*). We denote the iterates of the algorithm by x_t^m, where t denotes the counter of the inner loop (consisting of ℓ iterations), and $m \geq 0$ the counter of the outer loop. For our algorithm (unlike in SAGA), the iterate at the end of an inner loop coincides with the first iterate of the next inner loop, $x_\ell^m = x_0^{m+1}$. Whenever we only consider the iterates x_0^m we will drop the index zero for convenience.

For clarity, we will also index the snapshot points by m, that is we write θ_i^m for the snapshot point corresponding to the component f_i in the m^{th} outer loop. And consequently, $\Theta^m := \{\theta_i^m : i \in [n]\}$. Thus the update (E.2) now reads

$$\begin{aligned} x_{t+1}^m &= x_t^m - \eta g_{i_t}^m(x_t^m), \qquad \text{with} \\ g_{i_t}^m(x_t^m) &= \nabla f_{i_t}(x_t^m) - \nabla f_{i_t}(\theta_{i_t}^m) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta_i^m). \end{aligned} \tag{E.3}$$

It will be convenient to define

$$\alpha_i^m := \nabla f_i(\theta_i^m), \qquad \bar{\alpha}^m := \frac{1}{n}\sum_{i=1}^{n} \alpha_i^m. \tag{E.4}$$

Notation for Expectation. $\mathbb{E}$ denotes the full expectation with respect to the joint distribution of all chosen data points. Frequently, we will only consider the updates within one outer loop, and condition on the past iterates. Let $\mathcal{I}_t^m := \{i_0, \ldots, i_{t-1}\}$ denote the set of chosen indices in the m^{th} outer loop until the t^{th} inner loop iteration. Then $\mathbb{E}_{t,m} = \mathbb{E}_{\mathcal{I}_t^m}$ denotes the expectation with respect to the joint distribution of all indices in $\mathcal{I}_t^m$. The algorithm k-SVRG-V2 samples additional q indices, independent of $\mathcal{I}_\ell^m$ and we denote the expectation over those samples by $\mathbb{E}'_q$. Finally, we also denote $\mathbb{E}_{\ell,m}\mathbb{E}'_q$ as $\mathbb{E}'_{q,m}$ and $\mathbb{E}_{\ell,m}$ as $\mathbb{E}_m$.

E.3 The Algorithm

In this section, we present k-SVRG in detail. The pseudecode is given in Algorithm 14. k-SVRG consist of inner and outer loops similar to SVRG, however the size of the inner loops is much smaller. Recall that $t = 0, \ldots, \ell - 1$ denotes the counter of the inner loop (where $\ell = \lceil n/k \rceil$), and $m \geq 0$ denotes the counter of the outer loop. Similar as in SVRG, a new snapshot point (denoted by $\tilde{x}^{m+1}$) is computed as an average of the iterates x_t^m. However, in our case is a weighted average

$$\tilde{x}^{m+1} := \frac{1}{S_\ell}\sum_{t=0}^{\ell-1}(1-\eta\mu)^{\ell-1-t} x_t^m, \tag{E.5}$$

where the normalization S_ℓ is defined in the algorithm. Note that $\mu = 0$ for non-convex functions and the weighted average in (E.5) reduces to a uniform average.

In Algorithm 14, we describe two variants of k-SVRG. These variants differ in the way how the snapshot points θ_i^m are updated at the end of each inner loop.

V1 In k-SVRG-V1, we update the snapshot points as follows, before moving to the $(m+1)^{th}$ outerloop:

$$\theta_i^{m+1} := \begin{cases} \theta_i^m, & \text{if } i \notin \Phi^m, \\ \tilde{x}^{m+1}, & \text{otherwise.} \end{cases} \tag{E.6}$$

The set Φ^m keeps track of the selected indices in the inner loop. Hence, we don't need to store $|\Phi^m|$ copies of the the snapshot point $\tilde{x}^{m+1}$ in memory, it suffices to store one copy and the set Φ^m, as mentioned in Section E.2 before.

It is not required that the set of indices that are used to update the θ_i^m are identical with the indices used to compute $\tilde{x}^{m+1}$ in the inner loop. Moreover, also the number points

Algorithm 14 k-SVRG-V1 / k-SVRG-V2(q)

goal minimize $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$
init x_0^0, ℓ, η, μ, α_i^0 $\forall i \in [n]$, $\bar{\alpha}^0 \leftarrow \frac{1}{n}\sum_{i=1}^{n} \alpha_i^0$
$S_\ell \leftarrow \sum_{i=0}^{\ell-1}(1-\eta\mu)^i$
for $m = 0 \ldots M-1$
  init $\Phi^m \leftarrow \emptyset$
  for $t = 0 \ldots \ell - 1$
    pick $i_t \in [n]$ uniformly at random
    $\alpha_{i_t}^m \leftarrow \nabla f_{i_t}(\theta_{i_t}^m)$
    $x_{t+1}^m \leftarrow x_t^m - \eta\left(\nabla f_{i_t}(x_t^m) - \alpha_{i_t}^m + \bar{\alpha}^m\right)$
    $\Phi^m \leftarrow \Phi^m \cup \{i_t\}$
  end for
  $\tilde{x}^{m+1} \leftarrow \frac{1}{S_\ell}\sum_{t=0}^{\ell-1}(1-\eta\mu)^{\ell-1-t} x_t^m$
  $x_0^{m+1} \leftarrow x_\ell^m$
  if variant k-SVRG-V2(q)
    $\Phi^m \leftarrow$ sample without replacement (q, n)
  end if
  $\theta_i^{m+1} \leftarrow \begin{cases} \tilde{x}^{m+1}, & \text{if } i \in \Phi^m \\ \theta_i^m, & \text{otherwise} \end{cases}$
  $\bar{\alpha}^{m+1} \leftarrow \bar{\alpha}^m + \frac{1}{n}\sum_{i\in\Phi^m} \nabla f_i(\theta_i^{m+1}) - \frac{1}{n}\sum_{i\in\Phi^m} \nabla f_i(\theta_i^m)$
end for
return $\tilde{x}_M$

does not need to be the same. The following version of k-SVRG makes this independence explicit.

V2 In k-SVRG-V2(q), we sample q indices without replacement from $[n]$ at the end of the m^{th} outer loop, which form the set Φ^m, and then update the snapshot points as before in (E.6). The suggested choice of q is $\mathcal{O}(n/k)$, and whenever we drop the argument, we simply set $q = \ell = \lceil n/k \rceil$.

Memory Requirement. To estimate the memory requirement we need to know the number of different elements in the set Θ of snapshot points. The well-studied Coupon-Collector problem (cf. [91]) tells us that in expectation there are $\mathcal{O}(n\log n)$ uniform samples needed to pick every index of the set $[n]$ at least once. In Algorithm 14 precisely ℓ samples are picked in each iteration of the inner loop, which implies each single index in $[n]$ gets picked after $\mathcal{O}(k\log k)$ outer loops. Thus there are in expectation only $\mathcal{O}(k\log k)$ different different snapshot points at any time ($n \leq k\ell$). These statements do also hold with high probability at the expense of additional poly-log factors in n. Thus, $\tilde{\mathcal{O}}((dk+n)\log k)$ memory suffices to invoke Algorithm 14.

We can enforce a hard limit on the memory by slightly violating the random sampling assumption: instead of sampling without replacement in k-SVRG-V2, we just process all indices according to a random permutation, and reshuffle after each epoch (the pseudocode is given in Algorithm 15 in Appendix E.7). Clearly, as we process the indices by the order given by random permutations, each index gets picked at least once every $2n$ iterations, i.e. at least once after $2n/\ell \leq 2k$ outer loops. Therefore, there are at most $2k$ distinct snapshot points at any time.

k_2-SVRG k_2-SVRG deviates from k-SVRG-V2 on sampling of snapshot points. Instead of sampling $q = \ell$ distinct indices in each outer loop independently, we process the indices by blocks. Concretely, every k^{th} outer loop we sample a random partition $[n] = \mathcal{P}_0^m \cup \cdots \cup \mathcal{P}_{k-1}^m$, $|\mathcal{P}_i| = \ell$ for $i = 0, \ldots, k-1$ independently at random, and then process the indices of the sets $\mathcal{P}_i$ the $(m+i)^{th}$ outer loop (to not clutter the nation we assumed here $n = k\ell$). We give the pseudocode for k_2-SVRG in Appendix E.7.

Remark E.3.0.1 (Implementation)**.** *One of the main advantages of k-SVRG is that no full pass over the data is required at the end of an outer loop. The update of $\tilde{x}^{m+1}$ can be computed on the fly with the help of an extra variable. To implement the update of the θ_i's, we use the compressed representation of the set Θ as discussed above. The update of $\bar{\alpha}^{m+1}$ requires 2ℓ gradient computations for k-SVRG-V2, but only ℓ for k-SVRG-V1, as*

$$\frac{1}{n}\sum_{i\in\Phi^m} \nabla f_i(\theta_i^m) = \frac{1}{n}\sum_{i\in\Phi^m} \alpha_i^m \,. \tag{E.7}$$

for computed values α_i^m for $i \in \Phi^m$.

E.4 Theoretical Analysis

In this section, we provide the theoretical analysis for the proposed algorithms from the previous section. We will first discuss the convergence in the convex case in Section E.4.1 and then later will discuss the convergence in the non-convex setting in Section E.4.2. For both cases we will assume that the functions f_i, $i \in [n]$, are L-smooth. Let us recall the definition: A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth if it is differentiable and

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|\,, \quad \forall x, y \in \mathbb{R}^d. \tag{E.8}$$

E.4.1 Strongly Convex Problems

In this subsection we additionally assume f to be μ-strongly convex for $\mu > 0$, i.e. we assume it holds:

$$f(y) \geq f(x) + \langle \nabla f(x), y-x\rangle + \frac{\mu}{2}\|y-x\|^2\,, \quad \forall x, y \in \mathbb{R}^d. \tag{E.9}$$

It will also become handy to denote $f^{\delta}(x) := f(x) - f(x^{\star})$, following the notation in [89].

Lyapunov Function. Similar as in [54] and [89], we show convergence of the algorithm by studying a suitable Lyapunov function. In fact, we are using the same family of functions as in [89] where $\mathcal{L}\colon \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is defined as follows:

$$\mathcal{L}(x,H) := \|x - x^{\star}\|^2 + \gamma\sigma H\,, \tag{E.10}$$

with $\gamma := \frac{\eta n}{L}$ and $0 \leq \sigma \leq 1$ a constant parameter that we will set later. We will evaluate this function at tuples (x^m, H^m), where $x^m = x_0^m$ are the iterates of the algorithm. In order to show convergence we therefore also need to define a sequence of parameters H^m that are updated in sync with x^m. Clearly, if $H^m \to 0$ for $m \to \infty$, then convergence of $\mathcal{L}(x^m, H^m) \to 0$ implies $x^m \to x^{\star}$. We will now proceed to define a sequence H^m with this property. It is important to note that these quantities do only show up in the analysis, but neither need to be be computed nor updated by the algorithm.

Similar as in [89], we will define quantities H_i^m with the property $H_i^m \geq \|\alpha_i^m - \nabla f_i(x^{\star})\|^2$, and thus their sum, $H^m := \frac{1}{n}\sum_{i=1}^n H_i^m$ is an upper bound on $\mathbb{E}\|\alpha_i^m - \nabla f_i(x^{\star})\|^2$. Let us now proceed to precisely define H_i^m. For this let $h_i^m\colon \mathbb{R}^d \to \mathbb{R}$ be defined as

$$h_i^m(x) := f_i(x) - f_i(x^{\star}) - \langle x - x^{\star}, \nabla f_i(x^{\star})\rangle\,. \tag{E.11}$$

We initialize (conceptually) $\alpha_i^0 = 0$ and $H_i^0 = \|\nabla f_i(x^{\star})\|^2$ for $i \in [n]$, and then update the bounds H_i^m in the following manner:

$$H_i^{m+1} = \begin{cases} 2Lh_i^m(\tilde{x}^{m+1}), & \text{if } i \in \Phi^m, \\ H_i^m, & \text{otherwise.} \end{cases} \tag{E.12}$$

Here Φ^m denotes the set of indices that are used to compute $\tilde{x}^{m+1}$ in either k-SVRG-V1 or k-SVRG-V2, see Algorithm 14.

Convergence Results. We now show the linear convergence of k-SVRG-V1 (Theorem E.4.1.2) and k-SVRG-V2 (Theorem E.4.1.1).

Theorem E.4.1.1. *Let $\{x^m\}_{m\geq 0}$ denote the iterates in the outer loop of k-SVRG-V2(q). If $\mu > 0$, parameter $q \geq \frac{\ell}{3}$, and step size $\eta \leq \frac{1}{3(\mu n + 2L)}$ then*

$$\mathbb{E}'_{q,m}\mathcal{L}(x^{m+1}, H^{m+1}) \leq \big(1 - \eta\mu\big)^{\ell}\mathcal{L}(x^m, H^m)\,. \tag{E.13}$$

Proof Sketch. By applying Lemmas E.4.1.3 and E.4.1.4, we directly get the following

relation:

$$\mathbb{E}_m\|x^{m+1}-x^\star\|^2+\gamma\sigma\mathbb{E}'_{q,m}H^{m+1}\leq(1-\eta\mu)^\ell\|x^m-x^\star\|^2+p_2H^m-r_2\mathbb{E}_m f^\delta(\tilde{x}^{m+1}),\tag{E.14}$$

where p_2 and r_2 are constants that will be specified in the proof. From this expression it becomes clear that we get the statement of the theorem if we can ensure $p_2\leq(1-\eta\mu)^\ell$ and $r_2\geq 0$. These calculations will be detailed in the proof in Appendix E.9. □

Theorem E.4.1.2. *Let $\{x^m\}_{m\geq 0}$ denote the iterates in the outer loop of k-SVRG-V1. If $\mu>0$, and step size $\eta\leq\frac{2\left(1-\frac{\ell-1}{2n}\right)}{5(\mu n+2L)}<\frac{1}{5(\mu n+2L)}$ then*

$$\mathbb{E}_m\mathcal{L}(x^{m+1},H^{m+1})\leq(1-\eta\mu)^\ell\mathcal{L}(x^m,H^m).\tag{E.15}$$

Proof. The proof of Theorem E.4.1.2 is very similar to the one of Theorem E.4.1.1. A detailed proof is provided in the Appendix E.9. □

Let us state a few observations:

Remark E.4.1.1 (Convergence rate)**.** *Both results show convergence at a linear rate. The convergence factor $(1-\eta\mu)$ is the same that appears also in the convergence rates of SVRG and SAGA. For SAGA a decrease by this factor can be show in every iteration for the corresponding Lyapunov function. Thus, after ℓ steps, SAGA achieves a decrease of $(1-\eta\mu)^\ell$, i.e. of the same order[3] as k-SVRG. On the other hand, the proof for SVRG shows decrease by a constant factor after κ iterations. The same improvement is attained by k-SVRG after $\min\{\lceil n/\ell\rceil,\lceil\kappa/\ell\rceil\}$ inner loops, i.e. $\min\{n,\kappa\}$ total updates. Hence, our rates do not fundamentally differ from the rates of SVRG and SAGA (in case $n\gg\kappa$ we even improve compared to the former method), but they provide an interpolation between both results.*

Remark E.4.1.2 (Relation to SVRG)**.** *For $k=1$ and $q=\ell=n$, our algorithms resemble SVRG with geometric averaging. However, our proof gives the flexibility to prove convergence of SVRG with inner loop size n, instead of $\Omega(n+\kappa)$ as in [98]. The analysis of SVRG is further strengthened in many subtle details, for instance we don't require $x^m=\tilde{x}^m$ as in vanilla SVRG, we have shorter stalling phases (for $k\gg 1$) and the possibility to choose q and ℓ differently opens more possibilities for tuning.*

Remark E.4.1.3 (Relation to SAGA)**.** *In SAGA, exactly one snapshot point is updated per iteration. The same number of updates are performed (on average) per iteration for the setting $q=\ell$. Hofmann* et al. *[89] study a variant of SAGA that performs more updates per iteration ($q\geq\ell$), but there was no proposal of choosing $q<\ell$.*

[3] Note, the decrease is not exactly identical if different stepsizes are used.

Remark E.4.1.4 (Dependence of the convergence rate on q and k)**.** *For ease of presentation we have state here the convergence results in a simplified way, omitting dependence on k entirely (see also Remark E.4.1.1). However, some mild dependencies can be extracted from the proof. For instance, it is intuitively clear that choosing a larger q in Theorem E.4.1.1 should yield a better rate. This is indeed true. Moreover, also setting $q < \ell/3$ smaller will still give linear convergence, but at a lower rate. For our application we aim to choose q as small as possible (reducing computation), without sacrificing too much in the convergence rate.*

In the rest of this subsection, we will give some tools that are required to prove Theorems E.4.1.1 and E.4.1.2. The proof of both statements is given in Appendix E.9. Lemma E.4.1.3 establishes a recurrence relation between subsequent iterates in the outer loop.

Lemma E.4.1.3. *Let $\{x^m\}_{m\geq 0}$ denote the iterates in the outer loop of Algorithm 14. Then it holds:*

$$\mathbb{E}_m\|x_0^{m+1}-x^\star\|^2 \leq (1-\eta\mu)^\ell\|x_0^m-x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_m\left[f^\delta(\tilde{x}^{m+1})\right] \\ +2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m-\nabla f_i(x^\star)\|^2, \qquad \text{(E.16)}$$

where $\tilde{x}^{m+1} = \frac{1}{S_\ell}\sum_{t=0}^{\ell-1}(1-\eta\mu)^{\ell-1-t}x_t^m$ and $S_\ell = \sum_{t=0}^{\ell-1}(1-\eta\mu)^t$.

We further need to bound the expression $\|\alpha_i^m-\nabla f_i(x^\star)\|^2$ that appears in the right hand side of equation (E.16). Recall that we have already introduced bounds $H_i^m \geq \|\alpha_i^m-\nabla f_i(x^\star)\|^2$ for this purpose. We now follow closely the machinery that has been developed in [89] in order to show how these bounds decrease (in expectation) from one iteration to the next.

Lemma E.4.1.4. *Let the sequence $\{H^m\}_{m\geq 0}$ be defined as in Section E.4.1 and updated according to equation (E.12) and let $\{\tilde{x}^m\}_{m\geq 0}$ denote the sequence of snapshot points in Algorithm 14. Then it holds:*

$$\mathbb{E}_m H^{m+1} = \frac{2LQ_\ell}{n}\mathbb{E}_m f^\delta(\tilde{x}^{m+1}) + \left(1-\frac{1}{n}\right)^\ell H^m, \qquad \textit{(for k-SVRG-V1)} \qquad \text{(E.17)}$$

$$\mathbb{E}'_{q,m} H^{m+1} = \frac{2Lq}{n}\mathbb{E}_m f^\delta(\tilde{x}^{m+1}) + \left(1-\frac{q}{n}\right)H^m, \qquad \textit{(for k-SVRG-V2)} \qquad \text{(E.18)}$$

where $Q_\ell = \sum_{t=0}^{\ell-1}\left(1-\frac{1}{n}\right)^t$.

E.4.2 Non-convex Problems

In this section, we discuss the convergence of the proposed algorithm for non-convex problems. In order to employ Algorithm 14 on non-convex problems we use the setting $\mu = 0$. We limit our analysis for only non-convex smooth functions.

Throughout the section, we assume that each f_i is L-smooth (E.8), and provide the convergence rate of algorithm k-SVRG-V2 only. However, convergence of the algorithm k-SVRG-V1 for the non-convex case can be shown in the similar way as for k-SVRG-V2. The convergence also extends to the class of gradient dominated functions by standard techniques (cf. [10, 200]). We follow the proof technique from [200] to provide the theoretical justification of our approach. However, the proof is not straight forward, due to the difficulty that is imposed by the block wise update of the snapshot points in k-SVRG-V2.

Lyapunov Function. For the analysis of our algorithms, we again choose a suitable Lyapunov function similar to the one chosen in [200]. In the following, let M denote the total number of outer loops performed. For $m = 0, \dots, M$ define $\mathcal{L}^m \colon \mathbb{R}^d \times R$ as:

$$\mathcal{L}^m(x) := f(x) + \frac{c^m}{n} \sum_{i=1}^{n} \|x_0^m - \theta_i^m\|^2 , \tag{E.19}$$

where $\{c^m\}_{m=0}^M$ denotes a sequence of parameters that we will introduce shortly (note the superscript indices). By initializing $\theta_i^0 = x^0$ we have $\mathcal{L}^0(x^0) = f(x^0)$. If we define the sequence $\{c^m\}_{m=0}^M$ such that it holds $c^M = 0$ then $\mathcal{L}^M(x^M) = f(x^M)$. These two properties will be exploited in the proof below.

Similar to the previous section, we define quantities $H^m := \frac{1}{n}\sum_{i=1}^n H_i^m$ with $H_i^m := \|x_0^m - \theta_i^m\|^2$. With this notation we can equivalently write $\mathcal{L}^m(x) = f(x) + c^m H^m$. We now define the sequence $\{c^m\}_{m=0}^M$ and an auxiliary sequence $\{\Gamma^m\}_{m=1}^M$ that will be used in the proof:

$$c^m := c^{m+1}\big(1 - \frac{\ell}{n} + \gamma\eta\ell + 4b_1\eta^2L^2\ell^2\big) + 2b_1\eta^2L^3\ell , \tag{E.20}$$

$$\Gamma^m := \eta - c^{m+1}\frac{\eta}{\gamma} - b_1\eta^2L - 2b_1c^{m+1}\eta^2\ell , \tag{E.21}$$

with $b_1 := (1 - 2L^2\eta^2\ell^2)^{-1}$ and $\gamma \geq 0$ a parameter that will be specified later. As mentioned, we will set $c^M = 0$ and (E.20) provides the values of c^m for $m = M-1, \dots, 0$. It will be convenient to denote the update in the m^{th} outer loop and t^{th} inner loop with v_t^m, that is $x_{t+1}^m = x_t^m - \eta v_t^m$. Then we can define a matrix V^m that consists of the columns v_t^m for $t = 0, \dots, \ell-1$ and a matrix ∇F^m that consists of columns $\nabla f(x_t^m)$ for $t = 0, \dots, \ell-1$. Here $\|\cdot\|_F$ denotes the Frobenius norm. By the notation just defined we have $\|V^m\|_F^2 = \sum_{t=0}^{\ell-1}\|v_t^m\|^2$ and by the tower property of conditional expectations $\mathbb{E}_m\|V^m\|_F^2 = \sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2$. By similar reasoning

$$\mathbb{E}_m\|\nabla F^m\|_F^2 = \sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|\nabla f(x_t^m)\|^2 = \sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2 . \tag{E.22}$$

Convergence Results. Now we provide the main theoretical result of this subsection. Theorem E.4.2.1 shows sub-linear convergence for non-convex functions.

Theorem E.4.2.1. *Let* $\{x_t^m\}_{t=0,m=0}^{\ell-1,M}$ *denote the iterates of* k-SVRG-V2. *Let* $\{c^m\}_{m=0}^{M}$ *be defined as in* (E.20) *with* $c^M = 0$ *and* $\gamma \geq 0$ *and such that* $\Gamma^m > 0$ *for* $m = 0, \ldots, M-1$. *Then:*

$$\sum_{m=0}^{M-1} \mathbb{E}\|\nabla F^m\|_F^2 \leq \frac{f(x_0^0) - f^\star}{\Gamma}, \tag{E.23}$$

where $\Gamma := \min_{0\leq m\leq M-1} \Gamma^m$. *In particular, for parameters* $\eta = \frac{1}{5Ln^{2/3}}$, $\gamma = \frac{L}{n^{1/3}}$ *and* $\ell = \frac{3}{2}n^{1/3}$ *and* $n > 15$ *it holds:*

$$\sum_{m=0}^{M-1} \mathbb{E}\|\nabla F^m\|_F^2 \leq 15Ln^{2/3}\left(f(x_0^0) - f^\star\right). \tag{E.24}$$

Proof Sketch. We need to rely on some technical results that will be presented in Lemmas E.4.2.2, E.4.2.3 and E.4.2.4 below. Equation (E.23) can be readily be derived from Lemma E.4.2.4 by first taking expectation and then using telescopic summation. Since $\Gamma = \min_{0\leq m\leq M-1} \Gamma^m$, we get:

$$\Gamma \sum_{m=0}^{M-1} \mathbb{E}\|\nabla F^m\|^2 \leq \mathbb{E}\mathcal{L}^0(x_0^m) - \mathbb{E}\mathcal{L}^M(x_0^{m+1}). \tag{E.25}$$

By setting $\theta_i^0 = x_0^0$ for $i = 1, \ldots, n$ we have $\mathcal{L}^0(x_0^0) = f(x_0^0)$ and as $c^M = 0$ clearly $\mathcal{L}^M(x_0^M) = f(x_0^M)$. We find a lower bound on Γ as a final step in our proof. Details about all the constants are given in detail in the Appendix E.10. □

Remark E.4.2.1 (Upper bound on ℓ)**.** *It is important to note here that unlike in the convex setting, Theorem E.4.2.1 does not allow to set the number of steps in the inner loop, i.e.* ℓ*, arbitrarily large. That essentially means that the number of snapshot points cannot be reduced below a certain threshold in k-SVRG-V2 for non-convex problems. The limitation on* ℓ *occurs due to the fact that we cannot work with a Lyapunov function which only depends on the inner loop iteration as done in [201] and hence the expected variance keeps on adding itself to the next variance term which finally gives an extra dependence of the order* ℓ^2*. But we do believe that the limitation on* ℓ *can be improved further. Besides that limitation on* ℓ*, we get the same convergence rate for our method as that of non-convex SVRG and non-convex SAGA.*

Now we discuss the lemmas which are helpful in proving Theorem E.4.2.1. The proofs of these lemmas are deferred to Appendix E.10. Lemma E.4.2.2 establishes the recurrence relation between the second term of the Lyapunov function, H^{m+1}, with H^m.

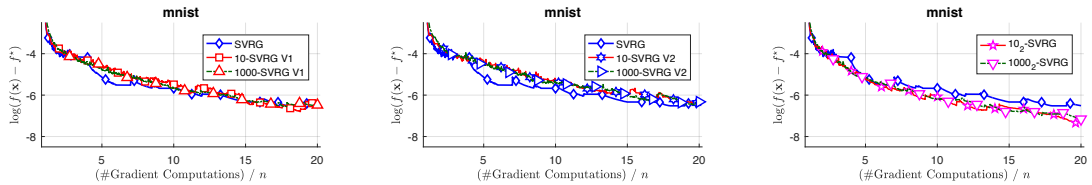


Figure E.2: Residual loss on *mnist* for SVRG, k-SVRG-V1 (left), k-SVRG-V2 (middle) and k_2-SVRG (right) for $k = \{10, 1000\}$.

Lemma E.4.2.2. *Consider the setting of Theorem E.4.2.1. Then, conditioned on the iterates obtained before the m^{th} outer loop, it holds for $\gamma > 0$:*

$$\mathbb{E}'_{\ell,m} H^{m+1} \leq \eta^2 \ell \, \mathbb{E}_{\ell,m} \|V^m\|_F^2 + \left(1 - \frac{\ell}{n}\right) \frac{\eta}{\gamma} \, \mathbb{E}_{\ell,m} \|\nabla F^m\|_F^2 + (1 + \gamma\eta\ell) \left(1 - \frac{\ell}{n}\right) H^m . \tag{E.26}$$

This result suggests that we now should relate the variance of the stochastic gradient update with the expected true gradient and the Lyapunov function. This is done in Lemma E.4.2.3, with the help of the result from Lemma E.10.0.2 which is provided in Appendix E.10.

Lemma E.4.2.3. *Consider the setting of Theorem E.4.2.1. Upon completion of the m^{th} outer loop it holds:*

$$(1 - 2L^2\eta^2\ell^2)\mathbb{E}_m \|V^m\|_F^2 = 2\mathbb{E}_m \|\nabla F^m\|^2 + 4L^2\ell H^m . \tag{E.27}$$

Finally, we can proceed to present the most important lemma of this section from which the main Theorem E.4.2.1 readily follows.

Lemma E.4.2.4. *Consider the setting of Theorem E.4.2.1, that is c^m, c^{m+1} and $\gamma > 0$ are such that $\Gamma^m > 0$. Then:*

$$\Gamma^m \cdot \mathbb{E}_m \|\nabla F^m\|^2 \leq \mathcal{L}^m(x_0^m) - \mathbb{E}'_{\ell,m} \mathcal{L}^{m+1}(x_0^{m+1}) . \tag{E.28}$$

E.5 Experiments

To support the theoretical analysis, we present numerical results on ℓ_2-regularized logistic regression problems, i.e. problems of the form

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-b_i \langle a_i, x \rangle) + \frac{\lambda}{2} \|x\|^2 . \tag{E.29}$$

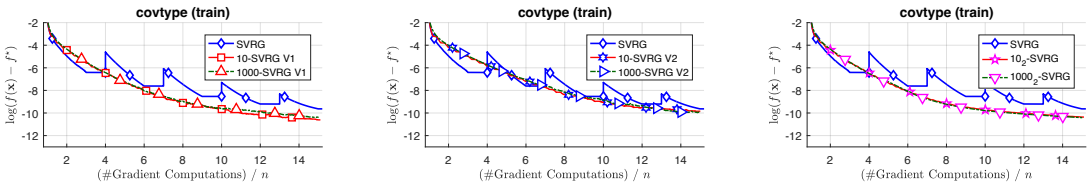


Figure E.3: Residual loss on *covtype (train)* for SVRG, k-SVRG-V1 (left), k-SVRG-V2 (middle) and k_2-SVRG (right) for $k = \{10, 1000\}$.

Dataset	d	n	L
covtype (test)	54	58 102	1311
covtype (train)	54	522 910	43 586
mnist	784	60 000	38 448

Table E.2: Summary of datasets used for experiments. We use $L = \frac{1}{4}\max_i \|a_i\|^2$, where a_i represents the i^{th} data point. The factor of 4 is due to the use of the logistic loss.

The regularization parameter λ is set to $1/n$, as in [179]. We use the datasets *covtype(train,test)* and *MNIST(binary)*[4]. Some statistics of the datasets are summarized in Table E.2. For all experiments we use $x_0 = 0$ and perform a *warm start* of the algorithms, that is we provide $\nabla f(x_0)$ as input. Several cold start procedures (where ∇f_i are injected one by one) have been suggested (cf. [54]) but discussing the effects of these heuristics is not the focus of this paper.

We conduct experiments with SAGA, SVRG (we fix the size of the inner loop to n) and the proposed k-SVRG for $k = \{1, 10, 100, 1000\}$ in all variants (k-SVRG-V1, k-SVRG-V2 and k_2-SVRG). For simplicity we use the parameters $l = q = \lceil n/k \rceil$ throughout.

The running time of the algorithms is dominated by two important components: the time for computation and the time to access the data. The actual numbers depend on the hardware and problem instances.

Gradient Computations (#GC). Fig. E.1 (left). We count the number of gradient evaluations of the form $\nabla f_i(x)$. In SAGA, each step of the inner loop only comprises one computation, whereas for SVRG, two gradients have to be computed in the inner loop. The figure nicely depicts the stalling of SVRG after one pass over the data (when a full gradient has to be computed *in situ*).

Effective Data Reads (#ER). Fig. E.1 (middle). We count the number of access to the data, that is when a d-dimensional vector needs to be fetched from memory. In the SVRG variants this is one data point in each iteration of the inner loop, and $\mathcal{O}(\lceil n/k \rceil)$ data points when updating the gradients (see Remark E.3.0.1). For SAGA in each

[4] All datasets are available at http://manikvarma.org/code/LDKL/download.html

Algorithm/Dataset	*covtype (test)*	*mnist*	*covtype (train)*
SVRG	$2.0/L$	$18.5/L$	$5.7/L$
k-SVRG-V1	$(1.2, 1.3, 1.7, 1.5)/L$	$(-, 17, 17, 14)/L$	
k-SVRG-V2	$(1.8, 1.7, 1.7, 1.8)/L$	$(-, 18, 17, 17.5)/L$	
k_2-SVRG	$(1.9, 1.9, 1.8, 1.8)/L$	$(-, 19, 18, 17.5)/L$	

Table E.3: Determined optimal stepsizes η for the datasets *covtype (test)* and *mnist* and parameters $k = (1, 10, 100, 1000)$.

iteration two values have to be fetched. For the k-SVRG variants the stalling phases are more equally distributed (for k large). Moreover, there is no big jump in function value as the current iterate does not have to be updated (a difference to SVRG).

E.5.1 Illustrative Experiment, Figure E.1

For the results displayed in Figure E.1 in Section E.1.1 we set the learning rate to an artificially low value $\eta = 0.1/L$ for all algorithms. This allows to emphasize the distinctive features of each method. Figure E.4 in the appendix depicts additional k-SVRG variants for the same setting.

E.5.2 Experiments on Large Datasets

Due to the large memory constrained of SAGA, we do not run SAGA on large scale problems. Even though for every method there is a *theoretical safe* stepsize η, it is common practice to tune the stepsize according to the dataset (cf. [54, 210]). By extensive testing we determined the stepsizes that achieve the smallest training error after $10n$ #ER for *covtype (test)* and after $30n$ #ER for *mnist*.[5] The determined optimal learning rates are summarized in Table E.3. For *covtype (train)* we figured $\eta = 5.7/L$ is a reasonable setting for all algorithms.

In Figure E.2 we compare all algorithms on *mnist*. We observe that k_2-SVRG performs best on *mnist*, followed by the other k-SVRG variants which perform very similar to SVRG. In Figure E.3 we compare all algorithms on *covertype (train)* and the picture is similar: k_2-SVRG works the best, followed by k-SVRG-V1, then k-SVRG-V2 and all variants of k-SVRG outperform SVRG. We observe that the parameter k seems to affect the performance only by a small factor on these datasets. However, it is not easy to predict the best possible k without tuning it but larger values of k do not seem to make performance worse; allowing to choose k as large as supported on the system used. Additional results are displayed in Appendix E.11.

[5] We like to emphasize that the optimal stepsize crucially depend on the maximal budget. I.e. the optimal values might be different if the application demands higher or lower accuracy.

E.6 Conclusion

We propose k-SVRG, a variance reduction technique suited for large scale optimization and show convergence on convex and non-convex problems at the same theoretical rates as SAGA and SVRG. Our algorithms have a very mild memory requirement compared to SAGA and the memory can be tuned according to the available resources. By tuning the parameter k, one can pick the algorithm that fits best to the available system resources. I.e. one should pick a picking large k for systems with fast memory, and smaller k when data access is slow (in order that the additional memory still fits in RAM). This can provide a huge amount of flexibility inn distributed optimization as we can choose different k on different machine. We could also imagine that automatic tuning of k as the optimization progresses, i.e. automatically adapting to the system resources, might yield the best performance in practice. However, this feature needs to be investigated further.

For future work, we plan to extend our analysis of k_2-SVRG using tools along the line of the recently proposed analysis of reshuffled SGD [86]. From the computational point of view, it is also important to investigate if the gradients at the snapshot points could be replaced with inexact approximations of the gradients which are computationally cheaper to compute.

Proofs for Main Results

E.7 Pseudo-code for k_2-SVRG

We provide the pseudo code k_2-SVRG in Algorithm 15 below. For simplicity we assume here $n \pmod{\ell} = 0$, i.e. $n = k\ell$.

Algorithm 15 k_2-SVRG

goal minimize $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$
init x_0^0, ℓ, η, μ, α_i^0 $\forall i \in [n]$ and $\bar{\alpha}^0 \leftarrow \frac{1}{n}\sum_{i=1}^{n} \alpha_i^0$
$S_\ell \leftarrow \sum_{t=0}^{\ell-1}(1-\eta\mu)^t$
$k \leftarrow \frac{n}{\ell}$
for $m = 0 \ldots M-1$
  ind $\leftarrow$ randperm(n)
  for $j = 0 \ldots k-1$
    init $\Phi^m \leftarrow \emptyset$
    for $t = 0 \ldots \ell - 1$
      pick $i_t \in [n]$ uniformly at random
      $\alpha_{i_t}^m \leftarrow \nabla f_{i_t}(\theta_{i_t}^m)$
      $x_{t+1}^m \leftarrow x_t^m - \eta\left(\nabla f_{i_t}(x_t^m) - \alpha_{i_t}^m + \bar{\alpha}_m\right)$
      $\Phi^m \leftarrow \Phi^m \cup \{\text{ind}[j*\ell + t]\}$
    end for
    $\tilde{x}^{m+1} \leftarrow \frac{1}{S_\ell}\sum_{t=0}^{\ell-1-t}(1-\eta\mu)^{\ell-t} x_t^m$
    $x_0^{m+1} \leftarrow x_\ell^m$
    $\theta_i^{m+1} \leftarrow \begin{cases} \tilde{x}^{m+1}, & \text{if } i \in \Phi_m \\ \theta_i^m, & \text{otherwise} \end{cases}$
  end for
  $\bar{\alpha}^{m+1} \leftarrow \bar{\alpha}^m + \frac{1}{n}\sum_{i\in\Phi^m} \nabla f_i(\theta_i^{m+1}) - \frac{1}{n}\sum_{i\in\Phi^m}\nabla f_i(\theta_i^m)$
end for

Like in Algorithm 14, no full pass over the data is required at the end of the outer loop. In particular, this requires only ℓ gradient computations, as explained in Remark E.3.0.1.

E.8 Definitions and Notations

We reiterate some definitions here again before proving the main results of this paper.

Function classes. A a differentiable convex function $f\colon \mathbb{R}^d \to \mathbb{R}$ is L-smooth if:

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x-y\|^2 \ \ \forall x, y \in \mathbb{R}^d, \tag{E.30}$$

which is equivalent to

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\| \ \ \forall x, y \in \mathbb{R}^d. \tag{E.31}$$

A differentiable non-convex function is L-smooth if (E.31) holds. A differentiable convex function $f\colon \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex if

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x-y\|^2 \ \ \forall x, y \in \mathbb{R}^d. \tag{E.32}$$

Frequently, we will be denoting $f^\star := f(x^\star)$.

Series Expansion. The following observation will be useful in the analysis later. For any integer k and real number $\zeta < 1$ we have

$$(1-\zeta)^k = 1 - k\zeta + \frac{k(k-1)}{2!}\zeta^2 - \frac{k(k-1)(k-2)}{3!}\zeta^3 + \mathcal{O}(\zeta^4), \tag{E.33}$$

and it is easily verified that whenever $\zeta \leq \frac{1}{k}$:

$$(1-\zeta)^k \geq 1 - k\zeta, \tag{E.34}$$

$$(1-\zeta)^k \leq 1 - k\zeta + \frac{k(k-1)}{2}\zeta^2. \tag{E.35}$$

Frequently used Inequalities. For $a, b \in \mathbb{R}^d$ we have:

$$\|a+b\|_2^2 \leq (1+\beta^{-1})\|a\|_2^2 + (1+\beta)\|b\|_2^2, \ \ \forall \beta > 0. \tag{E.36}$$

For for $\beta = 1$ this simplifies to:

$$\|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2. \tag{E.37}$$

Also the following inequality holds:

$$-\langle a, b \rangle \leq \frac{\gamma}{2}\|a\|_2^2 + \frac{1}{2\gamma}\|b\|_2^2, \ \ \forall \gamma > 0. \tag{E.38}$$

Notation for Non-Convex Proofs (see Section E.10). As defined in equation (E.3), we have the following optimization updates:

$$x_{t+1}^m = x_t^m - \eta \left(\nabla f_{i_t}(x_t^m) - \nabla f_{i_t}(\theta_{i_t}^m) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta_i^m) \right) = x_t^m - \eta v_t^m \tag{E.39}$$

where $v_t^m = \nabla f_{i_t}(x_t^m) - \nabla f_{i_t}(\theta_{i_t}^m) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta_i^m)$ as defined in Section E.4.2. Note that $\mathbb{E}_{\{i_t\}} v_t^m = \nabla f(x_t^m)$. As defined earlier in Section E.4.2,

$$\|V^m\|_F^2 := \sum_{t=0}^{\ell-1} \|v_t^m\|^2 \qquad \text{and} \qquad \|\nabla F^m\|_F^2 := \sum_{t=0}^{\ell-1} \|\nabla f(x_t^m)\|^2 . \tag{E.40}$$

Also, we will be using the following relations which immediately follow by taking expectation:

$$\mathbb{E}_m \|V^m\|_F^2 = \sum_{t=0}^{\ell-1} \mathbb{E}_{t+1,m} \|v_t^m\|^2 \tag{E.41}$$

$$\mathbb{E}_m \|\nabla F^m\|_F^2 = \sum_{t=0}^{\ell-1} \mathbb{E}_{t+1,m} \|\nabla f(x_t^m)\|^2 = \sum_{t=0}^{\ell-1} \mathbb{E}_{t,m} \|\nabla f(x_t^m)\|^2 . \tag{E.42}$$

E.9 Proofs for Convex Problems

In this section we provide the proof of Theorems E.4.1.1 and E.4.1.2. We first mention an important lemma from [89] which relates the two consecutive iterates for SAGA.

Lemma E.9.0.1 ([89]). *For the iterate sequence of any algorithm that evolves solutions according to equation (E.2), the following holds for a single update step, in expectation over the choice of i_t given x_t:*

$$\mathbb{E}_{\{i_t\}} \|x_{t+1} - x^\star\|^2 \le (1 - \eta\mu)\|x_t - x^\star\|^2 + 2\eta^2 \mathbb{E}_{\{i_t\}} \|\alpha_{i_t} - \nabla f_{i_t}(x^\star)\|^2 - 2\eta(1 - 2\eta L) f^\delta(x_t) .$$

The result in Lemma E.9.0.1 is the initial step towards proving a similar result to relate the iterates of two consecutive outer loops, as stated in Lemma E.4.1.3.

Proof of Lemma E.4.1.3. With Lemma E.9.0.1, we obtain

$$\begin{aligned}
\mathbb{E}_{\ell,m} \|x_\ell^m - x^\star\|^2 &\le (1 - \eta\mu)\mathbb{E}_{\ell-1,m} \|x_{\ell-1}^m - x^\star\|^2 - 2\eta(1 - 2L\eta)\mathbb{E}_{\ell-1,m} f^\delta(x_{\ell-1}^m) \\
&\quad + 2\eta^2 \mathbb{E}_{\ell,m} \|\alpha_{i_t}^m - \nabla f_{i_t}(x^\star)\|^2 \\
&= (1 - \eta\mu)\mathbb{E}_{\ell-1,m} \|x_{\ell-1}^m - x^\star\|^2 - 2\eta(1 - 2L\eta)\mathbb{E}_{\ell-1,m} f^\delta(x_{\ell-1}^m) \\
&\quad + 2\eta^2 \mathbb{E}_{\{i\}} \|\alpha_i^m - \nabla f_i(x^\star)\|^2
\end{aligned}$$

We now apply Lemma E.9.0.1 recursively to find the following:

$$\begin{aligned}
\mathbb{E}_{\ell,m}\|x_\ell^m - x^\star\|^2 &\le (1-\eta\mu)\mathbb{E}_{\ell-1,m}\|x_{\ell-1}^m - x^\star\|^2 - 2\eta(1-2L\eta)\mathbb{E}_{\ell-1,m}f^\delta(x_{\ell-1}^m) \\
&\qquad + 2\eta^2\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 \\
&\le (1-\eta\mu)^2\mathbb{E}_{\ell-2,m}\|x_{\ell-2}^m - x^\star\|^2 - 2\eta(1-2L\eta)\Big[\mathbb{E}_{\ell-1,m}f^\delta(x_{\ell-1}^m) \\
&+(1-\eta\mu)\mathbb{E}_{\ell-2,m}f^\delta(x_{\ell-2}^m)\Big] + 2\eta^2\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2\left[1+(1-\eta\mu)\right] \\
&\le (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)\sum_{t=0}^{\ell-1}(1-\eta\mu)^t\mathbb{E}_{\ell-t,m}f^\delta(x_{\ell-t-1}^m) \\
&\qquad + 2\eta^2\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2\cdot\sum_{t=0}^{\ell-1}(1-\eta\mu)^t \\
&= (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)\mathbb{E}_{\ell-1,m}\left[\sum_{t=0}^{\ell-1}(1-\eta\mu)^t f^\delta(x_{\ell-t-1}^m)\right] \\
&\qquad + 2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 \\
&= (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_{\ell-1,m}\left[\sum_{t=0}^{\ell-1}\frac{(1-\eta\mu)^t}{S_\ell}f^\delta(x_{\ell-t-1}^m)\right] \\
&\qquad + 2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 \\
&= (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_{\ell,m}\left[\sum_{t=0}^{\ell-1}\frac{(1-\eta\mu)^t}{S_\ell}f^\delta(x_{\ell-t-1}^m)\right] \\
&\qquad + 2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 \\
&= (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_{\ell,m}\left[\sum_{t=0}^{\ell-1}\frac{(1-\eta\mu)^{\ell-t-1}}{S_\ell}f^\delta(x_t^m)\right] \\
&\qquad + 2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 \qquad \text{(E.43)}
\end{aligned}$$

Since f is a convex function, we have by Jensen's inequality for weights $\alpha_i \ge 0$, $\sum_{i=1}^\ell \alpha_i = 1$,

$$f\left(\sum_{i=1}^\ell \alpha_i x_i\right) \le \sum_{i=1}^\ell \alpha_i f(x_i)\,. \qquad \text{(E.44)}$$

By definition $\tilde{x}^{m+1} = \frac{1}{S_\ell}\sum_{t=0}^{\ell-1}(1-\eta\mu)^{\ell-t-1}x_t^m$ and $x_\ell^m = x_0^{m+1}$. Hence from equations (E.43) and (E.44), we get the result:

$$\mathbb{E}_m\|x_0^{m+1} - x^\star\|^2 \le (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_m f^\delta(\tilde{x}^{m+1})$$
$$+2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 \square$$

Proof of Lemma E.4.1.4 Recall that we defined $h_i^m(x) = f_i(x) - f_i(x^\star) - \langle x - x^\star, \nabla f_i(x^\star)\rangle$. It is important to note

$$\mathbb{E}_{\{i\}}[h_i^m(x)] = f(x) - f^\star = f^\delta(x)\,. \tag{E.45}$$

We need to derive an upper bound on H^{m+1}. By the update equation (E.12) we have $H_i^{m+1} = H_i$ for $i \notin \Phi^m$ and $H_i^{m+1} = 2Lh_i^m(\tilde{x}^{m+1})$ for $i \in \Phi^m$. As $\tilde{x}^{m+1}$ is not known until the inner loop has terminated, we will now proof a slightly more general statement.

Define $H^{m+1}(x) := \sum_{i\notin\Phi^m} H_i^m + \sum_{i\in\Phi^m} 2Lh_i^m(x)$. We will now proof that the claimed statements hold for $H^{m+1}(x)$ and we will put $\tilde{x}^{m+1}$ in place of x at the end of the proof.

***k*-SVRG-V1** The process can be seen as doing sampling with replacement ℓ number of times. Define the auxiliary quantities $H_i^{m,0}(x) := H_i^m$ and $H^{m,t}(x)$ by the following equation

$$H_i^{m,t}(x) = \begin{cases} 2Lh_i^m(x), & \text{if } i^{th} \text{ data point is chosen in } t^{th} \text{ inner loop iteration.} \\ H_i^{m,t-1}(x), & \text{otherwise.} \end{cases}$$

Now for any fixed but arbitrary x, we have:

$$\begin{aligned}
\mathbb{E}_{\ell,m}H^{m+1}(x) = \mathbb{E}_{\ell,m}H^{m,\ell}(x) &= \mathbb{E}_{\ell,m}\left[\frac{1}{n}\sum_{i=1}^n H_i^{m,\ell}(x)\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{\ell,m}H_i^{m,\ell}(x) \\
&= \frac{2L}{n}\mathbb{E}_{\ell,m}[h_i^m(x)] + \left(1-\frac{1}{n}\right)\mathbb{E}_{\ell-1,m}H_i^{m,\ell-1}(x) \\
&= \frac{2L}{n}f^\delta(x) + \left(1-\frac{1}{n}\right)\left[\frac{2L}{n}f^\delta(x) + \left(1-\frac{1}{n}\right)\mathbb{E}_{\ell-1,m}H_i^{m,\ell-2}(x)\right] \\
&= \frac{2L}{n}f^\delta(x)\sum_{t=i}^{\ell}\left(1-\frac{1}{n}\right)^{t-1} + \left(1-\frac{1}{n}\right)^{\ell}H^m \\
&= \frac{2LQ_\ell}{n}f^\delta(x) + \left(1-\frac{1}{n}\right)^{\ell}H^m
\end{aligned} \tag{E.46}$$

where $Q_\ell = \sum_{t=0}^{\ell-1}\left(1-\frac{1}{n}\right)^t$. Now if we replace x by $\tilde{x}^{m+1}$ we get the claimed result:

$$\mathbb{E}_m H^{m+1} = \mathbb{E}_{\ell,m}H^{m+1} = \frac{2LQ_\ell}{n}\mathbb{E}_m f^\delta(\tilde{x}^{m+1}) + \left(1-\frac{1}{n}\right)^{\ell}H^m\,. \tag{E.47}$$

***k*-SVRG-V2** Finding the relation between H^{m+1} and H^m is much more simpler for

k-SVRG-V2 as a set of independet q points are used for the update of H^{m+1}.

$$\begin{aligned}\mathbb{E}_{\ell,m}\mathbb{E}'_q H^{m+1} &= \mathbb{E}_{\ell,m}\mathbb{E}'_q H^{m,\ell} = \mathbb{E}_{\ell,m}\mathbb{E}'_q\left[\frac{1}{n}\sum_{=1}^{n} H_i^{m,\ell}\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\ell,m}\mathbb{E}'_q H_i^{m,\ell}\\ &= \frac{2Lq}{n}\mathbb{E}_{\ell,m}\mathbb{E}'_q[h_i^m(\tilde{x}^{m+1})] + \left(1-\frac{q}{n}\right)H^m\\ &= \frac{2Lq}{n}\mathbb{E}_{\ell,m} f^{\delta}(\tilde{x}^{m+1}) + \left(1-\frac{q}{n}\right)H^m, \end{aligned} \tag{E.48}$$

which is the claimed bound. □

Using the results obtained in Lemmas E.4.1.3 and E.4.1.4, we are now ready to prove the main theoretical results of the Section E.4.1.

Proof of Theorem E.4.1.1. We apply the results from Lemma E.4.1.3 and Lemma E.4.1.4 for k-SVRG-V2 to estimate the Lyapunov function:

$$\begin{aligned}\mathbb{E}'_{q,m}\mathcal{L}(x_0^{m+1}, H^{m+1}) &= \mathbb{E}_m\|x_0^{m+1}-x^\star\|^2 + \gamma\sigma\mathbb{E}'_{q,m}H^{m+1}\\ &\le (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_m f^\delta(\tilde{x}^{m+1})\\ &+ 2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 + \gamma\sigma\left[\frac{2Lq}{n}\mathbb{E}_m f^\delta(\tilde{x}^{m+1}) + \left(1-\frac{q}{n}\right)H^m\right]\\ &\le (1-\eta\mu)^\ell\|x_0^m - x^\star\| + H^m\underbrace{\left[\gamma\sigma\left(1-\frac{q}{n}\right)+2\eta^2 S_\ell\right]}_{=:p_2}\\ &\quad - \underbrace{\left(2\eta(1-2L\eta)S_\ell - \gamma\sigma\frac{2Lq}{n}\right)}_{=:r_2}\mathbb{E}_m f^\delta(\tilde{x}^{m+1}). \end{aligned} \tag{E.49}$$

Now in equation (E.49), we need to find parameters such that

$$p_2 = \gamma\sigma\left(1-\frac{q}{n}\right)+2\eta^2 S_\ell \le \gamma\sigma(1-\eta\mu)^\ell, \qquad \text{(Condition 1)}$$

$$r_2 = 2\eta(1-2L\eta)S_\ell - \gamma\sigma\frac{2Lq}{n} \ge 0. \qquad \text{(Condition 2)}$$

Condition 1: If we choose $\eta \le \frac{\sigma\frac{q}{\ell}}{\mu n+2L}$, then (Condition 1) is satisfied. We show the calculations below:

$$\begin{aligned}\gamma\sigma\left(1-\frac{q}{n}\right)+2\eta^2 S_\ell - \gamma\sigma(1-\eta\mu)^\ell &= \gamma\sigma\left(1-\frac{q}{n}\right) + \frac{2\eta^2\left(1-(1-\eta\mu)^\ell\right)}{\eta\mu} - \gamma\sigma(1-\eta\mu)^\ell\\ &= \frac{n\eta}{L}\sigma\left(1-\frac{q}{n}\right) + \frac{2\eta\left(1-(1-\eta\mu)^\ell\right)}{\mu}\end{aligned}$$

$$
\begin{aligned}
&\quad - \frac{n\eta}{L}\sigma(1-\eta\mu)^{\ell} \\
&\qquad\qquad (\gamma = \tfrac{n\eta}{L}) \\
&= \eta\left(\sigma\frac{n}{L} + \frac{2}{\mu} - \sigma\frac{q}{L}\right) - \eta\left(\sigma\frac{n}{L} + \frac{2}{\mu}\right)(1-\eta\mu)^{\ell}
\end{aligned} \tag{E.50}
$$

After division by $\eta\left(\sigma\frac{n}{L} + \frac{2}{\mu}\right)$, the right hand side reads as $1 - \frac{\sigma q}{\sigma n + 2\frac{L}{\mu}} - (1-\eta\mu)^{\ell}$, and by the observation in (E.34) we see that the condition is satisfied for $\eta \leq \frac{\sigma\frac{q}{\ell}}{\mu n + 2L}$.

Condition 2: If we choose $\eta \leq \min\left\{\frac{1}{2L}\left(1 - \frac{2q\sigma\left(n+2\frac{L}{\mu}\right)}{\ell\left((2n-q)+4\frac{L}{\mu}\right)}\right), \frac{\sigma\frac{q}{\ell}}{\mu n+2L}\right\}$ then the (Condition 2) is satisfied. The outline of the calculations are provided below. As $\gamma = \frac{\eta n}{L}$ the condition reads as

$$
\eta(1-2L\eta)S_{\ell} - \sigma\eta q \geq 0\,, \tag{E.51}
$$

which can be equivalently stated as $2L\eta \leq 1 - \sigma\frac{q}{S_{\ell}}$, or

$$
2L\eta \leq 1 - \sigma\frac{q\eta\mu}{1-(1-\eta\mu)^{\ell}}\,, \tag{E.52}
$$

using the definition of S_{ℓ}. Observe

$$
\begin{aligned}
1 - \sigma\frac{q\eta\mu}{1-(1-\eta\mu)^{\ell}} &\leq 1 - \frac{\sigma q}{\ell - \frac{\ell(\ell-1)}{2}\eta\mu} && \text{(with (E.35))} \\
&\leq 1 - \frac{\sigma q}{\ell\left(1 - \frac{\ell}{2}\eta\mu\right)} \\
&\leq 1 - \frac{\sigma q}{\ell\left(1 - \frac{\sigma q\mu}{2(\sigma n\mu + 2L)}\right)}
\end{aligned}
$$

where we used $\eta \leq \frac{\sigma\frac{q}{\ell}}{\mu n+2L}$. Since $\sigma \leq 1$ we have $\frac{\sigma q\mu}{\sigma\mu n+2L} \leq \frac{q\mu}{\mu n+2L}$ and we can further estimate:

$$
\begin{aligned}
&\leq 1 - \frac{\sigma q}{\ell\left(1 - \frac{q\mu}{2(n\mu+2L)}\right)} \\
&\leq 1 - \frac{2q\sigma\left(n+2\frac{L}{\mu}\right)}{\ell\left((2n-q)+4\frac{L}{\mu}\right)}\,.
\end{aligned} \tag{E.53}
$$

Hence, the condition in equation (E.52) is satisfied if

$$\eta \leq \frac{1}{2L}\left(1 - \frac{2q\sigma\left(n+2\frac{L}{\mu}\right)}{\ell\left((2n-q)+4\frac{L}{\mu}\right)}\right) \tag{E.54}$$

as claimed.

Finally, if we choose $q \geq \frac{\ell}{3}$ and $\sigma = \frac{\ell}{2q}\left(\frac{2L}{2L+\mu n} + \frac{2n+2L/\mu}{2n-q+4L/\mu}\right)^{-1}$ then choosing $\eta \leq \frac{1}{2(\mu n+2L)}$ satisfies both the constraints. □

Proof of Theorem E.4.1.2. We apply the result from Lemma E.4.1.3 and Lemma E.4.1.4 for k-SVRG-V1 to estimate the Lyapunov function:

$$\begin{aligned}
&\mathbb{E}_m\mathcal{L}(x_0^{m+1}, H^{m+1}) = \mathbb{E}_m\|x_0^{m+1} - x^\star\|^2 + \gamma\sigma\mathbb{E}_m H^{m+1} \\
&\leq (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 - 2\eta(1-2L\eta)S_\ell\mathbb{E}_m f^\delta(\tilde{x}^{m+1}) \\
&\qquad + 2\eta^2 S_\ell\mathbb{E}_{\{i\}}\|\alpha_i^m - \nabla f_i(x^\star)\|^2 + \gamma\sigma\left[\frac{2LQ_\ell}{n}\mathbb{E}_m f^\delta(\tilde{x}^{m+1}) + \left(1-\frac{1}{n}\right)^\ell H^m\right] \\
&\leq (1-\eta\mu)^\ell\|x_0^m - x^\star\|^2 + H^m\underbrace{\left[\gamma\sigma\left(1-\frac{1}{n}\right)^\ell + 2\eta^2 S_\ell\right]}_{=:p_1} \\
&\qquad - \underbrace{\left(2\eta(1-2L\eta)S_\ell - \gamma\sigma\frac{2LQ_\ell}{n}\right)}_{=:r_1}\mathbb{E}_m f^\delta(\tilde{x}^{m+1})
\end{aligned} \tag{E.55}$$

Now in equation (E.55), we need to find parameters such that

$$p_1 = \gamma\sigma\left(1-\frac{1}{n}\right)^\ell + 2\eta^2 S_\ell \leq \gamma\sigma(1-\eta\mu)^\ell, \qquad \text{(Condition 1)}$$

$$r_1 = 2\eta(1-2L\eta)S_\ell - \gamma\sigma\frac{2LQ_\ell}{n} \geq 0. \qquad \text{(Condition 2)}$$

Condition 1: If we choose $\eta \leq \frac{\sigma\left(1-\frac{\ell-1}{2n}\right)}{\mu n+2L}$ then (Condition 1) is satisfied. We show the calculations below:

$$\begin{aligned}
\gamma\sigma\left(1-\frac{1}{n}\right)^\ell + 2\eta^2 S_\ell &= \gamma\sigma\left(1-\frac{1}{n}\right)^\ell + 2\eta\frac{(1-\eta\mu)^\ell}{\mu} \\
&= \eta\left(\sigma\frac{n}{L}\left(1-\frac{1}{n}\right)^\ell + \frac{2}{\mu}\left(1-(1-\eta\mu)^\ell\right)\right)
\end{aligned}$$

$$\leq \eta \left(\sigma \frac{n}{L} \left(1 - \frac{\ell}{n} + \frac{\ell(\ell-1)}{2n^2} \right) + \frac{2}{\mu} \left(1 - (1-\eta\mu)^{\ell} \right) \right) \tag{E.56}$$

with (E.35). Hence, (Condition 1) is satisfied if it holds:

$$\eta \left(\sigma \frac{n}{L} \left(1 - \frac{\ell}{n} + \frac{\ell(\ell-1)}{2n^2} \right) + \frac{2}{\mu} \left(1 - (1-\eta\mu)^{\ell} \right) \right) \leq \sigma \frac{\eta n}{L} (1-\eta\mu)^{\ell}. \tag{E.57}$$

We now finish the proof similarly as the proof of (Condition 1) in the proof of Theorem E.4.1.1 above. With the help of equation (E.34) we derive that $\eta \leq \frac{\sigma\left(1-\frac{\ell-1}{2n}\right)}{\mu n+2L}$ is a sufficient condition to imply (Condition 1).

Condition 2: If we choose $\eta \leq \min \left\{ \frac{1}{2L} \left(1 - \frac{2\sigma\left(n+2\frac{L}{\mu}\right)}{2n-\ell\left(1-\frac{\ell-1}{2n}\right)+4\frac{L}{\mu}} \right), \frac{\sigma\left(1-\frac{\ell-1}{2n}\right)}{\mu n+2L} \right\}$ then (Condition 2) is satisfied. By the definition of Q_ℓ and S_ℓ, the condition can equivalently be written as

$$\begin{aligned} 2\eta(1-2L\eta)S_\ell - \gamma\sigma\frac{2L}{n}\sum_{t=1}^{\ell}\left(1-\frac{1}{n}\right)^{t-1} &= 2\eta(1-2L\eta)\frac{1-(1-\eta\mu)^{\ell}}{\eta\mu} \\ &\quad -\gamma\sigma\frac{2L}{n}\frac{1-(1-\frac{1}{n})^{\ell}}{\frac{1}{n}} \geq 0. \end{aligned} \tag{E.58}$$

From equation (E.34), we have $\left(1-\frac{1}{n}\right)^{\ell} \geq 1-\frac{\ell}{n}$. Hence it suffices to choose η such that

$$2\eta(1-2L\eta)\frac{1-(1-\eta\mu)^{\ell}}{\eta\mu} - \gamma\sigma\frac{2L\ell}{n} \geq 0. \tag{E.59}$$

We simplify the above equation further to get:

$$2\eta(1-2L\eta)\frac{1-(1-\eta\mu)^{\ell}}{\eta\mu} - \gamma\sigma\frac{2L\ell}{n} = 2\eta\left((1-2L\eta)\frac{1-(1-\eta\mu)^{\ell}}{\eta\mu} - \sigma\ell \right) \tag{E.60}$$

$$\geq 2\eta\ell \underbrace{\left((1-2L\eta)\left(1-\frac{\ell-1}{2}\eta\mu\right) - \sigma \right)}_{=:s_1}, \tag{E.61}$$

with (E.35). We will now derive a condition on η such that $s_1 \geq 0$. By rearranging the terms in s_1 we see that it suffices to hold

$$2L\eta \leq 1 - \frac{\sigma}{1-\frac{\ell-1}{2}\eta\mu} \leq 1 - \frac{\sigma}{1-\frac{\ell}{2}\eta\mu} \leq 1 - \frac{\sigma}{1-\frac{\ell}{2}\frac{\sigma\mu\left(1-\frac{\ell-1}{2n}\right)}{\sigma\mu n+2L}} \tag{E.62}$$

where we used the assumption $\eta \leq \frac{\sigma\left(1-\frac{\ell-1}{2n}\right)}{\mu n+2L}$ in the last inequality. Thus it suffices if

$$\eta \leq \frac{1}{2L}\left(1-\frac{2\sigma\left(n+2\frac{L}{\mu}\right)}{2n-\ell\left(1-\frac{\ell-1}{2n}\right)+4\frac{L}{\mu}}\right). \tag{E.63}$$

Finally, we see that if we choose $\eta \leq \frac{2\left(1-\frac{\ell-1}{2n}\right)}{5(\mu n+2L)}$ and $\sigma = \left(2\frac{L\left(1-\frac{\ell-1}{2n}\right)}{L+\mu n}+\frac{n+2\frac{L}{\mu}}{2n-\ell(1-\frac{\ell-1}{2n})+4\frac{L}{\mu}}\right)^{-1}$ (which is of the same order as the σ in the theorem E.4.1.1 upto a constant factor) then (Condition 1) and (Condition 2) both hold simultaneously. □

E.10 Proofs for Non-Convex Problems

In this section we derive the proof of Theorem E.4.2.1. First of all, we mention a result from [200] which is not directly applicable to our case as the setting is different, but which served as an inspiration for the proof.

Lemma E.10.0.1 ([200]). *Consider the SAGA updates for non-convex optimization problem where each f_i is L-smooth and $v_t = \nabla f_{i_t}(x_t) - \nabla f_{i_t}(\theta_{i_t}) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta_i)$ in equation (E.2) then:*

$$\mathbb{E}\|v_t\|^2 \leq 2\mathbb{E}\|\nabla f(x_t)\|^2 + \frac{2L^2}{n}\sum_{i=1}^n \mathbb{E}\|x_t-\theta_i\|^2. \tag{E.64}$$

We will now derive a similar statement that holds for our proposed algorithm.

Lemma E.10.0.2. *Consider the setting of Theorem E.4.2.1. Then it holds:*

$$\mathbb{E}_{t+1,m}\|v_t^m\|^2 \leq 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + 4L^2\eta^2 t\sum_{j=0}^{t-1}\mathbb{E}_{j+1,m}\|v_j^m\|^2 + \frac{4L^2}{n}\sum_{i=1}^n\|x_0^m-\theta_i^m\|^2. \tag{E.65}$$

Proof. We use the following notation, $\xi_t^m := (\nabla f_{i_t}(x_t^m) - \nabla f_{i_t}(\theta_t^m))$. Now,

$$\begin{aligned}
\mathbb{E}_{t+1,m}\|v_t^m\|^2 &= \mathbb{E}_{t+1,m}\left\|\xi_t^m + \frac{1}{n}\sum_{i=1}^n \nabla f(\theta_i^m)\right\|^2 \\
&= \mathbb{E}_{t+1,m}\left\|\xi_t^m + \frac{1}{n}\sum_{i=1}^n \nabla f(\theta_i^m) - \nabla f(x_t^m) + \nabla f(x_t^m)\right\|^2 \\
&\overset{(E.37)}{\leq} 2\mathbb{E}_{t+1,m}\left\|\nabla f(x_t^m)\right\|^2 + 2\mathbb{E}_{t+1,m}\left\|\xi_t^m - \mathbb{E}_{\{t\}}\xi_t^m\right\|^2 \\
&\leq 2\mathbb{E}_{t+1,m}\left\|\nabla f(x_t^m)\right\|^2 + 2\mathbb{E}_{t,m}\mathbb{E}_{\{t\}}\left\|\xi_t^m - \mathbb{E}_{\{t\}}\xi_t^m\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&\leq 2\mathbb{E}_{t+1,m}\left\|\nabla f(x_t^m)\right\|^2 + 2\mathbb{E}_{t,m}\mathbb{E}_{\{t\}}\left\|\xi_t^m\right\|^2 \\
&\leq 2\mathbb{E}_{t+1,m}\left\|\nabla f(x_t^m)\right\|^2 + \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}_{t,m}\|\nabla f_i(x_t^m) - \nabla f_i(\theta_i^m)\|^2 \\
&= 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + \frac{2}{n}\sum_{i=1}^{n}\mathbb{E}_{t,m}\|\nabla f_i(x_t^m) - \nabla f_i(x_0^m) + \nabla f_i(x_0^m) - \nabla f_i(\theta_i^m)\|^2 \\
&\overset{(E.37)}{\leq} 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + \frac{4}{n}\sum_{i=1}^{n}\mathbb{E}_{t,m}\|\nabla f_i(x_t^m) - \nabla f_i(x_0^m)\|^2 \\
&\qquad\qquad + \frac{4}{n}\sum_{i=1}^{n}\|\nabla f_i(x_0^m) - \nabla f_i(\theta_i^m)\|^2 \\
&\overset{(E.31)}{\leq} 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + 4L^2\mathbb{E}_{t,m}\|x_t^m - x_0^m\|^2 + \frac{4}{n}\sum_{i=1}^{n}\|\nabla f_i(x_0^m) - \nabla f_i(\theta_i^m)\|^2 \\
&= 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + 4L^2\eta^2\mathbb{E}_{t,m}\Big\|\sum_{j=0}^{t-1}v_j^m\Big\|^2 + \frac{4}{n}\sum_{i=1}^{n}\|\nabla f_i(x_0^m) - \nabla f_i(\theta_i^m)\|^2 \\
&\leq 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + 4L^2\eta^2 t\sum_{j=0}^{t-1}\mathbb{E}_{t,m}\|v_j^m\|^2 + \frac{4}{n}\sum_{i=1}^{n}\|\nabla f_i(x_0^m) - \nabla f_i(\theta_i^m)\|^2 \\
&\leq 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + 4L^2\eta^2 t\sum_{j=0}^{t-1}\mathbb{E}_{j+1,m}\|v_j^m\|^2 + \frac{4L^2}{n}\sum_{i=1}^{n}\|x_0^m - \theta_i^m\|^2 .
\end{aligned}
\tag{E.66}
$$

Hence, finally we have

$$
\mathbb{E}_{t+1,m}\|v_t^m\|^2 \leq 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2 + 4L^2\eta^2 t\sum_{j=0}^{t-1}\mathbb{E}_{j+1,m}\|v_j^m\|^2 + \frac{4L^2}{n}\sum_{i=1}^{n}\|x_0^m - \theta_i^m\|^2 .\square
$$

Lemma E.10.0.3. *Consider the iterates $\{x_t^m\}$ of Algorithm 14 and the new snapshot point at the end of the m^{th} outer loop, $\tilde{x}^{m+1} = \frac{1}{\ell}\sum_{t=0}^{\ell-1}x_t^m$. Then the following relation holds:*

$$
\begin{aligned}
\mathbb{E}_m\|x^{m+1} - \tilde{x}^{m+1}\|^2 \leq \frac{\eta^2(\ell+1)(2\ell+1)}{6\ell}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|v_t^m\|^2 &\leq \eta^2\ell\mathbb{E}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|v_t^m\|^2 \\
&= \eta^2\ell\mathbb{E}_{\ell,m}\|V^m\|_F^2 .
\end{aligned}
\tag{E.67}
$$

Proof.

$$
\begin{aligned}
\mathbb{E}_m\|x^{m+1} - \tilde{x}^{m+1}\|^2 &= \mathbb{E}_{\ell,m}\|x^{m+1} - \tilde{x}^{m+1}\|^2 = \mathbb{E}_{\ell,m}\|x_\ell^m - \tilde{x}^{m+1}\|^2 \\
&= \mathbb{E}_{\ell,m}\left\|x_\ell^m - \frac{1}{\ell}\sum_{t=0}^{\ell-1}x_t^m\right\|^2 = \frac{1}{\ell^2}\mathbb{E}_{\ell,m}\Big\|\sum_{t=0}^{\ell-1}(x_\ell^m - x_t^m)\Big\|^2
\end{aligned}
$$

$$= \frac{1}{\ell^2}\mathbb{E}_{\ell,m}\Big\| - \eta \sum_{t=0}^{\ell-1}(i+1)v_t^m\Big\|^2$$
$$= \frac{\eta^2}{\ell^2}\mathbb{E}_{\ell,m}\Big\| \sum_{t=0}^{\ell-1}(i+1)v_t^m\Big\|^2 . \tag{E.68}$$

Applying Cauchy-Schwarz in (E.68) gives,

$$\mathbb{E}_{\ell,m}\Big\| \sum_{t=0}^{\ell-1}(i+1)v_t^m\Big\|^2 \leq \frac{\ell(\ell+1)(2\ell+1)}{6}\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2 , \tag{E.69}$$

from which the final expression follows:

$$\mathbb{E}_m\|x^{m+1}-\tilde{x}^{m+1}\|^2 \leq \frac{\eta^2(\ell+1)(2\ell+1)}{6\ell}\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2 \leq \eta^2\ell\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2$$
$$= \eta^2\ell\mathbb{E}_m\|V^m\|_F^2 . \ \square$$

Proof of Lemma E.4.2.2. We take the expectation of the Lyapunov function:

$$\mathbb{E}'_{\ell,m}\mathcal{L}^{m+1}(x_0^{m+1}) = \mathbb{E}_{\ell,m}f(x_0^{m+1}) + \frac{c_{m+1}}{n}\sum_{i=1}^{n}\mathbb{E}'_{\ell,m}\|x_0^{m+1}-\theta_i^{m+1}\|^2 . \tag{E.70}$$

Note that we here only analyze k-SVRG-V2 for which the samples to update the snapshot point are independent of the samples used to generate the sequence x_t^m. Also recall that $q=\ell$.

First we consider the second part of the Lyapunov function which is $\frac{c_{m+1}}{n}\sum_{i=1}^{n}\mathbb{E}'_{\ell,m}\|x_0^{m+1}-\theta_i^{m+1}\|^2$ and find its recurrence relation with $\frac{c_m}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2$.

$$\mathbb{E}'_{\ell,m}H^{m+1} = \sum_{i=1}^{n}\mathbb{E}'_{\ell,m}\|x_0^{m+1}-\theta_i^{m+1}\|^2$$
$$= \sum_{i=1}^{n}\left(\frac{\ell}{n}\mathbb{E}_{\ell,m}\|x_0^{m+1}-\tilde{x}^{m+1}\|^2 + \frac{n-\ell}{n}\mathbb{E}_{\ell,m}\|x_0^{m+1}-\theta_i^m\|^2\right)$$
$$= \ell\mathbb{E}_{\ell,m}\|x_0^{m+1}-\tilde{x}^{m+1}\|^2 + \left(1-\frac{\ell}{n}\right)\sum_{i=1}^{n}\mathbb{E}_{\ell,m}\|x_0^{m+1}-\theta_i^m\|^2 . \tag{E.71}$$

From Lemma E.10.0.3, we know that:

$$\mathbb{E}_{\ell,m}\|x^{m+1}-\tilde{x}^{m+1}\|^2 \leq \frac{\eta^2(\ell+1)(2\ell+1)}{6\ell}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|v_t^m\|^2 \leq \eta^2\ell\mathbb{E}_m\|V^m\|_F^2 . \tag{E.72}$$

We consider now the second term in (E.71), keeping in mind that $x_0^{m+1} = x_\ell^m$:

$$
\begin{aligned}
&\mathbb{E}_{\ell,m}\|x_0^{m+1} - \theta_i^m\|^2| = \mathbb{E}_{\ell,m}\|x_0^{m+1} - x_0^m + x_0^m - \theta_i^m\|^2 \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2 + \|x_0^m - \theta_i^m\|^2 - 2\langle x_0^m - x_0^{m+1}, x_0^m - \theta_i^m\rangle\right] \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2 + \|x_0^m - \theta_i^m\|^2 - 2\sum_{t=0}^{\ell-1}\langle x_t^m - x_{t+1}^m, x_0^m - \theta_i^m\rangle\right] \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 - 2\mathbb{E}_{\ell,m}\left[\left\langle\sum_{t=0}^{\ell-1}(x_t^m - x_{t+1}^m), x_0^m - \theta_i^m\right\rangle\right] \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 - 2\left[\left\langle\sum_{t=0}^{\ell-1}\mathbb{E}_{\ell,m}[x_t^m - x_{t+1}^m], x_0^m - \theta_i^m\right\rangle\right] \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 - 2\left[\left\langle\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}[x_t^m - x_{t+1}^m], x_0^m - \theta_i^m\right\rangle\right] \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 - 2\left[\left\langle\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\mathbb{E}_{\{t\}}[x_t^m - x_{t+1}^m], x_0^m - \theta_i^m\right\rangle\right] \\
&= \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 - 2\eta\left[\left\langle\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\nabla f(x_t^m), x_0^m - \theta_i^m\right\rangle\right] \\
&\overset{(E.38)}{=} \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 \\
&\qquad + 2\eta\left[\frac{1}{2\gamma}\sum_{t=0}^{\ell-1}\|\mathbb{E}_{t,m}\nabla f(x_t^m)\|^2 + \frac{\gamma\ell}{2}\|x_0^m - \theta_i^m\|^2\right] \qquad (\gamma > 0) \\
&\le \mathbb{E}_{\ell,m}\left[\|x_0^{m+1} - x_0^m\|^2\right] + \|x_0^m - \theta_i^m\|^2 \\
&\qquad + 2\eta\left[\frac{1}{2\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2 + \frac{\gamma\ell}{2}\|x_0^m - \theta_i^m\|^2\right] \\
&= \mathbb{E}_{\ell,m}\|x_0^{m+1} - x_0^m\|^2 + (1 + \gamma\eta\ell)\|x_0^m - \theta_i^m\|^2 + \frac{\eta}{\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2 \\
&= \eta^2\ell\sum_{t=0}^{\ell-1}\mathbb{E}_{\ell,m}\|v_t^m\|^2 + (1 + \gamma\eta\ell)\|x_0^m - \theta_i^m\|^2 + \frac{\eta}{\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2 \\
&= \eta^2\ell\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2 + (1 + \gamma\eta\ell)\|x_0^m - \theta_i^m\|^2 + \frac{\eta}{\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2. \qquad \text{(E.73)}
\end{aligned}
$$

Combining equation (E.73) and Lemma E.10.0.3, we get:

$$
\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}'_{\ell,m}\|x_0^{m+1} - \theta_i^{m+1}\|^2 \le \frac{\eta^2(\ell+1)(2\ell+1)}{6n}\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2 + \left(1 - \frac{\ell}{n}\right)\eta^2\ell\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2
$$

$$
\begin{aligned}
&+\left(1-\frac{\ell}{n}\right)\left[\frac{(1+\gamma\eta\ell)}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2+\frac{\eta}{\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\,\|\nabla f(x_t^m)\|^2\right]\\
&=\eta^2\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2\left(\frac{(\ell+1)(2\ell+1)}{6n}+\frac{\ell(n-\ell)}{n}\right)\\
&\quad+\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2+(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)\frac{1}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2\\
&\overset{(E.67)}{\leq}\eta^2\ell\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2+\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2\\
&\qquad\qquad+(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)\frac{1}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2\\
&=\eta^2\ell\mathbb{E}_{\ell,m}\|V^m\|_F^2+\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2\\
&\qquad\qquad+(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)\frac{1}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2\,. \qquad \text{(E.74)}
\end{aligned}
$$

Hence, we have:

$$
\mathbb{E}'_{\ell,m}H^{m+1}\leq\eta^2\ell\mathbb{E}_{\ell,m}\|V^m\|_F^2+\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2+(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)H^m\,.\square
$$

Proof of Lemma E.4.2.3. From Lemma E.10.0.2, we have:

$$
\mathbb{E}_{t+1,m}\|v_t^m\|^2\leq 2\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2+4L^2\eta^2t\sum_{j=0}^{t-1}\mathbb{E}_{j+1,m}\|v_j^m\|^2+\frac{4L^2}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2\,. \qquad \text{(E.75)}
$$

We sum the equation (E.75) for $t=0$ to $t=\ell-1$ to get the following:

$$
\begin{aligned}
\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2&\leq 2\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\left\|\nabla f(x_t^m)\right\|^2+4L^2\eta^2\sum_{t=0}^{\ell-1}t\sum_{j=0}^{t-1}\mathbb{E}_{j+1,m}\|v_j^m\|^2\\
&\qquad+\frac{4L^2l}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2\\
&\leq 2\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2+2L^2\eta^2\ell(\ell-1)\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2+\frac{4L^2\ell}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2\\
&\leq 2\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2+2L^2\eta^2\ell^2\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2+\frac{4L^2\ell}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2\,. \qquad \text{(E.76)}
\end{aligned}
$$

Since, $\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2 = \mathbb{E}_m\|V^m\|_F^2$, we get the following relation:

$$\begin{aligned}(1-2L^2\eta^2\ell^2)\mathbb{E}_{\ell,m}\|V^m\|_F^2 &\leq 2\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 + \frac{4L^2\ell}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2 \\ &\leq 2\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 + 4L^2\ell H^m\,. \end{aligned} \tag{E.77}$$

Hence, finally we have:

$$(1-2L^2\eta^2\ell^2)\mathbb{E}_m\|V^m\|_F^2 \leq \mathbb{E}_m\|\nabla F^m\|_F^2 + 4L^2\ell H^m\,. \tag{E.78}$$

Remark E.10.0.1. *Unfortunately, equation (E.78) limits us to choose ℓ as large as we would like (e.g. $\ell = n$ in case $k=1$), as otherwise the term $(1-2L^2\eta^2\ell^2)$ would become too small. In the proof of Theorem E.4.2.1 we will choose $\eta = \mathcal{O}(\frac{1}{Ln^{2/3}})$ and hence ℓ should be less than of the order of $\mathcal{O}(n^{2/3})$.*

□

Proof of Lemma E.4.2.4 The Lyapunov function is of the form:

$$\mathbb{E}'_{\ell,m}\mathcal{L}^m(x_0^{m+1}) = \mathbb{E}_{\ell,m}f(x_0^{m+1}) + \frac{c_{m+1}}{n}\sum_{i=1}^{n}\mathbb{E}'_{\ell,m}\|x_0^{m+1}-\theta_i^m\|^2\,. \tag{E.79}$$

First we analyze the term $\mathbb{E}_{\ell,m}f(x_0^{m+1})$ in the Lyapunov function. By the smoothness assumption:

$$f(x_{t+1}^m) \leq f(x_t^m) + \left\langle \nabla f(x_t^m), x_t^m - x_{t+1}^m\right\rangle + \frac{L}{2}\|x_t^m - x_{t+1}^m\|^2\,. \tag{E.80}$$

Now if we take expectation conditioned on x_t^m, we get:

$$\mathbb{E}_{\{i_t\}}f(x_{t+1}^m) \leq f(x_t^m) - \eta\|\nabla f(x_t^m)\|^2 + \frac{\eta^2 L}{2}\mathbb{E}_{\{i_t\}}\|v_t^m\|^2\,. \tag{E.81}$$

In equation (E.81), we apply the property of tower of conditional expectations and sum equation (E.81) from $t=0$ to $t=\ell-1$ in the m^{th} outer loop to get the following:

$$\mathbb{E}_{\ell,m}f(x_\ell^m) \leq f(x_0^m) - \eta\sum_{t=0}^{\ell-1}\mathbb{E}_{t,m}\|\nabla f(x_t^m)\|^2 + \frac{\eta^2 L}{2}\sum_{t=0}^{\ell-1}\mathbb{E}_{t+1,m}\|v_t^m\|^2\,. \tag{E.82}$$

Hence, we have:

$$\mathbb{E}_m f(x_0^{m+1}) \leq f(x_0^m) - \eta\mathbb{E}_m\|\nabla F^m\|_F^2 + \frac{\eta^2 L}{2}\mathbb{E}_m\|V^m\|_F^2\,. \tag{E.83}$$

We now analyze the complete Lyapunov function by using the results from Lemmas E.4.2.2

and E.4.2.3.

$$
\begin{aligned}
\mathbb{E}'_{\ell,m}\mathcal{L}^m(x_0^{m+1}) &= \mathbb{E}_{\ell,m}f(x_0^{m+1}) + \frac{c_{m+1}}{n}\sum_{i=1}^{n}\mathbb{E}'_{\ell,m}\|x_0^{m+1}-\theta_i^m\|^2 \\
&\overset{(E.83)}{\leq} f(x_0^m) - \eta\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 + \frac{\eta^2 L}{2}\mathbb{E}_{\ell,m}\|V^m\|_F^2 + \frac{c_{m+1}}{n}\sum_{i=1}^{n}\mathbb{E}_{\ell,m}\|x_0^{m+1}-\theta_i^m\|^2 \\
&\overset{(E.26)}{\leq} f(x_0^m) - \eta\mathbb{E}_m\|\nabla F^m\|_F^2 + \frac{\eta^2 L}{2}\mathbb{E}_m\|V^m\|_F^2 \\
&\quad + c^{m+1}\eta^2\ell\mathbb{E}_m\|V^m\|_F^2 + c^{m+1}\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\mathbb{E}_m\|\nabla F^m\|_F^2 \\
&\quad + c^{m+1}(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)\frac{1}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2 \\
&= f(x_0^m) - \left(\eta - c^{m+1}\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\right)\mathbb{E}_m\|\nabla F^m\|_F^2 + \left(\frac{\eta^2 L}{2} + c^{m+1}\eta^2\ell\right)\mathbb{E}_m\|V^m\|_F^2 \\
&\quad + \left[c^{m+1}(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)\right]\frac{1}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2. \qquad \text{(E.84)}
\end{aligned}
$$

Let $b_1 := \frac{1}{1-2L^2\eta^2\ell^2}$, as in the main text. Hence from Lemma E.4.2.3, we get:

$$
\begin{aligned}
\mathbb{E}'_{\ell,m}\mathcal{L}^m(x_0^{m+1}) &\leq f(x_0^m) - \left(\eta - c^{m+1}\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma}\right)\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 \\
&\quad + \left(\frac{\eta^2 L}{2} + c^{m+1}\eta^2\ell\right)\left[2b_1\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 + \frac{4b_1L^2\ell}{n}\sum_{i=i}^{n}\|x_0^m-\theta_i^m\|^2\right] \\
&\quad + \left(c^{m+1}(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right)\right)\frac{1}{n}\sum_{i=0}^{n}\|x_0^m-\theta_i^m\|^2 \\
&\leq f(x_0^m) - \left(\eta - c^{m+1}\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma} - b_1\eta^2 L - 2b_1c^{m+1}\eta^2\ell\right)\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 \\
&\quad + \left[c^{m+1}(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right) + 2b_1\eta^2L^3\ell + 4b_1c^{m+1}\eta^2L^2\ell^2\right]\frac{1}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2 \\
&\leq f(x_0^m) - \left(\eta - c^{m+1}\left(1-\frac{\ell}{n}\right)\frac{\eta}{\gamma} - b_1\eta^2 L - 2b_1c^{m+1}\eta^2\ell\right)\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2 \\
&\quad + \left[c^{m+1}(1+\gamma\eta\ell)\left(1-\frac{\ell}{n}\right) + 2b_1\eta^2L^3\ell + 4b_1c^{m+1}\eta^2L^2\ell^2\right]\frac{1}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2 \\
&\leq f(x_0^m) - \underbrace{\left(\eta - c^{m+1}\frac{\eta}{\gamma} - b_1\eta^2 L - 2b_1c^{m+1}\eta^2\ell\right)}_{=\Gamma^m}\mathbb{E}_{\ell,m}\|\nabla F^m\|_F^2
\end{aligned}
$$

$$+\underbrace{\left[c^{m+1}\left(1-\frac{\ell}{n}+\gamma\eta\ell+4b_1\eta^2L^2\ell^2\right)+2b_1\eta^2L^3\ell\right]}_{=c^m}\frac{1}{n}\sum_{i=1}^{n}\|x_0^m-\theta_i^m\|^2\,. \tag{E.85}$$

We finally get:

$$\Gamma^m\mathbb{E}_{\ell,m}\|\nabla F^m\|^2\leq\mathcal{L}^m(x_0^m)-\mathbb{E}'_{\ell,m}\mathcal{L}^{m+1}(x_0^{m+1})\,, \tag{E.86}$$

and the claim follows.

□

Proof of Theorem E.4.2.1. We add equation (E.28) from Lemma E.4.2.4 for $m=0$ to $m=M-1$ and take expecation with respect to the joint distribution of all the selection so far which gives:

$$\sum_{m=0}^{M-1}\Gamma^m\mathbb{E}\|\nabla F^m\|^2\leq\mathcal{L}^0(x_0^0)-\mathbb{E}\mathcal{L}^M(x_0^M,H^M)\,. \tag{E.87}$$

Since $\Gamma=\min_{0\leq m\leq M-1}$, we get

$$\Gamma\sum_{i=0}^{M-1}\mathbb{E}\|\nabla F^m\|^2\leq\mathcal{L}^0(x_0^0)-\mathbb{E}\mathcal{L}^M(x_0^M)\,, \tag{E.88}$$

from (E.87) and the first part of the theorem follows. To show the second part we need to derive a lower bound on Γ for the given parameters $\eta=\frac{1}{5Ln^{2/3}}$, $\gamma=\frac{L}{n^{1/3}}$ and $\ell=\frac{3}{2}n^{1/3}$. Observe that for these parameters $b_1\leq 2$.

First, let us derive an upper bound on c^m. Let $\lambda:=\ell\left(\frac{1}{n}-\gamma\eta-4b_1\eta^2L^2\ell\right)$. We have $\frac{8\ell}{25n}\leq\lambda\leq 1$, where the upper bound is immediate and the lower bound follows from $\left(\frac{1}{n}-\gamma\eta-4b_1\eta^2L^2\ell\right)\geq\left(\frac{1}{n}-\frac{1}{5n}-\frac{12}{25n}\right)=\frac{8}{25n}$, using $b_1\leq 2$. Observe that we have $c^m=c^{m+1}(1-\lambda)+2b_1\eta^2L^3\ell$. Using this relationship and $c^M=0$, it is easy to see that

$$c^m=2b_1\eta^2L^3\ell\frac{1-(1-\lambda)^{M-m}}{\lambda}\leq\frac{2b_1\eta^2L^3\ell}{\lambda}\leq\frac{L}{2n^{1/3}}\,, \tag{E.89}$$

for all $m=0,\ldots,M$. Now we are ready to derive a lower bound on Γ^m.

Using $\frac{\eta}{\gamma}=\frac{1}{5L^2n^{1/3}}$ and $c^m\leq\frac{L}{2n^{1/3}}$, we get:

$$\begin{aligned}\Gamma^m&\geq\frac{1}{5Ln^{2/3}}-\frac{1}{10Ln^{2/3}}-b_1\eta^2L-2b_1c^{m+1}\eta^2\ell\\&=\frac{1}{10Ln^{2/3}}-\underbrace{\left(b_1\eta^2L+2b_1c^{m+1}\eta^2\ell\right)}_{=:g_1}\end{aligned} \tag{E.90}$$

Now we consider the term g_1. As $b_1 \leq 2$ we have

$$\begin{aligned} g_1 &\leq 2\eta^2 L + 4c^{m+1}\eta^2 \ell \\ &\leq \frac{2}{25Ln^{4/3}} + \frac{3}{25Ln^{4/3}} \leq \frac{1}{30Ln^{2/3}}, . \end{aligned} \tag{E.91}$$

where the last inequality is due to $n > 15$. By combining (E.90) and (E.91) we get $\Gamma_m \geq \frac{1}{15Ln^{2/3}}$ for $m = 0, \ldots, M$. Hence, $\Gamma \geq \frac{1}{15Ln^{2/3}}$. □

E.11 Additional Experimental Results

E.11.1 Illustrative Experiment with more k-SVRG variants

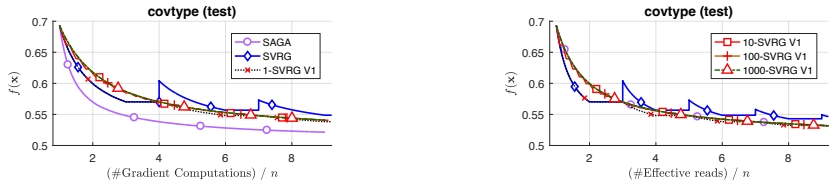


Figure E.4: Illustrating the different convergence behavior of SAGA, SVRG and k-SVRG-V1 for $k = \{1, 10, 100, 1000\}$.

E.11.2 Dataset: *covtype (test)*

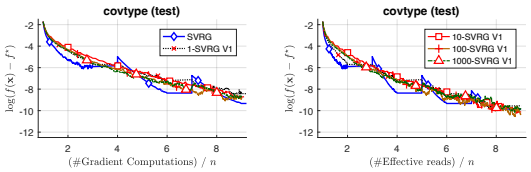


Figure E.5: Evolution of residual loss on *covtype (test)* for SVRG and k-SVRG-V1 for $k = \{1, 10, 100, 1000\}$.

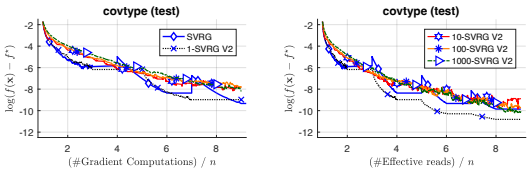


Figure E.6: Evolution of residual loss on *covtype (test)* for SVRG and k-SVRG-V2 for $k = \{1, 10, 100, 1000\}$.

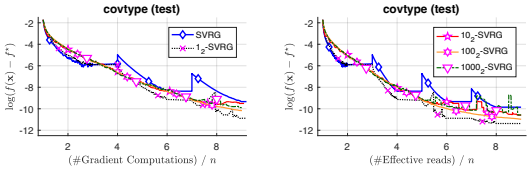


Figure E.7: Evolution of residual loss on *covtype (test)* for SVRG and k_2-SVRG for $k = \{1, 10, 100, 1000\}$.

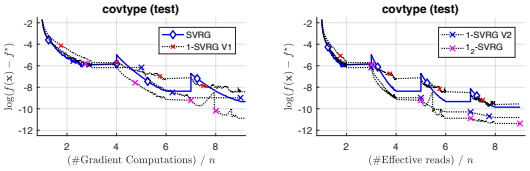


Figure E.8: Evolution of residual loss on *covtype (test)* for SVRG, 1-SVRG-V1, 1-SVRG-V2 and 1_2-SVRG.

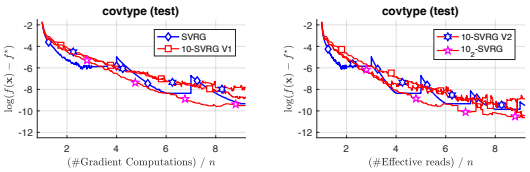


Figure E.9: Evolution of residual loss on *covtype (test)* for SVRG, 10-SVRG-V1, 10-SVRG-V2 and 10_2-SVRG.

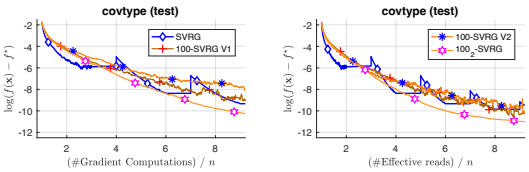


Figure E.10: Evolution of residual loss on *covtype (test)* for SVRG, 100-SVRG-V1, 100-SVRG-V2 and 100_2-SVRG.

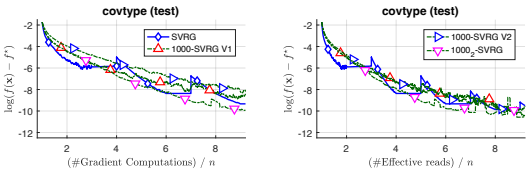


Figure E.11: Evolution of residual loss on *covtype (test)* for SVRG, 1000-SVRG-V1, 1000-SVRG-V2 and 1000_2-SVRG.

E.11.3 Dataset: *mnist*

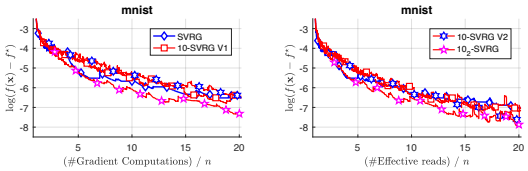


Figure E.12: Evolution of residual loss on *mnist* for SVRG, 10-SVRG-V1, 10-SVRG-V2 and 10_2-SVRG.

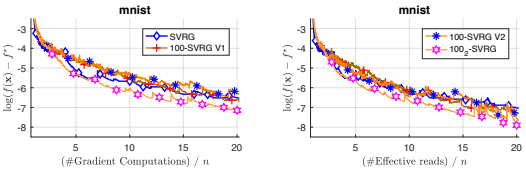


Figure E.13: Evolution of residual loss on *mnist* for SVRG, 100-SVRG-V1, 100-SVRG-V2 and 100_2-SVRG.

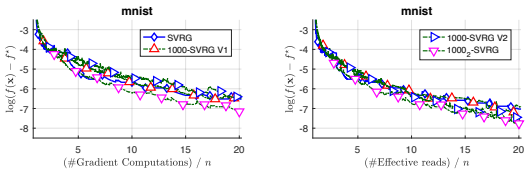


Figure E.14: Evolution of residual loss on *mnist* for SVRG, 1000-SVRG-V1, 1000-SVRG-V2 and 1000_2-SVRG.

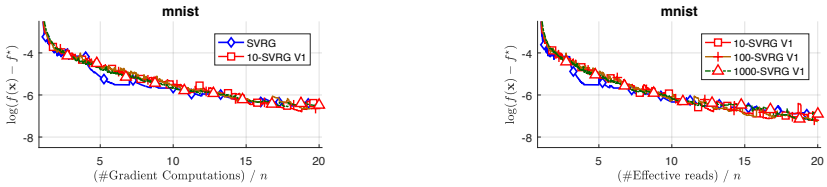


Figure E.15: Evolution of residual loss on *mnist* for SVRG and k-SVRG-V1 for $k = \{10, 100, 1000\}$.

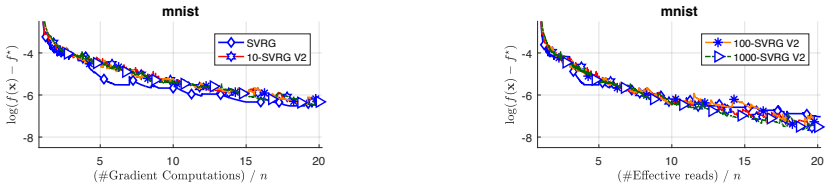


Figure E.16: Evolution of residual loss on *mnist* for SVRG and k-SVRG-V2 for $k = \{10, 100, 1000\}$.

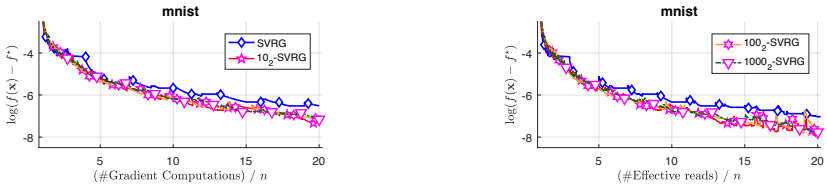


Figure E.17: Evolution of residual loss on *mnist* for SVRG and k_2-SVRG for $k = \{10, 100, 1000\}$.

E.11.4 Dataset: *covtype (train)*

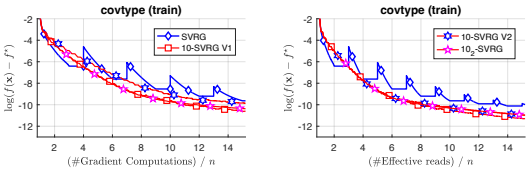


Figure E.18: Evolution of residual loss on *covtype (train)* for SVRG, 10-SVRG-V1, 10-SVRG-V2 and 10_2-SVRG.

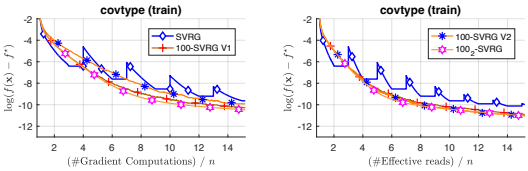


Figure E.19: Evolution of residual loss on *covtype (train)* for SVRG, 100-SVRG-V1, 100-SVRG-V2 and 100_2-SVRG.

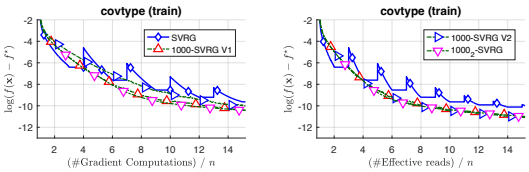


Figure E.20: Evolution of residual loss on *covtype (train)* for SVRG, 1000-SVRG-V1, 1000-SVRG-V2 and 1000_2-SVRG.

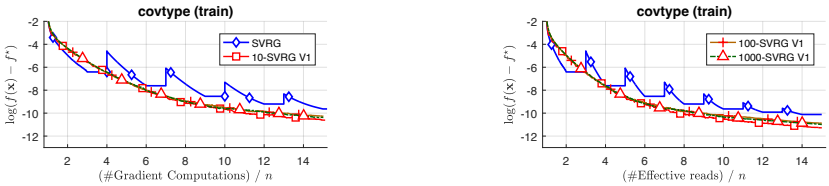


Figure E.21: Evolution of residual loss on *covtype (train)* for SVRG and *k*-SVRG-V1 for $k = \{10, 100, 1000\}$.

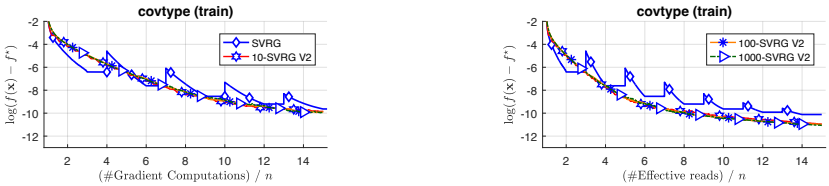


Figure E.22: Evolution of residual loss on *covtype (train)* for SVRG and *k*-SVRG-V2 for $k = \{10, 100, 1000\}$.

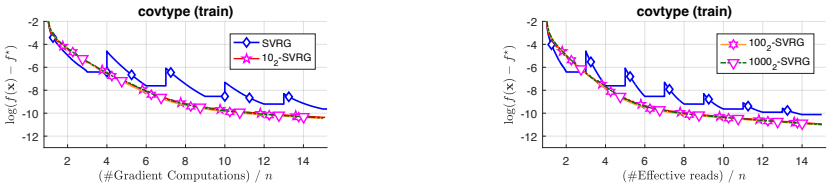


Figure E.23: Evolution of residual loss on *covtype (train)* for SVRG and k_2-SVRG for $k = \{10, 100, 1000\}$.

Appendix F

A Simpler Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism

Pooria Joulani[*1], Anant Raj[*2], András György[1], Csaba Szepesvári[1]

1 – Google Deepmind, London

2 – MPI for Intelligent Systems, Tübingen

* – denotes Equal Contribution

Abstract

Recently there have been several attempts to extend Nesterov's accelerated algorithm to smooth stochastic and variance-reduced optimization. In this paper, we show that there is a simpler approach to acceleration: applying *optimistic* online learning algorithms and querying the gradient oracle at the *online average* of the intermediate optimization iterates. In particular, we tighten a recent result of Cutkosky [50] to demonstrate theoretically that online iterate averaging results in a reduced optimization gap, independently of the algorithm involved. We show that carefully combining this technique with existing generic optimistic online learning algorithms yields the optimal accelerated rates for optimizing strongly-convex and non-strongly-convex, possibly composite objectives, with deterministic as well as stochastic first-order oracles. We further extend this idea to variance-reduced optimization. Finally, we also provide "universal" algorithms that achieve the optimal rate for smooth and non-smooth composite objectives simultaneously without further tuning, generalizing the results of Kavis *et al.* [111] and solving a number of their open problems.

F.1 Introduction

Our goal in this paper is to obtain algorithms with optimal convergence rates for the following problem:

$$\text{find } x^\star = \arg\min_{x\in\mathcal{X}} \ell(x) = f(x) + \phi(x)\,, \tag{F.1}$$

where $\mathcal{X}$ is a convex constraint set in the d-dimensional Euclidean space, f is convex and smooth, and ϕ is a (possibly non-smooth) convex function. When $\phi = 0$, and given access to (noise-free) gradients of f, Nesterov's accelerated gradient algorithms [175]achieve optimal rates of convergence for Problem (F.1). Several recent papers, summarized in table F.1, have attempted to obtain similarly accelerated rates that improve upon the sub-optimal rates of Stochastic Gradient Descent (SGD) when the gradients of f are corrupted by noise and/or when $\phi \neq 0$.

Despite the major effort to obtain these extensions, existing results suffer from several limitations such as: (a) inhibiting noise in the gradient [9, 258]; (b) potentially querying the gradient oracle outside the constraint set [50, 134] (c) not providing optimal rates for strongly-convex objectives [50]; (d) extra logarithmic terms appearing in the error bounds [50, 134]; (e) not handling proximal updates when $\phi \neq 0$ [50, 111, 134] or (f) relying on prior knowledge of problem parameters [24, 42, 94, 123, 250, 266].

In this paper, we demonstrate a simple direct approach to deriving accelerated rates: following Cutkosky [50], we propose running an online learning algorithm and feeding it with (possibly noisy) first-order information obtained at the *weighted average* of its iterates. Then, building on the recent simple, tight modular analysis techniques of generic optimistic online learning algorithms [102, 103], we are able to alleviate all the aforementioned limitations, design new accelerated algorithms with straightforward convergence analyses, and solve a number of problems left open in previous work.

F.1.1 Contributions and Related Work

Our main contributions can be summarized as follows: 1. We provide a direct, simple template for deriving and analyzing accelerated algorithms for stochastic and deterministic convex optimization with composite objectives. We further extend the above framework to variance-reduced stochastic non-strongly-convex optimization.
2. For composite non-strongly-convex objectives, we provide a new *universal* algorithm (in the sense of 174): given only access to the proximal projection oracle of ϕ onto the constraint set, without prior knowledge of the smoothness or noise level, the new algorithm simultaneously achieves the optimal rate of convergence for smooth and non-smooth f. This, together with the fact that the algorithm uses coordinate-wise adaptive step-sizes, resolves two problems left open by Kavis *et al.* [111].

In particular, in theorems F.3.0.1 and F.3.0.2, we tighten the recent analysis of online iterate averaging by Cutkosky [50]. Compared to their Theorem 1, theorem F.3.0.2

	$\mathcal{X}$	f	ϕ	Oracle	Universal	Notes
Tseng [250]	Any	Non-SC	✓	D	-	
Beck and Teboulle [24]	$\mathbb{R}^d$	Non-SC	✓	D	-	
Hu *et al.* [94]	$\mathbb{R}^d$	SC / Non-SC	✓	S+D	-	Assumes bounded trajectory
Xiao [266]	Any	Non-SC	✓	S + D	-	Dual-Averaging
Lan [123]	Any	Non-SC	✓	S+D	-	Not utilizing prox-map of ϕ
Chen *et al.* [42]	Any Any	SC / Non-SC SC	✓ ✓	S+D S+D	- -	 Exponential noise-free rate
Allen-Zhu and Orecchia [9]	Any Any	Non-SC SC	- -	D D	- -	Linear-coupling Exponential rate
Wang and Abernethy [258]	Any Any	Non-SC SC	✓ -	D D	- -	Primal-dual view Exponential rate
This paper (theorem F.4.1.1)	**Any** **Any**	**SC / Non-SC** **SC**	✓ ✓	**S + D** **D**	**-** **-**	 **Exponential rate**
Cutkosky [50]	Compact	Non-SC	-	S+D	✓	Accessing f outside $\mathcal{X}$
Levy *et al.* [134]	Compact	Non-SC	-	S+D	✓	Accessing f outside $\mathcal{X}$
Kavis *et al.* [111]	Compact	Non-SC	-	S + D	✓	
This paper (theorem F.4.2.1)	**Compact**	**Non-SC**	✓	**S + D**	✓	

Table F.1: Summary of previous work obtaining accelerated rates of convergence. Cutkosky [50] analyses strongly-convex optimization as well, but the rates are sub-optimal (i.e., non-accelerated). Here, "Non-SC" means non-strongly convex (that is, strong-convexity of f is not required), "SC" means strongly convex, "D" and "S" stand for deterministic and stochastic oracles, respectively. Universality means the algorithm achieves the smooth and non-smooth rates simultaneously without requiring the knowledge of the problem's smoothness and noise level. The "bounded-trajectory" assumption means that the error bound scales with the maximum distance of the iterates from the optimum $x^\star$, but the algorithm does not enforce this to be bounded (e.g., through projection to a compact set). See also the survey by Bubeck [35] and the recent book of Nesterov [175].

exposes additional terms that reduce the optimization gap. These terms, whose absence prevented Cutkosky [50, Theorem 3] from getting the optimal accelerated rates, are similar to what Wang and Abernethy [258] obtained through an indirect formulation of acceleration as a two-player game.

Next, we show how to utilize the aforementioned reduction to obtain accelerated rates. This is achieved by using properly-tuned *optimistic online learning* algorithms [157, 195, 196] as the underlying optimization machinery. Importantly, this tuning can be done somewhat independently of the assumptions on the objective (such as the presence of noise, strong convexity, or a non-zero ϕ) or the algorithmic techniques (such as proximal updates or adaptive learning rates), thanks to the recent modular analyses of online learning algorithms by Joulani *et al.* [102, 103]. This results in a simple, straightforward acceleration framework.

Furthermore, we extend the analysis to variance-reduced optimization for smooth non-strongly-convex functions. We show that incorporating negative momentum (common in the accelerated SVRG literature, see, e.g., 6, 125) in our framework introduces an additional reduction in the optimization gap, enabling us to obtain the optimal convergence rate. We also analyze a simpler version of the variance reduced algorithm without negative momentum, which enjoys a variance-reduced, though still sub-optimal rate of convergence

for the last iterate.

Finally, we provide a universal algorithm for non-strongly-convex composite optimization, extending the works of Cutkosky [50], Kavis *et al.* [111], Levy *et al.* [134] to the case when $\phi \neq 0$. The new algorithm features proximal updates and coordinate-wise adaptive step-sizes, thus solving two problems left open by Kavis *et al.* [111]. Unlike Levy *et al.* [134] and Cutkosky [50], the new algorithm does not query the optimization oracle outside the constraint set $\mathcal{X}$, and does not suffer from extra log terms in the bound. Unlike the algorithm of Kavis *et al.* [111] (which is based on mirror-descent) our algorithm is based on dual-averaging, which is better suited to sparse learning with a proximal ℓ_1 penalty [152, 265].

Notation. $\mathbb{R}$ denotes the set of real numbers. For any positive integer n, $[n] = \{1, \dots, n\}$. Let $h : \mathcal{D} \to \mathbb{R}$ where $\mathcal{D} \subset \mathbb{R}^d$ for some positive integer d. The gradient or a subgradient of h is denoted by h'. When h is convex, the Bregman-divergence $B_h : \mathcal{D} \times \mathcal{D}^o \to \mathbb{R}$ is defined as $B_h(x,y) = h(x) - h(y) - \langle h'(y), x - y \rangle$, where $\mathcal{D}^o$ denotes the interior of $\mathcal{D}$. We say that h is μ-strongly convex with respect to (w.r.t.) a norm $\|\cdot\|$ if for all $x \in \mathcal{D}, y \in \mathcal{D}^o$, $\frac{\mu}{2}\|x-y\|^2 \leq B_h(x,y)$, and it is μ-strongly convex w.r.t. a function $n : \mathcal{D} \times \mathcal{D}^o \to [0,\infty)$ if $\mu \cdot n(x,y) \leq B_h(x,y)$ for all $x \in \mathcal{D}, y \in \mathcal{D}^o$ (note that h is μ-strongly convex w.r.t. a norm $\|\cdot\|$ if it is μ-strongly convex w.r.t. the function $\|\cdot\|^2/2$). For non-negative integers a, b and a sequence of numbers or vectors $x_0, x_1, \dots$, we let $x_{a:b} = \sum_{s=a}^{b} x_s$ if $a \leq b$ and 0 otherwise. With a slight abuse of notation, for a vector $x \in \mathbb{R}^d$, we denote its coordinates as $x = (x_1, \dots, x_d)$; whether the subscript refers to a coordinate or a time index is usually clear from the context (to reduce the possible ambiguity, we normally use x_i and x_j to index coordinates of x, and x_t and x_s to indicate a quantity corresponding to time steps t or s). For an event E, $\mathbb{I}\{E\}$ denotes its indicator function, that is $\mathbb{I}\{E\} = 1$ if E is true, otherwise $\mathbb{I}\{E\} = 0$. The base-2 logarithm of $x \in (0, +\infty)$ is denoted by $\log(x)$.

F.2 Preliminaries

For simplicity, we assume that an optimizer $x^\star \in \mathcal{X}$ of Problem (F.1) exists, i.e., $\ell^\star := \ell(x^\star) \leq \ell(x)$ for all $x \in \mathcal{X}$.[1]

Smoothness of functions. When f is differentiable over $\mathbb{R}^d$, given a norm $\|\cdot\|$, the following are equivalent definitions of smoothness of f [175, Theorem 2.1.5]: f is L-smooth if

1. for all $x, y \in \mathbb{R}^d$, $B_f(x,y) \leq \frac{L}{2}\|x-y\|^2$;
2. for all $x, y \in \mathbb{R}^d$, $\|f'(x) - f'(y)\|_* \leq L\|x-y\|$;

[1] We do not require $\mathcal{X}$ to be closed or compact, which are normally assumed to ensure $x^\star$ exists.

3. for all $x, y \in \mathbb{R}^d$,

$$\|f'(x) - f'(y)\|_*^2 \leq (2L) B_f(x, y)\,. \tag{F.2}$$

Throughout the paper, we only require[2] that f is differentiable over $\mathcal{X}$, and use (F.2), holding for all $x, y \in \mathcal{X}$, as the notion of smoothness under which accelerated rates are obtained. Alternatively, assuming smoothness assumption holds only for $x, y \in \mathcal{X}$, one can still obtain the same rates with a very similar analysis as we provide, at the expense of an additional gradient oracle call per step of the algorithms; we leave the details for an extended version of the paper.

Iterative optimization. We consider first-order sequential optimization procedures with access to a stochastic gradient oracle that returns unbiased estimates of f'. A sequential optimization method then, in iteration t, queries the oracle at a point $y_t \in \mathcal{X}$, receives a gradient estimate g_t such that $\mathbb{E} g_t | \mathcal{H}_t = f'(y_t)$ where $\mathcal{H}_t = \sigma\left((g_s)_{s=1}^{t-1}, (y_s)_{s=1}^{t}\right)$ is the sigma-algebra generated by all the information used by the algorithm before making the query at y_t to the gradient oracle. In case $\phi \neq 0$, we also assume that the optimization method has access to the prox-function of ϕ (cf. Eq. F.6). After T iterations, the algorithm produces an estimate $\bar{x}_T$ of $x^\star$, based on all the information it has seen, where the quality of the estimate is measured by the error $\mathbb{E}\ell(\bar{x}_T) - \ell^\star$.

Online linear optimization. One way to design and analyze iterative optimization methods is through *online linear optimization* (OLO) algorithms. An OLO algorithm sequentially comes up, at each time step $t \in [T]$, with a prediction x_t, then receives a linear loss function $\langle \alpha_t u_t, \cdot \rangle$, with the aim of maintaining a small cumulative composite loss $\sum_{t=1}^T \alpha_t \left(\langle u_t, x_t - x \rangle + \phi(x_t) - \phi(x)\right)$, a.k.a. its *regret* compared to a competitor point x. Here $u_t \in \mathbb{R}^d$ is unknown to the algorithm before selecting x_t, but the non-negative weights α_t are known ahead of time. One can convert an OLO algorithm to an iterative optimization algorithm by using $y_t = x_t$ to query the oracle, using $u_t = g_t$ in the linear loss to the OLO algorithm, and employing the average $\bar{x}_T = \sum_{t=1}^T \frac{\alpha_t}{\alpha_{1:T}} x_t$ as the final estimate of $x^\star$.

The appeal of this "vanilla online-to-batch" approach (algorithm 16), is that it reduces the convergence analysis of $\bar{x}_T$ for convex f and ϕ to the regret analysis of the underlying OLO algorithm. In particular, by Jensen's inequality,

$$\begin{aligned}
\mathbb{E}\ell(\bar{x}_T) - \ell^\star &\leq \sum_{t=1}^T \mathbb{E} \frac{\alpha_t \left(\langle f'(x_t), x_t - x^\star \rangle + \phi(x_t) - \phi(x^\star)\right)}{\alpha_{1:T}} \\
&= \mathbb{E} \frac{\sum_{t=1}^T \alpha_t \left(\langle g_t, x_t - x^\star \rangle + \phi(x_t) - \phi(x^\star)\right)}{\alpha_{1:T}} \leq \mathbb{E} \frac{\mathcal{R}_T(x^\star)}{\alpha_{1:T}},
\end{aligned} \tag{F.3}$$

[2] Extensions when f is non-differentiable at boundary points are straightforward.

Algorithm 16 Vanilla Online-to-Batch

Input: Stochastic gradient oracle, non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, online linear optimization algorithm $\mathcal{A}$
Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$
for $t = 1$ **to** $T-1$ **do**
  Get stochastic gradient g_t at the *current* iterate x_t
  Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
  Let x_{t+1} be the next iterate from $\mathcal{A}$
end for
return the average iterate $\frac{\sum_{t=1}^T \alpha_t x_t}{\alpha_{1:T}}$.

Algorithm 17 Anytime Online-to-Batch [50]

Input: Stochastic gradient oracle, non-negative weights $(\alpha_t)_{t=1}^T$ with $\alpha_1 > 0$, online linear optimization algorithm $\mathcal{A}$
Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$ and let $\bar{x}_1 \leftarrow x_1$
for $t = 1$ **to** $T-1$ **do**
  Get stochastic gradient g_t at the *average* iterate $\bar{x}_t$
  Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
  Let x_{t+1} be the next iterate from $\mathcal{A}$
  Let $\bar{x}_{t+1} \leftarrow \frac{\sum_{s=1}^{t+1} \alpha_s x_s}{\alpha_{1:t+1}}$
end for
return the average iterate $\bar{x}_T$

where $\mathcal{R}_T(x^\star)$ is an upper-bound for the regret of the OLO algorithm. Thus, to analyze the convergence of $\bar{x}_T$, one can simply plug-in an off-the-shelf regret bound (reviewed at the end of this section) for the underlying OLO algorithm.

Anytime online-to-batch. An alternative, elegant online-to-batch conversion (algorithm 17) was recently proposed by Cutkosky [50], which uses the "online" average $\bar{x}_t = \sum_{s=1}^t \frac{\alpha_s}{\alpha_{1:t}} x_s$ as the query point, i.e., $y_t = \bar{x}_t$. Cutkosky [50, Theorem 1] showed (with $\phi = 0$) that (F.3) holds under this conversion scheme as well. In the next section, we show that in fact algorithm 17 enjoys a tighter version of (F.3) that enables us to prove accelerated rates.

Generic regret bound. Next, we recall the regret bound for a general family of OLO algorithms known as "adaptive optimistic follow the regularized leader" or AO-FTRL

[157, 195, 196]. At time t, AO-FTRL makes its t-th prediction as

$$x_t = \operatorname*{arg\,min}_{x \in \mathcal{X}} \left\langle \sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t, x \right\rangle + \alpha_{1:t}\phi(x) + r_{0:t-1}(x)\,, \tag{F.4}$$

where, the $r_t : \mathcal{X} \to \mathbb{R}$ are convex *regularizer* functions, and for every t, $\tilde{g}_t$, the *optimistic* part of the update, is interpreted as a prediction of g_t before it is received.

It is straightforward to see that AO-FTRL captures a wide range of algorithms used in optimization [153, 265]. For example, the dual-averaging algorithm of Xiao [265] corresponds to the case when $\phi = 0$ and $r_{0:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$ for $\eta_t > 0$, in which case it is easy to verify that

$$x_t = \Pi_{\mathcal{X}}\left(-\frac{\sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t}{\eta_t} \right), \tag{F.5}$$

where $\Pi_{\mathcal{X}}$ denotes Euclidean projection onto set $\mathcal{X}$. More generally, allowing coordinatewise step sizes $\eta_t \in [0,\infty)^d$ and a possibly non-zero ϕ, with $r_{0:t-1}(x) = \frac{1}{2}\sum_{j=1}^d \eta_{t,j} x_j^2$ we recover the proximal (a.k.a. "composite-objective" or "regularized") dual-averaging update [265]:

$$\begin{aligned} x_t &= \mathbf{prox}_{\alpha_{1:t}\phi, \eta_t}\left(-\sum_{s=1}^{t-1} \alpha_s g_s - \alpha_t \tilde{g}_t \right) \\ &= \operatorname*{arg\,min}_{x \in \mathcal{X}} \ \alpha_{1:t}\phi(x) + \frac{1}{2}\sum_{j=1}^d \eta_{t,j}\left(x_j - \frac{z_{t-1,j}}{\eta_{t,j}} \right)^2, \end{aligned} \tag{F.6}$$

where $\mathbf{prox}_{\alpha_{1:t}\phi,\eta_t}$ is the prox-function of $\alpha_{1:t}\phi$ with coordinatewise step sizes $\eta_{t,j}$ and $z_{t-1} = -\left(\sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t\right)$. Note that AdaGrad-style updates [60] can be recovered by setting η_t based on the past gradient estimates $g_s, \tilde{g}_s$ (for $s < t$). If $r_t \geq 0$, the cumulative regularizer $\alpha_{1:t}\phi + r_{0:t-1}$ is 1-strongly convex w.r.t. a norm $\|\cdot\|_{(t)}$, and the AO-FTRL update is well-defined, that is, the minimizer $x_t \in \mathcal{X}$ exists and $\left\langle \sum_{s=1}^{t-1} \alpha_s g_s + \alpha_t \tilde{g}_t, x_t \right\rangle + \alpha_{1:t}\phi(x_t) + r_{0:t-1}(x_t)$ is finite, then Theorem 6 of Joulani *et al.* [103] gives the following regret bound (see appendix F.11):

$$\mathcal{R}_T(x^\star) = r_{0:T-1}(x^\star) + \sum_{t=1}^T \frac{1}{2}\alpha_t^2 \|g_t - \tilde{g}_t\|_{(t)*}^2. \tag{F.7}$$

F.3 Acceleration with Anytime Online-to-Batch

First, we present a lemma that generalizes the regret decomposition of Joulani *et al.* [102] to work with the averaging scheme of Cutkosky [50]. Crucially, the decomposition keeps

track of some negative Bregman-divergence terms, which are instrumental in reducing the contribution of the OLO regret to the error of $\bar{x}_T$.

Lemma F.3.0.1. *For $t = 1, 2, \dots T$, let $\alpha_t > 0$ and $x_t \in \mathbb{R}^d$, and define $\bar{x}_t = (\sum_{s=1}^t \alpha_s x_s)/\alpha_{1:t}$, $B_t = \alpha_t B_f(x^\star, \bar{x}_t)$, and $\bar{B}_t^f = \alpha_{1:t-1} B_f(\bar{x}_{t-1}, \bar{x}_t), t > 1$. Then, if ϕ is convex,*

$$\alpha_{1:T}\left(\ell(\bar{x}_T) - \ell^\star\right) \le \sum_{t=1}^T \alpha_t \left(\left\langle f'(\bar{x}_t), x_t - x^\star\right\rangle + \phi(x_t) - \phi(x^\star)\right) - B_{1:T} - \bar{B}_{2:T}^f . \tag{F.8}$$

The lemma immediately gives rise to the following generic error bound, which improves upon Theorem 1 of Cutkosky [50] by keeping around the aforementioned $-\bar{B}_t^f$ and $-B_t$ terms. While the B_t are the usual Bregman-divergence terms (also appearing in the vanilla online-to-batch) that are utilized to get fast rates for strongly convex functions (and can be dropped in general as long as the function is star-convex; see 103), the important new terms here are the $-\bar{B}_t^f$ terms, which allow us to prove accelerated rates for online averaging.

Corollary F.3.0.2 (Generic Error Bound)**.** *Under the assumptions of Lemma F.3.0.1, if for all $t = 1, 2, \dots, T$, $g_t \in \mathbb{R}^d$ satisfies $\mathbb{E} g_t | \bar{x}_t = f'(\bar{x}_t)$ and we have*

$$\sum_{t=1}^T \alpha_t \left(\langle g_t, x_t - x^\star\rangle + \phi(x_t) - \phi(x^\star)\right) \le \mathcal{R}_T(x^\star) \tag{F.9}$$

for some upper-bound $\mathcal{R}_T(x^\star)$, then

$$\mathbb{E}\ell(\bar{x}_T) - \ell(x^\star) \le \mathbb{E}\frac{\mathcal{R}_T(x^\star) - B_{1:T} - \bar{B}_{2:T}^f}{\alpha_{1:T}}. \tag{F.10}$$

The corollary follows since g_t is a conditionally unbiased estimate of $f'(\bar{x}_t)$, so the first term on the r.h.s. of (F.8) is, in expectation, equal to the term on the l.h.s. of (F.9), and hence upper-bounded by $\mathbb{E}\mathcal{R}_T(x^\star)$. Next, we prove the lemma.

Proof of Lemma F.3.0.1. Writing $f(\bar{x}_T)$ as a telescoping sum,

$$\begin{aligned}
f(\bar{x}_T) - f(x^\star) &= -f(x^\star) + \frac{\alpha_1 f(\bar{x}_1)}{\alpha_{1:T}} + \sum_{t=2}^T \frac{\alpha_{1:t} f(\bar{x}_t) - \alpha_{1:t-1} f(\bar{x}_{t-1})}{\alpha_{1:T}} \\
&= \sum_{t=1}^T \frac{\alpha_t \left(f(\bar{x}_t) - f(x^\star)\right)}{\alpha_{1:T}} + \sum_{t=2}^T \frac{\alpha_{1:t-1}\left(f(\bar{x}_t) - f(\bar{x}_{t-1})\right)}{\alpha_{1:T}} \\
&= \sum_{t=1}^T \frac{\alpha_t \left\langle f'(\bar{x}_t), \bar{x}_t - x^\star\right\rangle - B_t}{\alpha_{1:T}} + \sum_{t=2}^T \frac{\alpha_{1:t-1}\left\langle f'(\bar{x}_t), \bar{x}_t - \bar{x}_{t-1}\right\rangle - \bar{B}_t^f}{\alpha_{1:T}} \\
&= \sum_{t=1}^T \frac{\alpha_t \left\langle f'(\bar{x}_t), \bar{x}_t - x^\star\right\rangle - B_t}{\alpha_{1:T}} + \sum_{t=2}^T \frac{\alpha_t \left\langle f'(\bar{x}_t), x_t - \bar{x}_t\right\rangle - \bar{B}_t^f}{\alpha_{1:T}}
\end{aligned}$$

$$= \frac{\sum_{t=1}^{T} \alpha_t \langle f'(\bar{x}_t), x_t - x^\star \rangle - B_{1:T} - \bar{B}^f_{2:T}}{\alpha_{1:T}},$$

where the third step follows since by the definition of Bregman divergence, $f(z) - f(y) = \langle f'(z), z-y \rangle - B_f(y,z)$, the fourth step follows since by the definition of $\bar{x}_t$, for $t = 2,3,\ldots,T$ we have $\alpha_t(\bar{x}_t - x_t) = \alpha_{1:t-1}(\bar{x}_{t-1} - \bar{x}_t)$, and the last step uses $\bar{x}_1 = x_1$. The proof is completed by $\phi(\bar{x}_T) - \phi(x^\star) \leq \sum_{t=1}^{T} \frac{\alpha_t}{\alpha_{1:T}} (\phi(x_t) - \phi(x^\star))$, which holds by Jensen's inequality. □

Acceleration. The main idea behind deriving accelerated rates is combining (F.7) with (F.10), and selecting α_t and $\tilde{g}_t$ appropriately so that the negative terms $-\bar{B}^f_t$ in (F.10) offset the contribution of the terms $\frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*}$ in (F.7) to the final error bound of $\bar{x}_T$. For example, let f be L-smooth over $\mathbb{R}^d$ or assume otherwise that (F.2) holds with the norm $\|\cdot\| = \|\cdot\|_2$. Suppose the optimization algorithm uses the dual averaging update (F.5) with $\alpha_t = t$, $\eta_t = \eta = 2L$, deterministic gradients $g_t = f'(\bar{x}_t)$, and $\tilde{g}_t = g_{t-1}$. Then, $r_{0:t-1}$ is 1-strongly convex w.r.t. the norm $L\|\cdot\|_2^2$, and the norm terms $\frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*}$ in (F.7) can be bounded as

$$\frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*} = \alpha_t^2 \frac{1}{4L}\|f'(\bar{x}_t) - f'(\bar{x}_{t-1})\|_2^2 \leq \frac{\alpha_t^2}{2\alpha_{1:t-1}}\bar{B}^f_t = \frac{t^2}{t(t-1)}\bar{B}^f_t \leq \bar{B}^f_t,$$

where the first inequality follows using (F.2). Hence, $\mathcal{R}_T(x^\star) - \bar{B}^f_{2:T} \leq L\|x^\star\|_2^2 + \frac{1}{4L}\|f'(x_1)\|_2^2$. Noticing that $\alpha_{1:T} = \Omega(T^2)$ gives the well-known accelerated $\mathcal{O}(1/T^2)$ rate for the error of $\bar{x}_T$. The next theorem, proved in appendix F.7, makes this argument precise for the general setting with noise, non-zero ϕ and generic AO-FTRL.

Theorem F.3.0.3. *In algorithm 17, let the base method $\mathcal{A}$ generate its iterates by the AO-FTRL update (F.4), using $\tilde{g}_t = g_{t-1}$ as the optimistic prediction of g_t for $t > 1$ and arbitrary $\tilde{g}_1$. Suppose that f and ϕ are convex, and there exists a norm $\|\cdot\|$ such that either f is 1-smooth w.r.t. $\|\cdot\|$ over $\mathbb{R}^d$ or otherwise (F.2) holds with $L = 1$ for all $x, y \in \mathcal{X}$. Further suppose that for all $t \in [T]$, $r_{t-1} \geq 0$ is convex, the AO-FTRL update (F.4) is well-defined with finite value at the optimum x_t, and there exist $\beta_t > 0$ and a norm $\|\cdot\|_{(t)}$ such that $\alpha_{1:t}\phi + r_{0:t-1}$ is 1-strongly-convex w.r.t. $\frac{\beta_t}{2}\|\cdot\|^2 + \frac{1}{2}\|\cdot\|^2_{(t)}$. Then, if $\alpha_t^2\beta_t^{-1} \leq \alpha_{1:t-1}$ for all $t > 1$, we have*

$$\mathbb{E}\ell(\bar{x}_T) - \ell^\star \leq \sum_{t=1}^{T} \mathbb{E}\frac{r_{t-1}(x^\star) - r_{t-1}(x_t) - B_t}{\alpha_{1:T}} + \sum_{t=1}^{T} \mathbb{E}\frac{\alpha_t^2\|\sigma_t - \sigma_{t-1}\|^2_{(t)*}}{2\alpha_{1:T}} + \mathbb{E}\frac{\alpha_1^2\|f'(\bar{x}_1) - \tilde{g}_1\|_*^2}{2\beta_1\alpha_{1:T}}, \tag{F.11}$$

where $\sigma_t = g_t - f'(\bar{x}_t), t \in [T]$, and $\sigma_0 = 0$.

F.4 Applications

In this section we use the framework of the previous section, and Theorem F.3.0.3 in particular, to obtain accelerated convergence rates with proximal updates, noisy gradients, and universal algorithms.

F.4.1 Accelerated Proximal Dual-Averaging

First, we show that with appropriately setting α_t and η_t, one can obtain the optimal accelerated rates for the proximal dual averaging update (F.6). In particular, we consider the case of a single step size for all coordinates (with a slight abuse of notation, $\eta_{t,i} = \eta_t$ for all i) and $r_{0:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$. Then, under the conditions of theorem F.3.0.3, we have

$$\mathbb{E}\ell(\mathbf{x}_T) - \ell^\star \le \sum_{t=1}^{T} \mathbb{E}\frac{\alpha_t^2\|\sigma_t - \sigma_{t-1}\|_{(t)*}^2}{2\alpha_{1:T}} + \mathbb{E}\frac{\eta_T\|x^\star\|_2^2 - \sum_{t=1}^{T}(\eta_t - \eta_{t-1})\|x_t\|_2^2 - 2B_t}{2\alpha_{1:T}} + \mathbb{E}\frac{\alpha_1^2}{2\beta_1\alpha_{1:T}}\|f'(\mathbf{x}_1) - \tilde{g}_1\|_*^2. \tag{F.12}$$

Thus, the optimal rates follow immediately by properly setting η_t and α_t, as captured by the following corollary.

Corollary F.4.1.1 (Accelerated Proximal Dual-Averaging)**.** *Let f and ϕ be convex and assume that either f is L-smooth over $\mathbb{R}^d$ or otherwise (F.2) holds for all $x, y \in \mathcal{X}$. Consider the online-averaged (stochastic) proximal dual averaging algorithm, given by Algorithm 17 with update (F.6) using $\tilde{g}_t = g_{t-1}$ as the optimistic prediction of g_t for $t > 1$, and $\tilde{g}_1 = 0$, where the gradient estimates g_t are unbiased, that is, $\mathbb{E}g_t|\bar{x}_t = f'(\bar{x}_t)$. Let $\sigma_*^2 = \max_{t=1}^{T}\mathbb{E}\|\sigma_t\|_2^2$, where $\sigma_t = g_t - f'(\bar{x}_t)$, and let $D = \max\{\|x^\star\|_2, \|x_1 - x_f^*\|_2\}$, where x_f^* is the minimizer of f over $\mathbb{R}^d$. Then we have the following error bounds:*

(i) If $\eta_t = 4L + \eta\alpha_t\sqrt{t}$ for some $\eta > 0$ and $\alpha_t = t$, we have

$$\mathbb{E}\ell(\bar{x}_T) - \ell^\star \le \frac{\left(4L + \frac{L}{4} + \eta T\sqrt{T}\right)D^2 + \frac{4\sigma_*^2}{\eta}T\sqrt{T}}{T(T+1)} = \mathcal{O}\left(\frac{LD^2}{T^2} + \frac{\eta D^2 + \eta^{-1}\sigma_*^2}{\sqrt{T}}\right).$$

(ii) If ϕ_t is μ-strongly-convex then using $\eta_t = 4L$ and $\alpha_t = t$, we have

$$\mathbb{E}\ell(\bar{x}_T) - \ell^\star \le \frac{\left(4L + \frac{L}{4}\right)D^2 + \frac{8\sigma_*^2 T}{\mu}}{T(T+1)} = \mathcal{O}\left(\frac{LD^2}{T^2} + \frac{\sigma_*^2}{\mu T}\right).$$

(iii) If $g_t = f'(x_t)$ (i.e., the noiseless case) and ϕ is μ-strongly-convex, then for $\eta_t = 0$ and

any sequence of $\alpha_t > 0, t \in [T]$ *satisfying*

$$\sqrt{c\kappa} \geq \frac{\alpha_{1:t}}{\alpha_t} \geq \sqrt{2\kappa} \qquad t > 1\,, \tag{F.13}$$

for some $c \geq 2$ *where* $\kappa = (L+\mu)/\mu$ *denotes the condition number, we have*

$$\ell(\bar{x}_T) - \ell^\star \leq \frac{\|f'(x_1)\|^2 \left(1 - \frac{1}{\sqrt{c\kappa}}\right)^{T-1}}{2\mu}\,. \tag{F.14}$$

Remark F.4.1.1. *The above rates of* $\mathcal{O}\left(1/T^2\right)$ *for a non-strongly-convex* f *are optimal in* T *when there is no noise (*$\sigma_t = 0$*), and the bound (F.14) also almost matches the optimal* $\mathcal{O}\left((1-1/\sqrt{\kappa})^T\right)$ *rate for the noiseless strongly-convex case. When there is noise, the worst-case rate of* $\mathcal{O}\left(1/\sqrt{T}\right)$ *(for non-strongly-convex* f*) and* $\mathcal{O}\left(1/T\right)$ *(for strongly-convex* f*) are unavoidable, according to the lower-bounds of Nemirovsky and Yudin [167]: when the noise dominates, there is no hope of exploiting the smoothness in the signal (i.e., the gradient). Therefore, similarly to our paper, all previous work obtain only a lower-order improvement, e.g., from* $1/T + \sigma/\sqrt{T}$ *(of smooth non-strongly-convex SGD) to* $1/T^2 + \sigma/\sqrt{T}$*. If the noise is small, the latter rate is closer to the noise-free optimal rate of* $1/T^2$*, and determines the convergence speed of the algorithm in the initial stages of optimization. In contrast, the former bound (for SGD) is sub-optimal in the noise-free case. The possible improvements are lower-order in case of noisy strongly-convex optimization as well.*

Proof of theorem F.4.1.1. First, notice that with any step size $\eta_t = 4L + \gamma_t$, the algorithm is equivalent to algorithm 17 with AO-FTRL as the base algorithm, using regularizers $r_{0:t-1} = \frac{4L+\gamma_t}{2}\|\cdot\|_2^2$, which satisfy the conditions of theorem F.3.0.3 with $\beta_t = 4$, $\|\cdot\|^2 = L\|\cdot\|_2^2$, and $\|\cdot\|_{(t)}^2 = (\gamma_t + \alpha_{1:t}\mu)\|\cdot\|_2^2$, where μ is the strong-convexity parameter of ϕ (i.e, $\mu = 0$ in part (i), and $\mu > 0$ in parts (ii) and (iii)). Hence, starting from (F.12), with $\alpha_t = t$ we have

$$\mathbb{E}\ell(\bar{x}_T) - \ell(x^\star) \leq \sum_{t=1}^{T} \frac{t^2\mathbb{E}\|\sigma_t - \sigma_{t-1}\|_2^2}{(\gamma_t + \alpha_{1:t}\mu)T(T+1)} + \frac{(4L+\gamma_T)\|x^\star\|_2^2}{T(T+1)} - \frac{\sum_{t=1}^{T}(\gamma_t - \gamma_{t-1})\mathbb{E}\|x_t\|_2^2}{T(T+1)} + \frac{\mathbb{E}\|f'(\bar{x}_1)\|_2^2}{4T(T+1)L}\,.$$

In the above, $\mathbb{E}\|\sigma_t - \sigma_{t-1}\|_2^2 \leq 4\sigma_*^2$. In addition, since f is convex and satisfies (F.2), we have $\frac{1}{2L}\|f'(\bar{x}_1)\|_2^2 \leq B_f(x_1, x_f^*) \leq \frac{L}{2}\|x_1 - x_f^*\|^2$ where x_f^* is the minimizer of f over $\mathbb{R}^d$. Then, plugging in $\gamma_t = \eta\alpha_t\sqrt{t}$ (respectively, $\gamma_t = 0$) and dropping the non-positive terms $-(\gamma_t - \gamma_{t-1})\|x_t\|_2^2$ immediately gives part (i) (respectively, part (ii)).

To prove part (iii), first recall that for $\eta_{t,j} > 0$, (F.6) is equivalent to the AO-FTRL update (F.4) with $r_{0:t-1}(x) = \frac{1}{2}\sum_{j=1}^{d}\eta_{t,j}x_j^2$. For $\eta_t = 0$, we define the update to be AO-FTRL with

$r_{0:t-1} = 0$, hence the update will be of the form $x_t = \arg\min_{x\in\mathcal{X}} \langle z_{t-1}, x\rangle + \alpha_{1:t}\phi(x)$ (recall that $z_{t-1} = -\sum_{s=1}^{t-1} \alpha_s g_s - \alpha_t \tilde{g}_t$). Then, since ϕ is strongly-convex, despite having $r_s = 0$ for all s, we have that $\alpha_{1:t}\phi + r_{0:t-1}$ is strongly-convex w.r.t. $\beta_t \frac{L}{2}\|\cdot\|_2^2$ with $\beta_t = \alpha_{1:t}\frac{\mu}{L}$. Hence, by theorem F.3.0.3,[3] we have

$$\alpha_{1:T}\left(\ell(\bar{x}_T) - \ell(x^\star)\right) \le \frac{\alpha_1}{2\mu}\|f'(x_1) - \tilde{g}_1\|^2, \tag{F.15}$$

as long as for all $t > 1$, the assumption $\alpha_t^2 \beta_t^{-1} \le \alpha_{1:t-1}$ of theorem F.3.0.3 is satisfied: that is, we have

$$\frac{\alpha_t^2}{\alpha_{1:t}\alpha_{1:t-1}} \le \frac{\mu}{L} = \frac{1}{\kappa - 1}. \tag{F.16}$$

It remains to show that (F.16) is satisfied, and simplify the bound (F.15). To that end, note that on the one hand, by (F.13) we have $\alpha_{1:t-1}/\alpha_t \ge \sqrt{2\kappa} - 1$, which in turn implies $\frac{\alpha_{1:t}\alpha_{1:t-1}}{\alpha_t^2} \ge 2\kappa - \sqrt{2\kappa} \ge \kappa - 1$, proving (F.16). On the other hand, (F.13) implies $\alpha_{1:t-1} \le (1 - \frac{1}{\sqrt{c\kappa}})\alpha_{1:t}$ for all $t > 1$; therefore $\alpha_1 \le \alpha_{1:T}\left(1 - \frac{1}{\sqrt{c\kappa}}\right)^{T-1}$. Putting this back into (F.15) finishes the proof. □

F.4.2 A Proximal Adaptive Universal Algorithm

Next, we present the universal convergence of algorithm 17 with AdaGrad-style step sizes, proved in appendix F.8.

Theorem F.4.2.1. *Suppose that the iterates x_t are given by AO-FTRL with AdaGrad step sizes, i.e., using (F.4) with $r_0 = 0$,*

$$r_t(x) = \gamma \sum_{j=1}^{d} \frac{\eta_{t,j} - \eta_{t-1,j}}{2}(x_j - x_{t,j})^2, \qquad t \ge 1,$$

where $\gamma > 0$, $\eta_{t,j} = \sqrt{\sum_{s=1}^{t} \alpha_s^2 (g_{s,j} - \tilde{g}_{s,j})^2}, t > 0$ and $\eta_0 = 0$. Further suppose that g_t are unbiased estimates of $f'(\bar{x}_t)$, and we use $\tilde{g}_t = g_{t-1}, t > 1$ and $\tilde{g}_1 = 0$. Let R be an upper-bound on $|x_j^\star - x_{t,j}|^2$. Then the following hold:

(i) If $\mathbb{E}g_{t,j}^2 \le G_j^2$ for all $t \in [T]$, then

$$\mathbb{E}\ell(\bar{x}_T) - \ell^\star \le \sum_{j=1}^{d} \mathbb{E}\frac{\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)}{\alpha_{1:T}} \sqrt{\sum_{t=1}^{T} \alpha_t^2 G_{t,j}^2} = \mathcal{O}\left(\frac{R\sum_{j=1}^{d} G_j}{\sqrt{T}}\right),$$

[3] Note that instead of using a norm, here we set $\|\cdot\|_{(t)}$ in theorem F.3.0.3 to be zero. While this is not a valid choice, an inspection of the proof of the theorem verifies that the theorem still holds in this case if the dual norm is set to zero and $\sigma_t = 0$ for all t.

for $\gamma = 2/R$, where $G_{t,j} := (g_{t,j} - \tilde{g}_{t,j})$.

(ii) If f is L-smooth over $\mathbb{R}^d$ or otherwise (F.2) holds for all $x, y \in \mathcal{X}$, and $\mathbb{E}\sigma_{t,j}^2 \le \sigma_j^2$ for all $t \in [T]$ (recall that $\sigma_t = g_t - f'(\bar{x}_t)$), then

$$\mathbb{E}\ell(\bar{x}_T) - \ell^\star \le \frac{1}{\alpha_{1:T}} \sum_{j=1}^{d} 6L \left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)^2 + \frac{1}{\alpha_{1:T}} \left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right) \left(\Delta + \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{T} 6\alpha_t^2 \sigma_j^2}\right)$$
$$= \mathcal{O}\left(\frac{LdR^2 + \Delta R}{T^2} + \frac{\max_j \sigma_j d R}{\sqrt{T}}\right),$$

for $\gamma = 2/R$, where $\Delta = \sum_{j=1}^{d} \sqrt{2\mathbb{E}|f'(x_{1,j})|^2}$.

Remark F.4.2.1. *Both bounds above are achieved by the same algorithm, without further prior knowledge about f or the values of L and σ. The first bound is a data-adaptive bound that holds even if f is non-smooth, and is optimal when the data is sparse [61]. The second bound is of the optimal rate $\mathcal{O}(1/T^2 + \sigma/\sqrt{T})$ when f is smooth.*

Remark F.4.2.2. *The bound R required by the theorem is enforced, e.g., when $\mathcal{X}$ is compact. This implies that in the unconstrained optimization setting, similarly to Levy* et al. *[134], we assume that we are still given a compact set $\mathcal{X}$ containing $x^\star$ and project to that set in the algorithm.*

F.5 Accelerated Variance-Reduced Methods

In this section, we apply our framework to the variance reduced setting. In this setting, we assume $f = \mathbb{E}F(\cdot, \xi)$ is the expected value of functions $F : \mathbb{R}^d \times \Xi \to \mathbb{R}$, where ξ is a random variable from some set Ξ, with distribution P_Ξ. At time step t, the algorithm receives a realization $\zeta_t \sim P_\Xi$, and can query the gradient oracle $F'(\cdot, \zeta_t)$ at (potentially multiple) points in $\mathcal{X}$. In addition, the algorithm can query the exact (non-stochastic) gradient oracle f' from time to time. Then, the gradient estimate g_t at $\bar{x}_t$ is calculated as

$$g_t = F'(\bar{x}_t, \zeta_t) - F'(\tilde{x}_t, \zeta_t) + f'(\tilde{x}_t), \tag{F.17}$$

where $\tilde{x}_t$ is the *snapshot* point at time t, i.e., the most recent point at which f' has been queried prior to time t.

The underlying operational assumption in computing g_t is that calls to F' are computationally cheaper than calls to f', and hence the latter is queried less frequently. This is in particular the case in finite sum minimization problems, where $f = \frac{1}{n}\sum_{i=1}^{n} f_i$ for some functions $f_i : \mathcal{X} \to \mathbb{R}$, $F(x, i) = f_i(x)$ for all $i \in [n]$, and ζ_t has a uniform distribution over $[n]$. In this case, the computational complexity of an algorithm is measured by the number of times the gradient of any f_i is computed, so a single access to the full gradient oracle f' has a computation cost of $\mathcal{O}(n)$.

Let $\mathcal{H}_1 = \emptyset$ and $\mathcal{H}_t = \{\zeta_1, \zeta_2, \ldots, \zeta_{t-1}\}$ for $t > 1$, i.e., $\mathcal{H}_t$ is the history of random realizations up to time t. To ensure g_t is an unbiased estimate of $f'(\bar{x}_t)$, i.e., $\mathbb{E} g_t | \mathcal{H}_t = f'(\bar{x}_t)$, we assume that for any t and any $\mathcal{H}_t$-measurable x,

$$\begin{aligned} &\mathbb{E} F'(x, \zeta_t) | \mathcal{H}_t = f'(x)\,, \text{and,} \\ &\mathbb{E} F(x, \zeta_t) | \mathcal{H}_t = f(x)\,. \end{aligned} \tag{F.18}$$

This is ensured, e.g., if $\zeta_t, t = 1, 2, \ldots$ is an i.i.d. sequence.

Algorithm. In this setting, instead of defining the query point $\bar{x}_t$ as the average of the previous outputs of the underlying online optimization algorithm $\mathcal{A}$, we define it as

$$\bar{x}_t = \frac{\alpha_{1:t-1}\bar{x}_{t-1} + \alpha_t x_t + p_t \tilde{x}_t}{\alpha_{1:t} + p_t} \tag{F.19}$$

where the $\alpha_t > 0$ are the averaging weights as before, and $p_t \geq 0$ incorporates a *negative momentum* (first introduced by Allen-Zhu [6]) towards the current snapshot point $\tilde{x}_t$. If $p_t = 0$, (F.19) reduces back to $\bar{x}_t = \frac{1}{\alpha_{1:t}} \sum_{s=1}^{t} \alpha_s x_s$.

The resulting algorithm, presented in algorithm 18, extends Algorithm 1 of Joulani *et al.* [103] to the anytime averaging scheme with negative momentum. algorithm 18 operates in epochs (the outer loop in the algorithm goes over the epochs): At the beginning of epoch s, the gradient snapshot is calculated. Then, in the sth run of the inner loop, from time $T_{1:s-1} + 1$ to $T_{1:s}$, an optimization algorithm $\mathcal{A}$ is run for T_s steps with the variance reduced gradient estimates (F.17) and averaging (F.19). Finally, the snapshot point is updated at the end of the epoch; the exact form of the update is given later for the different variants we consider.

Algorithm 18 Variance-Reduced Anytime Online-to-Batch with Negative Momentum

Input: Gradient oracle F' and f', non-negative weights $(\alpha_t)_{t=1}^{T}$ with $\alpha_1 > 0$, epoch lengths $T_1, T_2, \ldots, T_S$, online linear optimization algorithm $\mathcal{A}$
Get the initial point $x_1 \in \mathcal{X}$ from $\mathcal{A}$
$\tilde{x} \leftarrow x_1$, $\bar{x}_1 \leftarrow x_1$
for $s = 1$ **to** S **do**
  Compute and store the full gradient $f'(\tilde{x})$
  for $t = T_{1:s-1} + 1$ **to** $T_{1:s}$ **do**
    Get the gradient estimate g_t at $\bar{x}_t$ by (F.17)
    Send $\langle \alpha_t g_t, \cdot \rangle$ as the next linear loss to $\mathcal{A}$
    Let x_{t+1} be the next iterate from $\mathcal{A}$
    Let $\bar{x}_{t+1} \leftarrow \frac{\alpha_{1:t}\bar{x}_t + \alpha_{t+1}x_{t+1} + p_{t+1}\tilde{x}}{\alpha_{1:t+1} + p_{t+1}}$
  end for
  Update the snapshot point $\tilde{x}$.
end for
return the average iterate $\bar{x}_T$ and the latest snapshot $\tilde{x}$.

F.5.1 Warm-Up: No Negative Momentum

First, we consider a version of our accelerated variance-reduced method without negative momentum ($p_t = 0$ for all t), using the first iterate of each epoch (i.e., the last iterate of the previous epoch $s-1$ for $s > 1$) as the snapshot point: We let $\tilde{x} = \bar{x}_{t+1}$, so that in every epoch $s \in [S]$, $\tilde{x}_t = \bar{x}_{T_{1:s-1}+1}$ for all $t \in [T_{1:s-1}+1, T_{1:s}]$. We use AO-FTRL with regularizer $r_{1:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$ as the underlying algorithm $\mathcal{A}$, with the snapshot used as the optimistic gradient estimate: $\tilde{g}_t = f'(\tilde{x}_{t-1})$. Then, we have the following bound on the performance of the algorithm:

Theorem F.5.1.1. *Suppose that f, as well as $F(\cdot,\zeta)$ for all $\zeta \in \Xi$, are a) convex; and, b) either L-smooth w.r.t. $\|\cdot\|_2$ over $\mathbb{R}^d$ or otherwise satisfying (F.2) for all $x, y \in \mathcal{X}$. Further suppose that (F.18) holds. Assume that algorithm 18 is run with epoch lengths $T_s = \min\{\tau, 2^{s-1}\}$ for some maximum epoch length τ, snapshot update $\tilde{x} = \bar{x}_{t+1}$, $\alpha_t = t$, and $\mathcal{A}$ selected as AO-FTRL with regularizer $r_{1:t-1} = \frac{\eta_t}{2}\|\cdot\|_2^2$ for $\eta_t = 8L\tau^2$ and optimistic gradient estimates $\tilde{g}_1 = 0$ and $\tilde{g}_t = f'(\tilde{x}_{t-1}), t > 1$. Then, for any $T > \tau$,*

$$\mathbb{E}\ell(\bar{x}_T) - \ell(x^\star) \le \frac{8L\tau^2\|x^\star\|_2^2 + \frac{\|f'(\bar{x}_1)\|_2^2}{8L\tau^2}}{T(T+1)}\,.$$

Proof. By theorem F.11.0.1 in appendix F.11,

$$\mathcal{R}_T(x^\star) = \frac{\eta_T}{2}\|x^\star\|_2^2 + \sum_{t=1}^{T}\frac{\alpha_t^2}{2\eta_t}\|g_t - \tilde{g}_t\|_2^2$$

bounds the linearized composite-objective regret of $\mathcal{A}$. Combining with theorem F.3.0.2 and using $B_{1:T} \ge 0$,

$$\mathbb{E}\ell(\bar{x}_T) - \ell(x^\star) \le \frac{1}{\alpha_{1:T}}\mathbb{E}\frac{\eta_T}{2}\|x^\star\|_2^2 + \sum_{t=1}^{T}\frac{\alpha_t^2}{2\eta_t}\|g_t - \tilde{g}_t\|_2^2 - \bar{B}^f_{2:T}\,.$$

theorem F.9.0.1 in Appendix F.9 shows that

$$\sum_{t=2}^{T}\alpha_t^2\|g_t - \tilde{g}_t\|_2^2 \le 16L\tau^2\bar{B}^f_{2:T},$$

which then can be used to cancel all terms but $\|g_1 - \tilde{g}_1\|_2^2/(2\eta_1)$ from the summation above. Using $g_1 = f'(\bar{x}_1)$ and $\tilde{g}_1 = 0$, and substituting η_t finishes the proof. □

In the finite sum optimization setting, by selecting $\tau = n$, our algorithm achieves ε error after $\mathcal{O}\left(n\log n + n\sqrt{\frac{L}{\varepsilon}}\right)$ individual gradient evaluations, via a simple direct approach. More complicated methods, such as Catalyst [138], RPDG [124], Katyusha [6] and related papers achieve an iteration complexity of $\mathcal{O}\left(n\log\frac{1}{\varepsilon} + \sqrt{\frac{nL}{\varepsilon}}\right)$, which has a

better dependence on n in the dominant second term. However, these methods use an indirect approach (as termed by Allen-Zhu [6]), where non-strongly-convex functions are optimized by adding strongly-convex perturbations, and yet do not achieve the near-optimal rate of Lan *et al.* [125], which is obtained using negative momentum and epoch averaging. In the next section, we obtain this near-optimal bound.

F.5.2 Improved Variance-Reduced Acceleration

In this section, we use negative momentum to achieve a near-optimal accelerated variance-reduced rate: we set $p_t > 0$ in algorithm 18. In addition, unlike theorem F.5.1.1, the snapshot point at the end of epoch s is now given by an average:

$$\tilde{x}_{s+1} = \frac{1}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}} p_t} \sum_{t=T_{1:s-1}+1}^{T_{1:s}} p_t \, \bar{x}_t \,. \tag{F.20}$$

For simplicity, we assume $\phi = 0$, so that $\ell = f$.

A consequence of computing $\bar{x}_t$ via (F.19) with $p_t > 0$ is that for all $t = 1, 2, \ldots, T$,

$$\alpha_t(\bar{x}_t - x_t) = \alpha_{1:t-1}(\bar{x}_{t-1} - \bar{x}_t) + p_t(\tilde{x}_t - \bar{x}_t) \,. \tag{F.21}$$

Then, we will have the following error decomposition.

Lemma F.5.2.1 (Regret Decomposition)**.** *For $t \in [T]$, let $\alpha_t, p_t > 0$, $x_t, \tilde{x}_t \in \mathbb{R}^d$, and define $\bar{x}_t$ as in Equation (F.19), $B_t = \alpha_t B_f(x^\star, \bar{x}_t)$ and $\bar{B}_t^f = \alpha_{1:t-1} B_f(\bar{x}_{t-1}, \bar{x}_t)$. Then, for all $x^\star \in \mathbb{R}^d$,*

$$f(\bar{x}_T) - f^\star = \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\bar{x}_t), x_t - x^\star \rangle - B_{1:T} - \sum_{t=2}^{T} \bar{B}_t^f \right. \\ \left. + \sum_{t=1}^{T} p_t (f(\tilde{x}_t) - f(\bar{x}_t) - B_f(\tilde{x}_t, \bar{x}_t)) \right] \tag{F.22}$$

The above error decomposition, proved in appendix F.10, is similar to theorem F.3.0.1, but has an extra term due to the negative momentum, which will be helpful in further reducing the error. Then, the next theorem, proved in appendix F.10, provides the improved convergence rate.

Theorem F.5.2.2. *Consider the conditions of theorem F.5.1.1, but instead suppose that the snapshot update in appendix F.5 of algorithm 18 is given by (F.20), $\tilde{g}_t = g_{t-1}, t > 1$, we use p_t such that $0 < p_1 \leq 1$ and $p_t \geq \frac{15L\alpha_t^2}{\eta_t}, t \geq 1$, and we set $\eta_t = 1860LT_{s(t)}\log(2t)$,*

where $s(t)$ denotes the epoch containing iteration t. Then, for any $T \geq 1$,

$$\mathbb{E}\ell(\bar{x}_T) - \ell(x^\star) \leq \frac{3720LT_{s(T)}\|x^\star\|_2^2 + \frac{\|f'(\tilde{x}_1)\|_2^2}{930L\log 2} + 4(f(\tilde{x}_1) - f^\star)}{T^2}\log(2T).$$

The rate provided in Theorem F.5.2.2 is optimal up to a logarithmic factor. In particular, for the finite sum setting, with $\tau = n$, the algorithm needs $\tilde{\mathcal{O}}\left(n\log n + \sqrt{\frac{nL}{\varepsilon}}\right)$ individual gradient evaluations to reach ε error, matching the rate recently obtained by Lan *et al.* [125]. Unlike in previous work, our convergence guarantee holds for the last iterate instead of a snapshot point or the average of the last epoch.

F.6 Conclusions

We demonstrated that online iterate averaging combined with optimistic online learning can lead to accelerated rates in several scenarios. The resulting algorithms and their analyses are surprisingly simple and often yield the optimal rates. Exploring the full power of this method is left for future work. In particular, it would be interesting to extend this approach to obtain accelerated exponential rates for variance-reduced optimization of strongly-convex objectives, and remove the extra logarithmic terms in the non-strongly-convex variance-reduction case.

Proofs for Main Results

F.7 Proof of theorem F.3.0.3

Proof of theorem F.3.0.3. First, we bound the linear composite regret

$$\sum_{t=1}^{T} \alpha_t \left(\langle g_t, x_t - x^\star \rangle + \phi(x_t) - \phi(x^\star) \right)$$

by a bound $\mathcal{R}_T$ that results in slightly better constants compared to (F.7). For $t \in [T]$, let u_t, v_t be any two vectors such that $g_t - \tilde{g}_t = u_t + v_t$, and define $D_t = r_{t-1}(x^\star) - r_{t-1}(x_t)$. Then, theorem F.11.0.1 bounds the regret of the AO-FTRL updates made by $\mathcal{A}$ as follows:

$$\begin{aligned}
&\sum_{t=1}^{T} \left(\langle \alpha_t g_t, x_t - x^\star \rangle + \alpha_t \phi(x_t) - \alpha_t \phi(x^\star) \right) - D_{1:T} \\
&\leq \sum_{t=1}^{T} \left(\alpha_t \langle g_t - \tilde{g}_t, x_t - x_{t+1} \rangle - B_{\alpha_{1:t}\phi + r_{0:t-1}}(x_{t+1}, x_t) \right) \\
&\leq \sum_{t=1}^{T} \left(\langle \alpha_t u_t + \alpha_t v_t, x_t - x_{t+1} \rangle - \frac{\beta_t}{2} \|x_t - x_{t+1}\|^2 - \frac{1}{2} \|x_t - x_{t+1}\|_{(t)}^2 \right) \\
&\leq \sum_{t=1}^{T} \frac{\alpha_t^2 \|v_t\|_{(t)*}^2}{2} + \sum_{t=1}^{T} \frac{\alpha_t^2 \|u_t\|_*^2}{2\beta_t},
\end{aligned} \tag{F.23}$$

where the second step uses the strong-convexity of $\alpha_{1:t}\phi + r_{0:t-1}$, and the third step follows by the Fenchel-Young inequality $\langle z, x \rangle - \frac{\beta}{2}\|x\|^2 \leq \frac{1}{2\beta}\|z\|_*^2$. Now, let $u_1 = f'(\mathbf{x}_1) - \tilde{g}_1$, $u_t = f'(\mathbf{x}_t) - f'(\mathbf{x}_{t-1}), t = 2, \ldots, T$, and $v_t = \sigma_t - \sigma_{t-1}, t \in [T]$, and notice that for all $t = 2, \ldots, T$,

$$\begin{aligned}
\frac{\alpha_t^2 \|u_t\|_*^2}{2\beta_t} &\leq \frac{\alpha_{1:t-1} \|u_t\|_*^2}{2} = \frac{\alpha_{1:t-1}}{2} \|f'(\mathbf{x}_t) - f'(\mathbf{x}_{t-1})\|_*^2 \\
&\leq \alpha_{1:t-1} B_f(\mathbf{x}_{t-1}, \mathbf{x}_t) = \bar{B}_t^f,
\end{aligned} \tag{F.24}$$

using the assumption of $\alpha_t^2 \beta_t^{-1} \leq \alpha_{1:t-1}$ and (F.2) with $L = 1$. Plugging the definitions of u_t and v_t into (F.23) and using (F.24), we obtain

$$\sum_{t=1}^{T} \langle \alpha_t g_t, x_t - x^\star \rangle \leq \sum_{t=1}^{T} \left(D_t + \alpha_t \phi(x_t) - \alpha_t \phi(x^\star) \right)$$

$$+\sum_{t=1}^{T}\frac{\alpha_t^2\|\sigma_t-\sigma_{t-1}\|_{(t)*}^2}{2}+\frac{\alpha_1^2\|f'(\mathbf{x}_1)-\tilde{g}_1\|_*^2}{2\beta_1}+\bar{B}_{2:T}^f\,. \tag{F.25}$$

Applying theorem F.3.0.2, combining with (F.25), and cancelling the matching $\bar{B}_{2:T}$ terms concludes the proof. □

F.8 Proof of theorem F.4.2.1

Proof of theorem F.4.2.1 Starting from theorem F.3.0.1, we plug-in the bound of composite AO-FTRL from Mohri and Yang [157, Theorem 3] (using $f_t \leftarrow \langle \alpha_t g_t, \cdot\rangle$):

$$\begin{aligned}
&\sum_{t=1}^{T}\alpha_t\left(\langle g_t, x_t-x^\star\rangle+\phi(x_t)-\phi(x^\star)\right)\le r_{0:T}(x^\star)+\sum_{t=1}^{T}\alpha_t^2\|g_t-\tilde{g}_t\|_{(t)*}^2\\
&\le\frac{\gamma R^2}{2}\sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T}\alpha_t^2(g_{t,j}-\tilde{g}_{t,j})^2}+\frac{1}{\gamma}\sum_{j=1}^{d}\sum_{t=1}^{T}\frac{\alpha_t^2(g_{t,j}-\tilde{g}_{t,j})^2}{\sqrt{\sum_{s=1}^{t}\alpha_s^2(g_{s,j}-\tilde{g}_{s,j})^2}}\\
&\le\left(\frac{\gamma R^2}{2}+\frac{2}{\gamma}\right)\sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T}\alpha_t^2(g_{t,j}-\tilde{g}_{t,j})^2}\,,
\end{aligned}$$

where the first inequality follows from Mohri and Yang [157, Theorem 3] with the norm $\|y\|_{(t)*}=\sum_{j=1}^{d}\frac{1}{\gamma\eta_{t,j}}(y_j)^2$, the second from the definitions using $r_{0:T}\le\sum_{t=1}^{T}\sum_{j=1}^{d}\frac{\gamma R^2}{2}(\eta_{t,j}-\eta_{t-1,j})=\sum_{j=1}^{d}\frac{\gamma R^2}{2}\eta_{T,j}$, and the third from the standard inequality that $\sum_{t=1}^{T}a_t/\sqrt{a_{1:t}}\le 2\sqrt{a_{1:T}}$.

Putting back into theorem F.3.0.2, we obtain

$$\mathbb{E}\ell(\mathbf{x}_T)-\ell^\star\le\frac{1}{\alpha_{1:T}}\mathbb{E}\left(\frac{\gamma R^2}{2}+\frac{2}{\gamma}\right)\sum_{j=1}^{d}\sqrt{\sum_{t=1}^{T}\alpha_t^2(g_{t,j}-\tilde{g}_{t,j})^2}-\bar{B}_{2:T}^f\,. \tag{F.26}$$

Dropping the negative terms $\bar{B}_{2:T}^f$, using Jensen's inequality to take the expectation under the square-root, using $\alpha_t=t$ and applying the bound G_j completes the proof of the first part of the theorem.

To prove the second part, observe that for $t>1$, because $\tilde{g}_t=g_{t-1}$, we have

$$|g_{t,j}-\tilde{g}_{t,j}|=|f_j'(\mathbf{x}_t)+\sigma_{t,j}-f_j'(\mathbf{x}_{t-1})-\sigma_{t-1,j}|\le|\sigma_{t,j}|+|\sigma_{t-1,j}|+|f_j'(\mathbf{x}_t)-f_j'(\mathbf{x}_{t-1})|\,.$$

Now, denote $\Delta_{1,j}=|f_j'(\bar{x}_1)|$ and $\Delta_{t,j}=|f_j'(\mathbf{x}_t)-f_j'(\mathbf{x}_{t-1})|, t>1$. By Jensen's inequality, $(a+b+c)^2\le 3(a^2+b^2+c^2)$ for any real numbers a,b,c. In addition, for $t>1$,

$\sum_{j=1}^d \Delta_{t,j}^2 = \|f'(\mathbf{x}_t) - f'(\mathbf{x}_{t-1})\|_2^2 \leq 2LB_f(\mathbf{x}_{t-1}, \mathbf{x}_t)$ by (F.2). Hence,

$$\sum_{t=2}^T \alpha_t^2 (g_{t,j} - \tilde{g}_{t,j})^2 \leq \sum_{t=2}^T 3\alpha_t^2 \left[\sigma_{t,j}^2 + \sigma_{t-1,j}^2 + \Delta_{t,j}^2\right], \tag{F.27}$$

on the one hand, and on the other hand

$$-\bar{B}_{2:T}^f \leq -\sum_{j=1}^d \sum_{t=2}^T \frac{\alpha_{1:t-1}}{2L} \Delta_{t,j}^2 .$$

Next, using $\alpha_1 = 1$ and $\tilde{g}_1 = 0$, we have $\alpha_1^2 (g_{1,j} - \tilde{g}_{1,j})^2 \leq 2\sigma_{1,j}^2 + 2\Delta_{1,j}^2$ by Jensen's inequality. Putting back into (F.26),

$$\begin{aligned}
\mathbb{E}\ell(\mathbf{x}_T) - \ell^\star &\leq \frac{1}{\alpha_{1:T}} \left[\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right) \sum_{j=1}^d \mathbb{E} \sqrt{2\sigma_{1,j}^2 + 2\Delta_{1,j}^2 + \sum_{t=2}^T 3\alpha_t^2 \left[\sigma_{t,j}^2 + \sigma_{t-1,j}^2 + \Delta_{t,j}^2\right]} - \mathbb{E}\bar{B}_{2:T}^f \right] \\
&\leq \frac{1}{\alpha_{1:T}} \sum_{j=1}^d \left[\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right) \sqrt{\sum_{t=1}^T 6\alpha_t^2 \sigma_j^2 + 2\mathbb{E}\Delta_{1,j}^2 + \sum_{t=2}^T 3\alpha_t^2 \mathbb{E}\Delta_{t,j}^2} - \sum_{t=2}^T \frac{\alpha_{1:t-1}}{2L} \mathbb{E}\Delta_{t,j}^2 \right] \\
&\leq \frac{1}{\alpha_{1:T}} \sum_{j=1}^d \left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right) \sqrt{\sum_{t=1}^T 6\alpha_t^2 \sigma_j^2 + 2\mathbb{E}\Delta_{1,j}^2} \\
&\quad + \frac{1}{\alpha_{1:T}} \sum_{j=1}^d \left[\left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right) \sqrt{24L \sum_{t=2}^T \frac{\alpha_{1:t-1}}{2L} \mathbb{E}\Delta_{t,j}^2} - \sum_{t=2}^T \frac{\alpha_{1:t-1}}{2L} \mathbb{E}\Delta_{t,j}^2 \right],
\end{aligned}$$

using in the second step the concavity of the square root, Jensen's inequality and the upper-bound $\mathbb{E}\sigma_{s,j}^2 \leq \sigma_j, s \geq 1$, and in the last step $\alpha_t = t$ and the fact that for $t > 1$, $\frac{\alpha_t^2}{\alpha_{1:t-1}} \leq 4$. Next, we note that for $a, b \geq 0$, $2\sqrt{ab} - b \leq a$. Therefore,

$$\begin{aligned}
\mathbb{E}\ell(\mathbf{x}_T) - \ell^\star &\leq \frac{1}{\alpha_{1:T}} \sum_{j=1}^d \left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right) \sqrt{\sum_{t=1}^T 6\alpha_t^2 \sigma_j^2 + 2\mathbb{E}\Delta_{1,j}^2} \\
&\quad + \frac{1}{\alpha_{1:T}} \sum_{j=1}^d 6L \left(\frac{\gamma R^2}{2} + \frac{2}{\gamma}\right)^2 .
\end{aligned}$$

Separating the first square-root and using $\Delta = \sum_{j=1}^d \sqrt{2\mathbb{E}\Delta_{1,j}^2}$ completes the proof of the second part. □

F.9 Variance reduction for smooth functions

Lemma F.9.0.1. *Suppose the assumptions of theorem F.5.1.1 hold. Then*

$$\sum_{t=2}^{T} \alpha_t^2 \mathbb{E}\|g_t - \tilde{g}_t\|_2^2 \leq 16L\tau^2 \bar{B}_{2:T}^f \,.$$

Proof. Fix time step $t > 1$ within epoch $s > 1$ (recall there is only one step, $t = 1$, in epoch $s = 1$). We consider two cases:

a) Time step t is the first time step in epoch s:

This implies that $t = T_{1:s-1} + 1$. In addition, $\tilde{x}_t = \bar{x}_t$ by definition, and $\tilde{x}_{t-1} = \bar{x}_{t'}$ where $t' = T_{1:s-2} + 1$ is the first iterate of epoch $s-1$. Then,

$$\begin{aligned}
\mathbb{E}\|g_t - \tilde{g}_t\|_2^2 | \mathcal{H}_t &= \mathbb{E}\left\|F'(\mathbf{x}_t, \zeta_t) - F'(\tilde{x}_t, \zeta_t) + f'(\tilde{x}_t) - f'(\tilde{x}_{t-1})\right\|_2^2 | \mathcal{H}_t \\
&= \left\|f'(\tilde{x}_t) - f'(\tilde{x}_{t-1})\right\|_2^2 = \|f'(\mathbf{x}_t) - f'(\mathbf{x}_{t'})\|_2^2 \\
&= \left\| \sum_{k=T_{1:s-2}+1}^{T_{1:s-1}} \left(f'(\mathbf{x}_{k+1}) - f'(\mathbf{x}_k)\right) \right\|_2^2 \\
&\leq T_{s-1} \sum_{k=T_{1:s-2}+1}^{T_{1:s-1}} \left\| \left(f'(\mathbf{x}_{k+1}) - f'(\mathbf{x}_k)\right) \right\|_2^2 \\
&\leq 2L\tilde{\tau}_t \sum_{k=t-\tilde{\tau}_t+1}^{t} B_f(\mathbf{x}_k, \mathbf{x}_{k-1}) \,,
\end{aligned}$$

where the first inequality follows by Jensen's inequality and the convexity of $\|\cdot\|^2$, and the second inequality follows because the smoothness assumption on that implies (F.2) holds for f and any $x, y \in \mathcal{X}$, and we substitute $\tilde{\tau}_t := T_{s-1}$.

b) Time step t is not the first time step in epoch s:

In this case, let $\tilde{\tau}_t = t - (T_{1:s-1} + 1) > 0$ denote the number of time steps elapsed since the beginning of the epoch, so that $\tilde{x}_{t-1} = \tilde{x}_t = \mathbf{x}_{t-\tilde{\tau}_t}$. Then,

$$\begin{aligned}
\mathbb{E}\|g_t - \tilde{g}_t\|_2^2 | \mathcal{H}_t &= \mathbb{E}\left\|F'(\mathbf{x}_t, \zeta_t) - F'(\tilde{x}_t, \zeta_t) + f'(\tilde{x}_t) - f'(\tilde{x}_{t-1})\right\|_2^2 | \mathcal{H}_t \\
&= \mathbb{E}\left\|F'(\mathbf{x}_t, \zeta_t) - F'(\mathbf{x}_{t-\tilde{\tau}_t}, \zeta_t)\right\|_2^2 | \mathcal{H}_t \\
&= \mathbb{E}\left[\left\| \sum_{k=t-\tilde{\tau}_t+1}^{t} \left(F'(\mathbf{x}_k, \zeta_t) - F'(\mathbf{x}_{k-1}, \zeta_t)\right) \right\|_2^2 \right] \mathcal{H}_t \\
&\leq \tilde{\tau}_t \sum_{k=t-\tilde{\tau}_t+1}^{t} \mathbb{E}\| \left(F'(\mathbf{x}_k, \zeta_t) - F'(\mathbf{x}_{k-1}, \zeta_t)\right) \|_2^2 | \mathcal{H}_t
\end{aligned}$$

$$\leq 2\tilde{\tau}_t L \sum_{k=t-\tilde{\tau}_t+1}^{t} \mathbb{E} B_{F(\cdot,\zeta_t)}(\mathbf{x}_{k-1},\mathbf{x}_k)|\mathcal{H}_t$$
$$= 2\tilde{\tau}_t L \sum_{k=t-\tilde{\tau}_t+1}^{t} B_f(\mathbf{x}_{k-1},\mathbf{x}_k)\,,$$

where the last inequality follows from the smoothness assumption that ensures (F.2) holds for $F(\cdot,\zeta_t)$ and any $x,y\in\mathcal{X}$, and the last equality follows by (F.18). Multiplying by α_t^2 and summing up for all t, we get

$$\sum_{t=2}^{T}\alpha_t^2\tilde{\tau}_t \sum_{k=t-\tilde{\tau}_t+1}^{t} B_f(\mathbf{x}_{k-1},\mathbf{x}_k) = \sum_{k=2}^{T} B_f(\mathbf{x}_{k-1},\mathbf{x}_k)\sum_{t=2}^{T}\alpha_t^2\tilde{\tau}_t\,\mathbb{I}\{t-\tilde{\tau}_t+1\leq k\leq t\}\,.$$

Next, recall that by definition, $T_s=\min(\tau,2^{s-1})\leq T_{1:s-1}+1$ for any $s\geq 1$. Therefore, in case (a) above, $t=T_{1:s-2}+T_{s-1}+1\leq 2T_{1:s-2}+2=2(t-\tilde{\tau}_t)$. Similarly, in case (b) above, $t\leq T_{1:s}\leq 2T_{1:s-1}+1\leq 2(t-\tilde{\tau}_t)$. Thus, using $\alpha_t=t$,

$$\sum_{t=2}^{T}\alpha_t^2\tilde{\tau}_t\,\mathbb{I}\{t-\tilde{\tau}_t+1\leq k\leq t\}\leq \alpha_{2k-2}^2\sum_{t=2}^{T}\tilde{\tau}_t\,\mathbb{I}\{k\leq t\leq k+\tilde{\tau}_t-1\}\leq 4\tau^2(k-1)^2\leq 8\tau^2\alpha_{1:k-1}\,,$$

using in the first step the fact that $t\leq 2(t-\tilde{\tau}_t)\leq 2(k-1)$ by the argument above and the condition inside the indicator (which we also re-arranged), and in the second step $\alpha_t=t$ and $\tilde{\tau}_t\leq T_s\leq\tau$. Combining the above, we get

$$\sum_{t=2}^{T}\alpha_t^2\mathbb{E}\|g_t-\tilde{g}_t\|_2^2\leq 16L\tau^2\sum_{t=2}^{T}\alpha_{1:t-1}B_f(\mathbf{x}_{t-1},\mathbf{x}_t)=16L\tau^2\bar{B}^f_{2:T},$$

finishing the proof. □

F.10 Improved variance-reduced rate for smooth functions

Proof of Lemma F.5.2.1. The proof follows similar steps as the proof of theorem F.3.0.1, with the difference that we need to handle the negative momentum term as well:

$$f(\mathbf{x}_T)-f^\star=\frac{1}{\alpha_{1:T}}\sum_{t=1}^{T}\alpha_t f(\mathbf{x}_t)-f^\star+\overbrace{f(\mathbf{x}_T)-\frac{1}{\alpha_{1:T}}\sum_{t=1}^{T}\alpha_t f(\mathbf{x}_t)}^{:=\varepsilon_T}$$
$$=\frac{1}{\alpha_{1:T}}\sum_{t=1}^{T}\alpha_t(f(\mathbf{x}_t)-f^\star)+\varepsilon_T$$

$$
\begin{aligned}
&= \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), \mathbf{x}_t - x^\star \rangle - \sum_{t=1}^{T} \overbrace{\alpha_t B_f(x^\star, \mathbf{x}_t)}^{:=B_t} \right] + \varepsilon_T \\
&= \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), x_t - x^\star \rangle + \sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), \mathbf{x}_t - x_t \rangle - B_{1:T} \right] + \varepsilon_T \\
&= \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), x_t - x^\star \rangle - B_{1:T} \right] + \overbrace{\frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), \mathbf{x}_t - x_t \rangle + \varepsilon_T}^{:=\tilde{\Delta}_T}
\end{aligned}
\tag{F.28}
$$

Before evaluating $\tilde{\Delta}_T$, we consider the term,

$$
\begin{aligned}
\Delta_T &= \frac{1}{\alpha_{1:T}} \left[\sum_{t=2}^{T} \langle f'(\bar{x}_t), \alpha_{1:t-1}(\bar{x}_{t-1} - \bar{x}_t) \rangle \right] + \varepsilon_T \\
&= \frac{1}{\alpha_{1:T}} \left[\sum_{t=2}^{T} \alpha_{1:t-1} \left(f(\bar{x}_{t-1}) - f(\bar{x}_t) - B_f(\bar{x}_{t-1}, \bar{x}_t) \right) \right] + \varepsilon_T \\
&= \frac{1}{\alpha_{1:T}} \sum_{t=2}^{T} \alpha_{1:t-1} \left(f(\bar{x}_{t-1}) - f(\bar{x}_t) \right) - \frac{1}{\alpha_{1:T}} \sum_{t=2}^{T} \alpha_{1:t-1} B_f(\bar{x}_{t-1}, \bar{x}_t) + \varepsilon_T \\
&= \underbrace{\frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} \alpha_t f(\bar{x}_t) - f(\bar{x}_T) + \varepsilon_T}_{:=0} - \frac{1}{\alpha_{1:T}} \sum_{t=2}^{T} \underbrace{\alpha_{1:t-1} B_f(\bar{x}_{t-1}, \bar{x}_t)}_{:\bar{B}_t^f}
\end{aligned}
\tag{F.29}
$$

Let us now evaluate $\tilde{\Delta}_T$. We have :

$$
\begin{aligned}
\tilde{\Delta}_T &= \frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), \mathbf{x}_t - x_t \rangle + \varepsilon_T \\
&\overset{(F.21)}{=} \frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} \langle f'(\mathbf{x}_t), \alpha_{1:t-1}(\mathbf{x}_{t-1} - \mathbf{x}_t) + p_t(\tilde{x}_t - \mathbf{x}_t) \rangle + \varepsilon_T \\
&= \underbrace{\frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} \langle f'(\mathbf{x}_t), \alpha_{1:t-1}(\mathbf{x}_{t-1} - \mathbf{x}_t) \rangle + \varepsilon_T}_{:=\Delta_T} + \frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} p_t \langle f'(\mathbf{x}_t), \tilde{x}_t - \mathbf{x}_t \rangle \\
&= \frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} p_t \langle f'(\mathbf{x}_t), \tilde{x}_t - \mathbf{x}_t \rangle - \frac{1}{\alpha_{1:T}} \sum_{t=2}^{T} \bar{B}_t^f \\
&= \frac{1}{\alpha_{1:T}} \sum_{t=1}^{T} p_t (f(\tilde{x}_t) - f(\mathbf{x}_t) - B_f(\tilde{x}_t, \mathbf{x}_t)) - \frac{1}{\alpha_{1:T}} \sum_{t=2}^{T} \bar{B}_t^f
\end{aligned}
\tag{F.30}
$$

The second last equation comes directly from Equation F.29. Hence, finally we have:

$$f(\mathbf{x}_T) - f^\star = \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), x_t - x^\star \rangle - B_{1:T} - \sum_{t=2}^{T} \bar{B}_t^f + \sum_{t=1}^{T} p_t (f(\tilde{x}_t) - f(\mathbf{x}_t) - B_f(\tilde{x}_t, \mathbf{x}_t)) \right] \tag{F.31}$$

□

The following variance bound is standard in the literature (e.g., Allen-Zhu 6, Lemma 2.4). For completeness, we provide a proof using our assumptions and notation.

Lemma F.10.0.1. *Fix $t \geq 1$ and assume that (F.18) holds, and g_t is given by (F.17). Further assume that for all ζ, $F(\cdot, \zeta)$ is convex and L-smooth w.r.t. the 2-norm $\|\cdot\|$ over $\mathbb{R}^d$ or otherwise satisfies (F.2) for all $x, y \in \mathcal{X}$. Then,*

$$\mathbb{E}[\|g_t - f'(\mathbf{x}_t)\|^2] \leq \mathbb{E}\|F'(\bar{x}_t, \zeta_t) - F'(\tilde{x}_t, \zeta_t)\|^2 \leq 2L\mathbb{E}B_f(\tilde{x}_t, \mathbf{x}_t)\,.$$

Proof. By the smoothness assumption on $F(\cdot, \zeta)$, which implies (F.2), we have

$$\|F'(x, \zeta) - F'(x', \zeta)\|^2 \leq 2L \left(F(x, \zeta) - F(x', \zeta) - \langle F'(x', \zeta), x - x' \rangle \right), \tag{F.32}$$

for all $x, x' \in \mathcal{X}$. Now, thanks to $\mathbb{E}\|U - \mathbb{E}U\|^2 \leq \mathbb{E}\|U\|^2$ which holds for any random vector U, and noticing that $\tilde{x}_t$ and $\mathbf{x}_t$ are determined by $\mathcal{H}_t$ by construction, we have

$$\begin{aligned}
\mathbb{E}\|g_t - f'(\mathbf{x}_t)\|^2 \mid \mathcal{H}_t &= \mathbb{E}\|F'(\mathbf{x}_t, \zeta_t) - f'(\mathbf{x}_t) - (F'(\tilde{x}_t, \zeta_t) - f'(\tilde{x}_t))\|^2 \mid \mathcal{H}_t \\
&\leq \mathbb{E}\|F'(\mathbf{x}_t, \zeta_t) - F'(\tilde{x}_t, \zeta_t)\|^2 \mid \mathcal{H}_t \\
&\leq 2L\mathbb{E}F(\tilde{x}_t, \zeta_t) - F(\mathbf{x}_t, \zeta_t) - \langle F'(\mathbf{x}_t, \zeta_t), \tilde{x}_t - \mathbf{x}_t \rangle \mid \mathcal{H}_t \qquad \text{(by (F.32))} \\
&= 2L \left[f(\tilde{x}_t) - f(\mathbf{x}_t) - \langle f'(\mathbf{x}_t), \tilde{x}_t - \mathbf{x}_t \rangle \right].
\end{aligned}$$

Taking the expectation of both sides finishes the proof. □

Proof of Theorem F.5.2.2. Note that $B_f(\tilde{x}_1, \mathbf{x}_1) = 0$. Also since, $\phi = 0$, $\ell(x) = f(x)$ for all x, hence we will be using f instead of ℓ in the proof. Let $s(t)$ represents the epoch containing iteration t. From Lemma F.5.2.1, we have :

$$f(\mathbf{x}_T) - f^\star = \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \langle \alpha_t f'(\mathbf{x}_t), x_t - x^\star \rangle - B_{1:T} - \sum_{t=2}^{T} \bar{B}_t^f + \sum_{t=1}^{T} p_t (f(\tilde{x}_t) - f(\mathbf{x}_t) - B_f(\tilde{x}_t, \mathbf{x}_t)) \right]. \tag{F.33}$$

Now we have:

$$\frac{1}{\alpha_{1:T}}\left[\sum_{t=1}^{T}\langle \alpha_t f'(\mathbf{x}_t), x_t - x^\star\rangle\right] = \frac{1}{\alpha_{1:T}}\left[\sum_{t=1}^{T}\langle \alpha_t f'(\mathbf{x}_t) - \alpha_t g_t + \alpha_t g_t, x_t - x^\star\rangle\right]$$
$$= \frac{1}{\alpha_{1:T}}\left[\sum_{t=1}^{T}\langle \alpha_t g_t, x_t - x^\star\rangle + \delta_{1:T}\right],$$

where $\delta_t = \alpha_t\langle f'(\bar{x}_t) - g_t, x_t - x^\star\rangle$. By (F.18), we have $\mathbb{E}[\delta_t] = 0$ for all $t \in [T]$. By theorem F.11.0.1 in the appendix,

$$\frac{1}{\alpha_{1:T}}\mathbb{E}\left[\sum_{t=1}^{T}\langle \alpha_t g_t, x_t - x^\star\rangle\right] \le \mathcal{R}_T(x^\star) = \frac{\eta_T}{2}\|x^\star\|_2^2 + \sum_{t=1}^{T}\frac{\alpha_t^2}{2\eta_t}\mathbb{E}\left[[]\,\|g_t - \tilde{g}_t\|_2^2\right].$$

Putting back in (F.33), and using the fact that due to convexity of f, $B_{1:T} \ge 0$, we have:

$$\mathbb{E}f(\bar{x}_T) - f(x^\star) \le \frac{1}{\alpha_{1:T}}\mathbb{E}\frac{\eta_T}{2}\|x^\star\|_2^2 + \sum_{t=1}^{T}\frac{\alpha_t^2}{2\eta_t}\|g_t - \tilde{g}_t\|_2^2 - \bar{B}^f_{2:T} + \sum_{t=1}^{T} p_t\left(f(\tilde{x}_t) - f(\bar{x}_t) - B_f(\tilde{x}_t, \bar{x}_t)\right). \tag{F.34}$$

Next, using $\tilde{g}_t = g_{t-1}, t > 1$, and applying theorem F.10.0.1, we have

$$\begin{aligned}
\sum_{t=2}^{T}\frac{\alpha_t^2}{2\eta_t}\mathbb{E}\|g_t - \tilde{g}_t\|^2 &= \sum_{t=2}^{T}\frac{\alpha_t^2}{2\eta_t}\mathbb{E}\|g_t - f'(\bar{x}_t) + f'(\bar{x}_{t-1}) - g_{t-1} + f'(\bar{x}_t) - f'(\bar{x}_{t-1})\|^2 \\
&\le \sum_{t=2}^{T}\frac{3\alpha_t^2}{2\eta_t}\mathbb{E}\|g_t - f'(\bar{x}_t)\|^2 + \|f'(\bar{x}_{t-1}) - g_{t-1}\|^2 + \|f'(\bar{x}_t) - f'(\bar{x}_{t-1})\|^2 \\
&\le \sum_{t=2}^{T}\frac{3\alpha_t^2}{2\eta_t}\mathbb{E}2LB_f(\tilde{x}_t, \bar{x}_t) + 2LB_f(\tilde{x}_{t-1}, \bar{x}_{t-1}) + 2LB_f(\bar{x}_{t-1}, \bar{x}_t) \\
&\le \sum_{t=1}^{T}\frac{3L(\alpha_t^2 + \alpha_{t+1}^2)}{\eta_t}\mathbb{E}B_f(\tilde{x}_t, \bar{x}_t) + \sum_{t=2}^{T}\frac{12L}{\eta_t}\mathbb{E}\bar{B}_t^f,
\end{aligned}$$

where the second step uses Jensen's inequality and the convexity of $\|\cdot\|^2$, the third step follows by the smoothness assumption on f and theorem F.10.0.1, and the fourth step follows by $\eta_{t+1} \ge \eta_t$ and $\frac{\alpha_t^2}{\alpha_{1:t-1}} \le 4$. Putting back into (F.34) with $\tilde{g}_1 = 0$,

$$\mathbb{E}f(\bar{x}_T) - f(x^\star) \le \frac{1}{\alpha_{1:T}}\mathbb{E}\frac{\eta_T}{2}\|x^\star\|_2^2 + \frac{\alpha_1^2}{2\eta_1}\|g_1\|^2 + \underbrace{\sum_{t=1}^{T} p_t\left(f(\tilde{x}_t) - f(\bar{x}_t)\right)}_{\Gamma_T}$$

$$+\sum_{t=1}^{T}\left(\frac{3L(\alpha_t^2+\alpha_{t+1}^2)}{\eta_t}-p_t\right)\mathbb{E}B_f(\tilde{x}_t,\bar{x}_t)+\sum_{t=2}^{T}\left(\frac{12L}{\eta_t}-1\right)\mathbb{E}\bar{B}_t^f\,. \tag{F.35}$$

Noticing that $p_t \geq \frac{15L\alpha_t^2}{\eta_t} \geq \frac{3L(\alpha_t^2+\alpha_{t+1}^2)}{\eta_t}$ and $\eta_t \geq 12L$, the last two terms in (F.35) vanish.

In the rest of the proof, we will bound the term Γ_T and use induction to get the final convergence rate. Let us denote $\mathbb{E}[f(\bar{x}_t)-f^\star]$ with D_t and $\mathbb{E}[f(\tilde{x}_t)-f^\star]$ with $\tilde{D}_{s(t)}$. Let us also represent the snapshot point for epoch s by $\tilde{x}^s$. Hence, $\tilde{D}(s)=\mathbb{E}[f(\tilde{x}_s)]-f^\star$. We also assume that $T_{1:S}<T\leq T_{1:S+1}$. Then

$$\begin{aligned}\sum_{t=1}^{T}p_t(f(\tilde{x}_t)-f(\bar{x}_t)) &= \sum_{t=1}^{T}p_t(\tilde{D}_{s(t)}-D_t)\\ &\leq p_1\tilde{D}(1)+\sum_{s=2}^{S+1}\tilde{D}(s)\left(\sum_{t=T_{1:s-1}+1}^{T_{1:s}}p_t\right)-\sum_{s=1}^{S}\left(\sum_{t=T_{1:s-1}+1}^{T_{1:s}}p_tD_t\right)\\ &\quad .\end{aligned} \tag{F.36}$$

From the definition of $\tilde{x}_s$, we can apply Jensen's inequality to get

$$\sum_{t=1}^{T}p_t(f(\tilde{x}_t)-f(\bar{x}_t))\leq p_1\tilde{D}(1)+\sum_{s=1}^{S}\left(\sum_{t=T_{1:s-1}+1}^{T_{1:s}}p_tD_t\right)\underbrace{\left(\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}}p_t}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}}p_t}-1\right)}_{:=\Theta_s}. \tag{F.37}$$

Let us assume that for $p_t=\frac{\alpha_t^2}{\gamma_t T_{s(t)}}\;\forall\; t\in[T_{1:s-1}+1,T_{1:s}]$ where $\alpha_t=t$ and γ_t is an increasing sequence of positive numbers. We rewrite the term Θ_s.

$$\Theta_s=\left(\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}}p_t}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}}p_t}-1\right)\leq\left(\frac{T_s}{T_{s+1}}\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}}t^2/\gamma_t}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}}t^2/\gamma_t}-1\right)\leq\left(\frac{T_s}{T_{s+1}}\frac{(\gamma_{T_{1:s}})^{-1}\sum_{t=T_{1:s}+1}^{T_{1:s+1}}t^2}{(\gamma_{T_{1:s}})^{-1}\sum_{t=T_{1:s-1}+1}^{T_{1:s}}t^2}-1\right).$$

We let s' be the smallest index for which $T_s=\tau$ for all $s>s'$. Hence, for $s>s'$ we have following bound for Θ_s:

$$\begin{aligned}\Theta_s &\leq \left(\frac{T_s}{T_{s+1}}\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}}t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}}t^2}-1\right)=\left(\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}}t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}}t^2}-1\right)=\left(\frac{\sum_{t=T_{1:s}+1}^{T_{1:s}+\tau}t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}t^2}-1\right)\\ &=\left(\frac{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}(t+\tau)^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}t^2}-1\right)=\left(\frac{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}(t+\tau)^2-t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}t^2}\right)=\left(\frac{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}\tau(2t+\tau)}{\sum_{t=T_{1:s-1}+1}^{T_{1:s-1}+\tau}t^2}\right)\end{aligned}$$

$$\leq \frac{\tau^2(2T_{1:s-1}+3\tau)}{(T_{1:s-1}+1)^2\tau} \leq \frac{\tau(2T_{1:s-1}+3(T_{1:s-1}+1))}{(T_{1:s-1}+1)^2} \leq 5\frac{\tau}{T_{1:s-1}+1} = 5\frac{T_s}{T_{1:s-1}+1},$$

using in the last step the fact that $s' < s$ implies $\tau = T_s \leq T_{1:s-1}+1$ (which follows because if $T_{s-1} = 2^{s-2}$, then $T_{1:s-1}+1 = 2^{s-1} \geq T_s$, and is trivial if otherwise $T_{s-1} = \tau$). Now let us consider the case when $s \leq s'$. In that case, $T_{s+1} \leq 2^s = 2T_s$, which can be used to show $\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}} t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}} t^2} \leq 2\left(\frac{T_{1:s+1}}{T_{1:s-1}+1}\right)^2 \leq 2\left(\frac{T_{1:s}+2T_s}{T_{1:s-1}+1}\right)^2 \leq 2\left(\frac{2^{s+1}-1}{2^{s-1}}\right)^2 \leq 32$, so

$$\Theta_s \leq \left(\frac{T_s}{T_{s+1}}\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}} t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}} t^2} - 1\right) \leq \left(\frac{\sum_{t=T_{1:s}+1}^{T_{1:s+1}} t^2}{\sum_{t=T_{1:s-1}+1}^{T_{1:s}} t^2} - 1\right) \leq 31.$$

Hence,

$$\begin{aligned} D_T &\leq \frac{1}{\alpha_{1:T}}\left[\frac{\eta_T}{2}\|x^\star\|^2 + \frac{1}{2\eta_1}\|f'(x_1)\|^2 + p_1\tilde{D}(1) + \sum_{s=1}^{S}\Theta_s \sum_{t=T_{1:s-1}+1}^{T_{1:s}} \frac{\alpha_t^2}{\gamma_t T_s} D_t\right] \\ &\leq \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \sum_{s=1}^{S}\Theta_s \sum_{t=T_{1:s-1}+1}^{T_{1:s}} \frac{2t^2}{\gamma_t T_s} D_t\right]. \end{aligned} \tag{F.38}$$

In the above equation, we can choose $\gamma_t = 124\log(2t)$ then we have:

$$D_T \leq \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\hat{\eta}}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \sum_{s=1}^{S}\Theta_s \sum_{t=T_{1:s-1}+1}^{T_{1:s}} \frac{2t^2}{124\log(2t)\ T_s} D_t\right]. \tag{F.39}$$

Now, it is important to remember that $p_t \geq \frac{15L\alpha_t^2}{\eta_t}$, which is satisfied if $\eta_t = 1860LT_{s(t)}\log(2t)$. We now use (F.39) to prove the following statement by induction:

$$D_T \leq \frac{T_{s(T)}\log(2T)}{T^2}\left[3720L\|x^\star\|^2 + \frac{2}{\eta_1}\|f'(x_1)\|^2 + 4p_1\tilde{D}(1)\right] = \frac{CT_{s(T)}\log(2T)}{T^2},$$

where $C = 3720L\|x^\star\|^2 + \frac{2}{\eta_1}\|f'(x_1)\|^2 + 4p_1\tilde{D}(1)$.

For $T = 1$, the bound is satisfied trivially, since $C \geq 2D_1$ by (F.35). Next, let $T > 1$, and assume the induction hypothesis holds for all $t \leq T-1$. Then, we have:

$$\begin{aligned} D_T &\leq \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \sum_{s=1}^{S}\Theta_s \sum_{t=T_{1:s-1}+1}^{T_{1:s}} \frac{2t^2}{124\log(2t)\ T_s} D_t\right] \\ &\leq \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \sum_{s=1}^{S}\Theta_s \sum_{t=T_{1:s-1}+1}^{T_{1:s}} \frac{2}{124} C\right] \end{aligned}$$

$$= \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \sum_{s=1}^{S}\Theta_s T_s\frac{2}{124}C\right],$$

where the second step uses the induction hypothesis on D_t for $t < T$. Then, on the one hand, if $S \le s'$, we have:

$$\begin{aligned} D_T &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + 31(\log(T_S)+1)\frac{2T_S}{124}C\right] \\ &= \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \log(2T)\frac{T_S}{2}C\right], \end{aligned}$$

where in the first step we use $T_s \le T_S$, the earlier bound $\Theta_s \le 31$ for $s \le s'$, and $S = \log_2(2^{S-1})+1 = \log_2(T_S)+1$, and in the second step we use $1 = \log_2(2)$. On the other hand, if $S > s'$:

$$\begin{aligned} D_T &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \sum_{s=1}^{s'}\Theta_s\frac{2T_S}{124}C + \sum_{s=s'+1}^{S}\Theta_s\frac{2T_S}{124}C\right] \\ &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + 31(\log(\tau)+1)\frac{2T_S}{124}C + \sum_{s=s'+1}^{S}5\frac{\tau}{(s-s')\tau}\frac{2T_S}{124}C\right] \\ &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + 31\log(2\tau)\frac{2T_S}{124}C + 5\log(S-s'+1)\frac{2T_S}{124}C\right] \\ &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + 31\log(2\tau)\frac{2T_S}{124}C + 31\log(S-s')\frac{2T_S}{124}C\right] \\ &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \log(2T)\frac{T_S}{2}C\right]. \end{aligned}$$

In the derivations above, the second step follows because by the definition of s', $s' \le \log(\tau)+1$ and also $T_{1:s'}+1 = 2^{s'} \ge T_{s'+1} = \tau$. The third step uses $1 = \log_2(2)$ and $\sum_{s=s'+1}^{S} 1/(s-s') = \sum_{j=1}^{S-s'} 1/j \le \log(S-s'+1)$. The fourth step uses $\log(j+1) \le 6\log(j)$ for any $j > 1$. The fifth step uses $(S-s')\tau \le T_{1:S} < T$.

Hence, in both cases, we have

$$\begin{aligned} D_T &\le \frac{1}{T^2}\left[\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1) + \log(2T)\frac{T_S}{2}C\right] \\ &\le \frac{1}{T^2}\left[\underbrace{\eta_T\|x^\star\|^2 + \frac{1}{\eta_1}\|f'(x_1)\|^2 + 2p_1\tilde{D}(1)}_{\le \frac{CT_{s(T)}\log(2T)}{2}} + \frac{CT_{s(T)}\log(2T)}{2}\right] \end{aligned}$$

$$\leq \frac{CT_{s(T)}\log(2T)}{T^2},$$

using that $T_S \leq T_{S+1} = T_{s(T)}$. This concludes the proof. □

F.11 Regret bounds for online linear optimization

In this section, we provide the conditions and regret-bound $\mathcal{R}_T$ for vanilla AO-FTRL, which is used by theorems F.3.0.3 and F.4.2.1, as well as the SVRG results. The analysis is based on Joulani *et al.* [102, 103].

Theorem F.11.0.1. *Let $\phi : \mathcal{X} \to \mathbb{R}$ be a convex function. For $t \in [T]$, let $r_t : \mathcal{X} \to \mathbb{R}, \alpha_t > 0, g_t, \tilde{g}_t \in \mathbb{R}^d, x_t \in \mathcal{X}$ be such that the AO-FTRL update (F.4) is well-defined and x_t is given by (F.4) for all $t \in [T]$. Suppose that for all $t \in [T]$, r_{t-1} is convex and the objective in the AO-FTRL update (F.4) has finite value at the optimum x_t. Then, for any $x \in \mathcal{X}$,*

$$\sum_{t=1}^{T} \alpha_t \left(\langle g_t, x_t - x\rangle + \phi(x_t) - \phi(x)\right) \leq \sum_{t=1}^{T} \left(r_{t-1}(x) - r_{t-1}(x_t) - B_{\alpha_{1:t}\phi + r_{0:t-1}}(x_{t+1}, x_t)\right) + \sum_{t=1}^{T} \alpha_t \langle g_t - \tilde{g}_t, x_t - x_{t+1}\rangle . \tag{F.40}$$

If, in addition, for all $t \in [T]$, $B_{\alpha_{1:t}\phi + r_{0:t-1}}(x_{t+1}, x_t) \geq \frac{1}{2}\|\cdot\|^2_{(t)}$ for some norm $\|\cdot\|_{(t)}$, then

$$\sum_{t=1}^{T} \alpha_t \left(\langle g_t, x_t - x\rangle + \phi(x_t) - \phi(x)\right) \leq \sum_{t=1}^{T} \left(r_{t-1}(x) - r_{t-1}(x_t)\right) + \sum_{t=1}^{T} \frac{\alpha_t^2}{2}\|g_t - \tilde{g}_t\|^2_{(t)*} . \tag{F.41}$$

Proof. The assumption that ϕ and r_t are real-valued and defined on $\mathcal{X}$ ensures that they are proper, which, together with convexity ensures that $\alpha_{1:t}\phi + r_{0:t-1}$ is directionally differentiable [22, Prop. 17.2]. Together with the assumption that the AO-FTRL objective in (F.4) is finite-valued, we guarantee Assumption 1 of Joulani *et al.* [103], while their Assumption 5 is satisfied given that the combined linear-composite function $\alpha_t(\langle g_t, \cdot\rangle + \phi)$ is convex. The first bound (F.40) then follows from the intermediate bound C.1 in the proof of Theorem 6 of Joulani *et al.* [103], using $x^* \leftarrow x$, $g_t \leftarrow \alpha_t g_t$, $\tilde{g}_{T+1} \leftarrow 0$, $p_t \leftarrow 0, t \in [T]$, $\tilde{q}_t \leftarrow r_t + \alpha_{t+1}\phi, t \in \{0\} \cup [T-1]$, and $\tilde{q}_T \leftarrow 0$. The second bound (F.41) follows from the statement of Theorem 6 of Joulani *et al.* [103], using using $x^* \leftarrow x$, $p_t \leftarrow 0, t \in [T]$, and $\tilde{q}_t \leftarrow r_t + \alpha_{t+1}\phi, t \in \{0\} \cup [T-1]$, noting that Assumption 8 of Joulani *et al.* [103] is the extra condition we have assumed in the second part of the theorem. □

Appendix G

Importance Sampling via Local Sensitivit

Anant Raj[1], Cameron Musco[2], Lester Mackey[3]

1 – MPI for Intelligent Systems, Tübingen

2 – UMass Amherst, Massachusetts

3 – Microsoft Research, New England

Abstract

Given a loss function $F : \mathcal{X} \to \mathbb{R}^+$ that can be written as the sum of losses over a large set of inputs $a_1, \ldots, a_n$, it is often desirable to approximate F by subsampling the input points. Strong theoretical guarantees require taking into account the importance of each point, measured by how much its individual loss contributes to $F(x)$. Maximizing this importance over all $x \in \mathcal{X}$ yields the *sensitivity score* of a_i. Sampling with probabilities proportional to these scores gives strong guarantees, allowing one to approximately minimize of F using just the subsampled points.

Unfortunately, sensitivity sampling is difficult to apply since (1) it is unclear how to efficiently compute the sensitivity scores and (2) the sample size required is often impractically large. To overcome both obstacles we introduce *local sensitivity*, which measures data point importance in a ball around some center x_0. We show that the local sensitivity can be efficiently estimated using the *leverage scores* of a quadratic approximation to F and that the sample size required to approximate F around x_0 can be bounded. We propose employing local sensitivity sampling in an iterative optimization method and analyze its convergence when F is smooth and convex.

G.1 Introduction

In this work we consider finite sum minimization problems of the following form.

Definition G.1.0.1 (Finite Sum Problem)**.** Given data points $a_1, \ldots, a_n \in \mathbb{R}^d$, nonnegative functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}^+$, and a nonnegative function $\gamma : \mathbb{R}^d \to \mathbb{R}^+$, minimize over $x \in \mathcal{X} \subseteq \mathbb{R}^d$

$$F(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^T x) + \gamma(x). \tag{G.1}$$

Definition G.1.0.1 captures a number of important problems, including penalized empirical risk minimization (ERM) for linear regression, generalized linear models, and support vector machines. When n is large, minimizing $F(x)$ can be expensive. In some cases, for example, it may be impossible to load the full dataset $a_1, \ldots, a_n$ into memory.

G.1.1 Function Approximation via Data Subsampling

To reduce the burden of solving a finite sum problem, one commonly minimizes an approximation to F formed by independently subsampling data points a_i (and hence summands $f_i(a_i^T x)$) with some fixed probability weights. More formally:

Definition G.1.1.1 (Subsampled Finite Sum Problem)**.** Consider the setting of Definition G.1.0.1. Given a target sample size m and a probability distribution $P = \{p_1, \ldots, p_n\}$ over $[n] \triangleq \{1, \ldots, n\}$, select $i_1, ..., i_m$ i.i.d. from P and minimize over $x \in \mathcal{X} \subseteq \mathbb{R}^d$

$$F^{(P,m)}(x) := \frac{1}{mn} \sum_{j=1}^{m} \frac{f_{i_j}(a_{i_j}^T x)}{p_{i_j}} + \gamma(x). \tag{G.2}$$

We can see that for any x, $\mathbb{E}[F^{(P,m)}(x)] = F(x)$. If the sampled function concentrates well around $F(x)$, then it can serve effectively as a surrogate for minimizing F. Most commonly, P is set to the uniform distribution. Unfortunately, if $F(x)$ is dominated by the values of a relatively few large $f_i(a_i^T x)$, unless m is very large, uniform subsampling will miss these important data points and $F^{(P,m)}(x)$ will often underestimate $F(x)$. This can happen, for example, when $a_1, ..., a_n$ fall into clusters of non-uniform size. Data points in smaller clusters are important in selecting an optimal x but are often underrepresented in a uniform sample.

G.1.2 Importance Sampling via Sensitivity

A remedy to the weakness of uniform subsampling is to apply importance sampling: preferentially sample the functions $f_i(a_i^\top x)$ that contribute most significantly to $F(x)$. If,

for example, we set $p_i \propto \frac{f_i(a_i^T x)}{\sum_{i=1}^n f_i(a_i^T x)+\gamma(x)}$ for each $i \in [n]$, then a standard concentration argument would imply that $(1-\varepsilon)F(x) \leq F^{(P,m)}(x) \leq (1+\varepsilon)F(x)$ with probability at least $1-\delta$ if $m = \Theta\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$. However, typically the relative the importance of each point, $\frac{f_i(a_i^T x)}{\sum_{i=1}^n f_i(a_i^T x)+\gamma(x)}$, will depend on the choice of x. This motivates the definition of *sensitivity* [126].

Definition G.1.2.1 (Sensitivity)**.** For $a_1, \ldots, a_n \in \mathbb{R}^d$, the *sensitivity* of point a_i with respect to a finite sum function F (Definition G.1.0.1) with domain $\mathcal{X} \subseteq \mathbb{R}^d$ is

$$\sigma_{F,\mathcal{X}}(a_i) = \sup_{x \in \mathcal{X}} \frac{f_i(a_i^T x)}{\sum_{j=1}^n f_j(a_j^T x) + n\gamma(x)}.$$

The *total sensitivity* is defined as $\mathcal{G}_{F,\mathcal{X}} = \sum_{i=1}^n \sigma_{F,\mathcal{X}}(a_i)$.

A standard concentration argument yields the following approximation guarantee for sensitivity sampling.

Lemma G.1.2.1. *Consider the setting of Definition G.1.0.1. For all $i \in [n]$, let $s_i \geq \sigma_{F,\mathcal{X}}(a_i)$, $S = \sum_{i=1}^n s_i$, and $P = \left\{\frac{s_1}{S}, \ldots, \frac{s_n}{S}\right\}$. There is a fixed constant c such that, for any $\varepsilon, \delta \in (0,1)$, any fixed $x \in \mathcal{X}$, and $m \geq \frac{c \cdot S \log(2/\delta)}{\varepsilon^2}$,*

$$(1-\varepsilon)F(x) \leq F^{(P,m)}(x) \leq (1+\varepsilon)F(x)$$

with probability $\geq 1-\delta$.

That is, subsampling data points by their sensitivities approximately preserves the value of F *for any fixed* $x \in \mathcal{X}$ with high probability. It can thus be argued that F can be approximately minimized by minimizing the sampled function $F^{(P,m)}$. We first define:

Definition G.1.2.2 (Range Space)**.** A range space is a pair $\mathcal{R} = (\mathcal{F}, \text{ranges})$, where $\mathcal{F}$ is a set and ranges is a set of subsets of $\mathcal{F}$. The VC dimension $\Delta(\mathcal{R})$ is the size of the largest $G \subseteq \mathcal{F}$ such that G is shattered by ranges: i.e., $|\{G \cap R | R \in \text{ranges}\}| = 2^{|G|}$.

Let $\mathcal{F}$ be a finite set of functions mapping $\mathbb{R}^d \to \mathbb{R}^+$. For every $x \in \mathbb{R}^d$ and $r \in \mathbb{R}^+$, let $\text{range}_{\mathcal{F}}(x,r) = \{f \in \mathcal{F} | f(x) \geq r\}$ and $\text{ranges}(\mathcal{F}) = \{\text{range}_{\mathcal{F}}(x,r) | x \in \mathbb{R}^d, r \in \mathbb{R}^+\}$. We say $R_{\mathcal{F}} = (\mathcal{F}, \text{ranges}(\mathcal{F}))$ is the range space induced by $\mathcal{F}$.

With the notion of range space in place, we can recall the following general approximation theorem.

Theorem G.1.2.2 (Theorem 9 [159])**.** *Consider the setting of Definition G.1.0.1. For all $i \in [n]$, let $s_i \geq \sigma_{F,\mathcal{X}}(a_i)$, $S = \sum_{i=1}^n s_i$, and $P = \left\{\frac{s_1}{S}, \ldots, \frac{s_n}{S}\right\}$. For some finite c and all $\varepsilon, \delta \in (0, 1/2)$, if*

$$m \geq c \cdot \frac{S}{\varepsilon^2}\left(\Delta \log S + \log\left(\frac{1}{\delta}\right)\right),$$

then, with probability at least $1-\delta$,

$$(1-\varepsilon)F(x) \leq F^{(P,m)}(x) \leq (1+\varepsilon)F(x), \forall x \in \mathcal{X}$$

Here, Δ *is an upper bound on the VC-dimension* $\Delta(\mathcal{R}_{\mathcal{F}})$ *where* $\mathcal{F}$ *is the set* $\left\{\frac{f_1(a_1^T x)}{mn \cdot p_1}, \ldots, \frac{f_n(a_n^T x)}{mn \cdot p_n}\right\}$.

Munteanu *et al.* [159] show that $\Delta = d+1$ suffices for logistic regression where d is the dimension of the input points. If all f_i are from the class of invertible functions, then a similar bound on Δ can be expected.

Barriers to the Sensitivity Sampling in Practice

Theorem G.1.2.2 is quite powerful: it can be used to achieve sensitivity-sampling-based approximation algorithms with provable guarantees for a wide range of problems [65, 95, 146, 159]. However, there are two major barriers that have hindered more widespread practical adoption of sensitivity sampling:

Computability: It is difficult to compute or even approximate the sensitivity $\sigma_{F,\mathcal{X}}(a_i)$ since it is not clear how to take the supremum over all $x \in \mathcal{X}$ in the expression of Definition G.1.2.1. Closed form expressions for the sensitivity are known only in a few special cases, such as least squares regression (where the sensitivity is closely related to the well-studied *statistical leverage scores*).

Pessimistic Bounds: The sensitivity score is a very 'worst case' importance metric, since it considers the supremum of $\frac{f_i(a_i^T x)}{\sum_{j=1}^n f_j(a_j^T x)+n\gamma(x)}$ over all $x \in \mathcal{X}$, including, e.g., x that may be very far from the true minimizer of F. In many cases, it is possible to construct, for each a_i, some worst case x that forces this ratio to be high. Thus, all sensitivities are large and the total sensitivity $\mathcal{G}_{F,\mathcal{X}}$ is large. The sample complexities in Lemma G.1.2.1 and Theorem G.1.2.2 depend on $S \geq \mathcal{G}_{F,\mathcal{X}}$ and so will be too large to be useful in practice. See Figure G.1 for a simple example of when this issue can arise.

G.1.3 Our Approach: Local Sensitivity

We propose to overcome the above barriers via a simple idea: *local sensitivity*. Instead of sampling with the sensitivity over the full domain $\mathcal{X}$ as in Definition G.1.2.1, we consider the sensitivity over a small ball. Specifically, for some radius r and center y we let $B(r,y) = \{x \in \mathbb{R}^d : \|x-y\| < r\}$ and consider $\sigma_{F,\mathcal{X}\cap B(r,y)}(a_i)$. Sampling by this local sensitivity will give us a function $F^{(P,m)}$ that *approximates* F *well on the entire ball* $B(r,y)$. Thus, we can approximately minimize F on this ball. We can approximately minimize F globally via an iterative scheme: at each step we set x_i to the approximate optimum of F over the ball $B(r_i, x_{i-1})$ (computed via local sensitivity sampling). This approach has two major advantages:

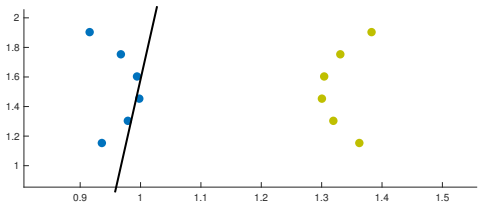


Figure G.1: Consider a classification problem with two classes A_1, A_2, shown in blue and green. Let $f_i(a_i^T x)$ be any loss function with $f_i(a_i^T x) = 0$ if a_i is correctly classified by the hyperplane defined by x. Since for each a_i, there is some x (e.g., corresponding to the black line shown) that misclassifies *only* a_i, we have $\sigma_{\mathcal{F},\mathbb{R}^d}(a_i) = 1$ for all a_i. Thus, the total sensitivity is $\mathcal{G}_{F,\mathcal{X}} = n$ and so the sampling results of Lemma G.1.2.1 and Theorem G.1.2.2 are vacuous – they require sampling $\geq n$ points, even for this simple task.

1. We can often locally approximate each F by a simple function, for which we can compute the local sensitivities in closed form. This will yield an approximation to the true local sensitivities. Specifically, we will consider a local quadratic approximation to F, whose sensitivities are given by the *leverage scores* of an appropriate matrix.
2. By definition, the local sensitivity $\sigma_{F,\mathcal{X}\cap B(r,y)}$ is *always* upper bounded by the global sensitivity $\sigma_{F,\mathcal{X}}$, and typically the sum of local sensitivities will be much smaller than the total sensitivity $\mathcal{G}_{F,\mathcal{X}}$. This allows us to take fewer samples to approximately minimize F locally over $B(r,y)$.

G.1.4 Related Work

The sensitivity sampling framework has been successfully applied to a number of problems, including clustering [18, 65, 146], logistic regression [95, 159], and least squares regression, in the form of leverage score sampling [45, 58, 148]. In these works, upper bounds are given on the sensitivity of each data point, and it is shown that the sum of these bounds, and thus the required sample size for approximate optimization, is small. We aim to expand the applicability of sensitivity-based methods to functions for which a bound on the sensitivity cannot be obtained or for which the total sensitivity is inherently large.

The local-sensitivity-based iterative method that we will discuss is closely related to quasi-Newton methods [56], especially those that approximate the Hessian via leverage score sampling [267, 269]. In each iteration, we estimate local sensitivities by considering the sensitivities of a local quadratic approximation to F. As shown in Section G.2, these sensitivities can be bounded using the leverage scores of the Hessian, and thus our sampling probabilities are closely related to those used in the above works. Unlike a quasi-Newton method however, we use the sensitivities to directly optimize F locally, rather than the quadratic approximation itself. In this way, our method is closer to a trust region method [40] or an approximate proximal point method [73].

Recently, [2] and [43] have suggested iterative algorithms for regularized least squares regression and ERM for linear models that sample a subset of data points by their leverage scores (closely related to sensitivities) in each step. These works employ this sampling in a different way than us, using the subsample to precondition each iterative step. While they give strong theoretical guarantees for the problems studied, this technique applies to a less general class of problems than our method.

The sensitivity scores for ℓ_2 regression are commonly known as leverage scores, and a long line of work [14, 208, see, e.g.,] has focused on approximating these scores more quickly. These approximation techniques do not extend to general sensitivity score approximation however. Additionally, our paper in no way attempts to develop a faster algorithm for leverage score sampling. We focus on introducing the notion of local sensitivity, which allows leverage score based methods to be applied to optimization problems well beyond ℓ_2 regression.

G.1.5 Road Map

Our contributions are presented as follows. In Section G.2 we show that the sensitivity scores of a quadratic approximation to a function are given by the leverage scores of an appropriate matrix. We use these scores to bound the local sensitivity scores of the true function. In Section G.3 we discuss how to subsample using these approximate local sensitivities with the aim of approximately minimizing the function over a small ball. We describe how to use this approach to iteratively optimize the function. In Section G.4 we give an analysis of this iterative method for convex functions.

G.2 Leverage Scores as Sensitivities of Quadratic Functions

We start by showing how to approximate the local sensitivity $\sigma_{F,\mathcal{X}\cap B(r,y)}$ over some ball by approximating F with a quadratic function on this ball. F's sensitivities can be approximated by those of this quadratic function, which we in turn bound in closed form by the leverage scores of an appropriate matrix (a rank-1 perturbation of F's Hessian at y). The leverage scores are given by:

Definition G.2.0.1 (Leverage Scores [5, 46])**.** For any $C \in \mathbb{R}^{n\times p}$ with i^{th} row c_i, the i^{th} λ-ridge leverage score is the sensitivity of $F(z) = \|Cz\|_2^2 + \lambda\|z\|_2^2$:

$$\ell_i^\lambda(C) := \max_{\{z\in\mathbb{R}^p:\|z\|_2>0\}} \frac{[Cz]_i^2}{\|Cz\|_2^2 + \lambda\|z\|_2^2}.$$

We have $\ell_i^\lambda(C) = c_i^T(C^TC+\lambda I)^{-1}c_i$. (See Lemma G.8.0.1 in Appendix G.8).

Our eventual iterative method will employ a proximal function, and thus in this section we consider this function, which reduces to F when $\lambda = 0$:

Definition G.2.0.2 (Proximal Function)**.** For a function $F : \mathcal{X} \to \mathbb{R}$, define $F_{\lambda,y}(x) = F(x) + \lambda \|x-y\|_2^2$.

Using Definition G.2.0.1 and the associated Lemma G.8.0.1 we establish the following in Appendix G.8.

Theorem G.2.0.1 (Sensitivity of Quadratic Approximation)**.** *Consider F as in Def. G.1.0.1 along with the quadratic approximation to the proximal function $F_{\lambda,y}$ (Def. G.2.0.2) around $y \in \mathcal{X}$. If $A \in \mathbb{R}^{n\times d}$ is the data matrix with i^{th} row equal to a_i, then*

$$\begin{aligned}\tilde{F}_{\lambda,y}(x) &:= \frac{1}{n}\sum_{i=1}^{n}\left[f_i(a_i^T y) + a_i^T(x-y)\cdot f'(a_i^T y) + \frac{1}{2}(a_i^T(x-y))^2 \cdot f''(a_i^T y)\right] + \gamma(x) + \lambda\|x-y\|_2^2 \\ &:= F(y) + (x-y)^T A^T \alpha_y + \frac{1}{2}(x-y)^T A^T H_y A(x-y) + \gamma(x) + \lambda\|x-y\|_2^2\end{aligned} \tag{G.3}$$

where $[\alpha_y]_i = \frac{1}{n} f_i'(a_i^T y)$, and H_y is the diagonal matrix with $[H_y]_{i,i} = \frac{1}{n} f''(a_i^T y)$. Assuming that H_y is nonnegative, the sensitivity scores of $\tilde{F}_{\lambda,y}$ with respect to $B(r,y)$ can be bounded as

$$\sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) \le \beta \cdot \ell_i^{\lambda}(C) + \frac{f_i(a_i^T y)}{\eta}, \tag{G.4}$$

where $C = [H_y^{1/2}A, \frac{1}{\delta}H_y^{-1/2}\alpha_y]$, $\ell_i^{\lambda}(C)$ is the leverage score of Def. G.2.0.1, $\eta = \min_{x\in B(r,y)} \tilde{F}_{\lambda,y}(x)$, $\delta = \min_{x\in B(r,y)} \gamma(x)$, and $\beta = \max\left(1, 1 - \frac{F(y) - \frac{1}{n}\sum_{i=1}^{n} \frac{f'(a_i^T y)^2}{4f''(a_i^T y)}}{\eta}\right)$.

Note that if we consider a small enough ball, where $\tilde{F}_{\lambda,y}$ well approximates $F_{\lambda,y}$, we expect $\eta = \min_{x\in B(r,y)} \tilde{F}_{\lambda,y}(x) = \Theta(F(y))$. Thus, the additive $\frac{f_i(a_i^T y)}{\eta}$ term on each sensitivity will contribute only a $\frac{\sum f_i(a_i^T y)}{\Theta(F(y))} = O(1)$ additive factor to the total sensitivity bound and sample size.

G.2.1 Efficient Computation of Leverage Score Sensitivities

The sensitivity upper bound (G.4) of Theorem G.2.0.1 can be approximated efficiently as long as we can efficiently approximate the leverage scores $\ell_i^{\lambda}(C) = c_i^T(C^T C + \lambda I)^{-1} c_i$,

where $C = [H_y^{1/2}A, \frac{1}{\delta}H_y^{-1/2}\alpha_y]$. We can use a block matrix inversion formula to find that

$$(C^TC+\lambda I)^{-1} = \begin{bmatrix} A^TH_yA+\lambda I & \frac{1}{\delta}A^T\alpha_y \\ \frac{1}{\delta}\alpha_y^TA & \|\alpha_y\|_2^2+\lambda \end{bmatrix}^{-1} = \begin{bmatrix} A_1 & A_2 \\ A_2^\top & \frac{1}{k} \end{bmatrix}$$

where $A_1 = (A^TH_yA+\lambda I)^{-1} + \frac{1}{k}(A^TH_yA+\lambda I)^{-1}A^T\alpha_y\alpha_y^TA(A^TH_yA+\lambda I)^{-1}$, $k = \|\alpha_y\|_2^2 + \delta^2\lambda - \alpha_y^TA(A^TH_yA+\lambda I)^{-1}A^T\alpha_y$, and $A_2 = -\frac{\delta}{k}(A^TH_yA+\lambda I)^{-1}A^T\alpha_y$.

Thus, if we have a fast algorithm for applying $(A^TH_yA+\lambda I)^{-1}$ to a vector we can quickly apply $(C^TC+\lambda I)^{-1}$ to a vector and compute the leverage scores $\ell_i^\lambda(C) = c_i^T(C^TC+\lambda I)^{-1}c_i$. Via standard Johnson-Lindenstrauss sketching techniques [226] it in fact suffices to apply this inverse to $O(\log n/\delta)$ vectors to approximate each score up to constant factor with probability $\geq 1-\delta$. In practice, one can use traditional iterative methods such as conjugate gradient, iterative sampling methods such as those presented in [45, 46], or fast sketching methods [44, 59].

G.2.2 True Local Sensitivity from Quadratic Approximation

As long as the quadratic approximation $\tilde{F}_{\lambda,y}$ approximates $F_{\lambda,y}$ sufficiently well on the ball $B(r,y)$, we can use Theorem G.2.0.1 to approximate the true local sensitivity $\sigma_{F_{\lambda,y},\mathcal{X}\cap B(r,y)}(a_i)$. We start by discussing our approximation assumptions. Defining α_y as in Theorem G.2.0.1, for some $B_y(x)$ which itself is a function of x we have:

$$F(x) = F(y) + (x-y)^\top A^\top\alpha_y + (x-y)^\top A^\top H_yA(x-y) + \gamma(x) + B_y(x)\|x-y\|_2^3.$$

Without loss of generality, we assume that $B_y(x) > 0$ for x in the above equation or we just shift the overall function vertically by adjusting $\gamma(\cdot)$ to have the quadratic appropriator be an under approximation of the true function. If the function F has a C Lipschitz-Hessian then we have:

$$F(x) \leq F(y) + (x-y)^\top A^\top\alpha_y + (x-y)^\top A^\top H_yA(x-y) + \gamma(x) + \frac{C}{6}\|x-y\|_2^3. \tag{G.5}$$

For simplicity, we also assume that (G.5) holds componentwise with Lipschitz Hessian constant C_i for $i \in [n]$. Adding the second order approximation of $F(x)$ to $\lambda\|x-y\|_2^2$ gives the approximate function $\tilde{F}_{\lambda,y}(x)$ as defined in (G.3). Theorem G.2.0.1 shows how to bound the sensitivities of $\tilde{F}_{\lambda,y}(x)$. Using (G.5) we prove a bound on the local sensitivities of $F_{\lambda,y}(x)$ itself in Appendix G.9:

Theorem G.2.2.1. *Consider $F_{\lambda,y}$ as in Defs. G.1.0.1, G.2.0.2, $y \in \mathcal{X}$, a radius r, and*

$\alpha = \min_{x \in B(r,y)} F_{\lambda,y}(x)$. *Then,* $\forall\, i \in [n]$,

$$\sigma_{F_{\lambda,y},B(r,y)}(a_i) \leq \sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) + \min\left(\frac{C_i r}{6n\lambda}, \frac{C_i r^3}{6n\alpha}\right).$$

Using this sensitivity bound, we can independently sample components with the computed scores as in Definition G.1.1.1, obtaining a $(1+\varepsilon)$ approximation of the function $F_{\lambda,y}(x)$. That is, letting $F^s_{\lambda,y}(x)$ represent the subsampled empirical loss function (sampled as in Theorem G.1.2.2), for $\tilde{O}\left(\frac{\Delta}{\varepsilon^2}\right)$ samples, we have $F^s_{\lambda,y}(x) \in (1 \pm \varepsilon) F_{\lambda,y}(x)\ \forall\ x \in B(y,R)$ with high probability.

G.3 Optimization via Local Sensitivity Sampling

In Theorem G.2.2.1 we showed how to bound the local sensitivities of a function $F := \sum_{i=1}^n f_i(a_i^T x) + \gamma(x)$ using the local sensitivities of a quadratic approximation to F, which are given by the leverage scores of an appropriate matrix (Theorem G.2.0.1). These sensitivities are only valid in a sufficiently small ball around some starting point y, roughly, where the quadratic approximation is accurate. In this section we show how they can be used to optimize F beyond this ball, specifically as part of an iterative method that locally optimizes F until convergence to a global optimum.

In the optimization literature, there are two popular techniques that iteratively optimize a function via local optimizations over a ball: (i) trust region methods [47] and (ii) proximal point methods [187]. Local sensitivity sampling can be combined with both of these classes of methods. We first focus on proximal point methods, discussing a related trust region approach in Section G.5. In the proximal point method, the idea is in each step to approximate a regularized minimum:

$$\begin{aligned} x^\star_{\lambda_t,y} = \arg\min F_{\lambda_t,y}(x) = \arg\min \left[F(x) + \lambda_t \|x-y\|_2^2\right] \\ \text{and } F^\star_{\lambda_t,y} = F_{\lambda_t,y}(x^\star_{\lambda_t,y}). \end{aligned} \tag{G.6}$$

Here λ_t is a regularization parameter depending on the iteration t. As discussed below, minimizing this regularized function is equivalent to minimizing F on a ball of a given radius.

G.3.1 Equivalence between Constrained and Penalized Formulation

When F is convex it is well known that for any λ minimizing the proximal function $F_{\lambda,y}$ is equivalent to minimizing F constrained to some ball around y. Consider the constrained optimization problem given in equation (G.7) where $B(r,y)$ is the ball of radius r centered

at y:

$$x^\star_{r,y} = \arg\min_{x \in B(r,y)} F(x). \tag{G.7}$$

Lemma G.3.1.1. *Let $x^\star = \arg\min_{x\in\mathbb{R}^d} F(x)$ for a convex function F. If $x^\star$ does not lie inside $B(r,y)$ then $x^\star_{r,y}$ also solves the following optimization problem:*

$$x^\star_{r,y} = \arg\min_{x\in\mathbb{R}^d} F(x) + \frac{\|\nabla F(x^\star_{r,y})\|}{2r} \cdot \|x-y\|_2^2. \tag{G.8}$$

Comparing equations (G.6) and (G.8), se see that $\lambda = \frac{\|\nabla F(x^\star_{r,y})\|}{2r} \Rightarrow r = \frac{\|\nabla F(x^\star_{r,y})\|}{2\lambda}$. While it is not directly possible to compute radius r in closed form without computing $x^\star_{r,y}$ itself, we can give a computable upper bound on r which will be crucial for our analysis.

Lemma G.3.1.2. *Consider the optimization problem (G.6) and its corresponding constrained counterpart (G.7) where F is a μ strongly convex function. Then, $x^\star_{\lambda,y}$ falls within a ball of radius $r = \frac{\|\nabla F(y)\|}{2\lambda+\mu}$ around y.*

Proofs for this sections are provided in the Appendix G.10.

Using the local sensitivity bound of Section G.2.2 we can approximate $F_{\lambda,y}$ on a ball of small enough radius. In applying sensitivity sampling to a proximal point method, it will be critical to ensure that λ_t is not too small. This will ensure that, by Lemma G.3.1.2, $x^\star_{\lambda_y}$ falls in a sufficiently small radius, and so an approximate minimum can be found via local sensitivity sampling.

G.3.2 Algorithmic Intuition

By Theorem G.1.2.2 if we subsample the proximal function $F_{\lambda_t,y}$ using the local sensitivity bound of Theorem G.2.2.1 for a sufficiently large radius r (as a function of λ_t via Lemma G.3.1.2), optimizing this function will return a value within a $1+\varepsilon$ factor of the true minimum $x^\star_{\lambda_t,y}$ with high probability. Abstracting away the sensitivity sampling technique, our goal becomes to analyze the convergence of the approximate proximal point method (APPM) when the optimum is computed up to $1+\varepsilon$ error in each iteration. We give pseudocode for this general method in Algorithm 19.

Algorithm 19 APPM

input $x_0 \in \mathbb{R}^d$, $\lambda_t > 0 \; \forall t \in [T]$.
input Black-box ε-oracle $\mathcal{P}_{F_{\lambda_1, x_0}}$
for $t = 1 \ldots$ T **do**
    $x_t \leftarrow P_{F_{\lambda_t, x_{t-1}}}(x)$
end for
output x_T

Definition G.3.2.1. An algorithm $\mathcal{P}_f$ is called *multiplicative* ε-oracle for a given function F if $F(x^\star) \leq F\left(\mathcal{P}_F(x)\right) \leq (1+\varepsilon)F(x^\star)$ where $x^\star$ if the true minimizer of F.

In Algorithm 19, we provide the pseudocode for APPM under the access of a *multiplicative* ε-oracle at each iterate. In our setting, $\mathcal{P}_F$ employs local sensitivity sampling.

G.4 Convergence Analysis for Smooth Convex Functions

In this section, we analyze the convergence of Algorithm 19 with an ε oracle obtained via local sensitivity sampling. We demonstrate how to set the regularization parameters λ_t in each step and then in the end provide a complete algorithm. Let $F^\star$ denote $F(x^\star)$. Throughout we make the following assumption about $F(x)$:

- F is μ-strongly convex, *i.e.*, for all $x, y \in \mathbb{R}^d$, $F(y) \geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\mu}{2}\|y - x\|_2^2$.

G.4.1 Approximate Proximal Point Method with Multiplicative Oracle

We first state convergence bounds for Approximate Proximal Point Method (Algorithm 19) with a blackbox multiplicative ε-oracle. Our first bound assumes strong convexity, our second does not. Proofs are given in Appendix G.11.

Theorem G.4.1.1. *For μ-strongly convex F, consider $\varepsilon_1, \ldots \varepsilon_T \in (0,1)$ and $x_0, \ldots, x_T \in \mathbb{R}^d$ such that $x_t = P_{F_{\lambda_t, x_{t-1}}}(x_{t-1})$ where $P_{F_{\lambda_t, x_{t-1}}}$ is an ε_t-oracle (see Algorithm 19). Then if $\varepsilon_t \leq \frac{\mu}{\mu + \lambda_t} \forall t \in [T]$, we have $F(x_t) - F^\star \leq \frac{1}{1-\varepsilon_t}\frac{\lambda_t}{\mu+\lambda_t}\left(F(x_{t-1}) - F^\star\right) + \frac{\varepsilon_t}{1-\varepsilon_t}F^\star \; \forall t \in [T]$ and*

$$F(x_T) - F^\star \leq \rho\left(F(x_0) - F^\star\right) + \delta F^\star$$

where $\rho = \prod_{t=1}^T \frac{1}{1-\varepsilon_t}\frac{\lambda_t}{\mu+\lambda_t}$ and $\delta = \sum_{t=1}^T \left(\frac{\varepsilon_t}{1-\varepsilon_t}\prod_{j=t+1}^T \frac{1}{1-\varepsilon_t}\frac{\lambda_j}{\mu+\lambda_j}\right)$.

Theorem G.4.1.2. *For a smooth convex function F, let $\varepsilon_1, \ldots, \varepsilon_T = \varepsilon$ where $\varepsilon \in (0, 1/2)$ and $x_0, \ldots, x_T \in \mathbb{R}^d$ be as in Theorem G.4.1.1. Then, we have*

$$F(x_T) - F^\star \leq \frac{2}{(1-\varepsilon)}\frac{\|x^\star - x_0\|_2^2}{\sum_{t=1}^T \frac{2}{\lambda_t}} + \frac{3\varepsilon}{1-\varepsilon}F^\star.$$

G.4.2 Local Sensitivity Sampling

We now discuss how to choose the parameters for Algorithm 19 when using local sensitivity sampling to implement the ε-oracle in each step. From Lemmas G.3.1.1 and G.3.1.2 it is clear that if λ_t goes down, the corresponding radius r_t goes up. However, in Theorem G.2.2.1, we bound the true local sensitivity at iteration t by a quantity depending on $\frac{r_t}{\lambda_t}$, which comes from the error in the quadratic approximation. Thus, if we choose λ_t very small, the term $\frac{r_t}{\lambda_t}$ will dominate in the local sensitivity approximation, and we won't see any advantage from local sensitivity sampling over, e.g., uniform sampling. Making λ_t large will improve the local sensitivity approximation but slow down convergence.

To balance these factors, we will choose λ_t of the order of r_t. In particular, considering Lemma G.3.1.2, we choose $\lambda_t = \sqrt{\|\nabla F(x_{t-1})\|_2}$. The lemma then gives that $r_t \leq \frac{\|\nabla F(x_{t-1})\|}{\sqrt{\|\nabla F(x_{t-1})\|}+\mu} \leq \sqrt{\|\nabla F(x_{t-1})\|_2}$. We here now provide an end to end algorithm which utilizes local sensitivity sampling in the approximate proximal point method framework presented in Algorithm 19. The pseudo-code and details of the algorithm are given in Algorithm 20 where we denote $F^s_{\lambda_t, x_{t-1}}(x)$ as the importance sampled subset of $F_{\lambda_t, x_{t-1}}(x)$ which has been obtained via local sensitivity sampling. Line 9 of Algorithm 20 can be considered as a black-optimization problem which is apparently a strongly-convex optimization problem and can be optimized exponentially fast.

On Convergence: With this choice of λ_t, the convergence rate of APPM under our strong convexity assumption will be $\mathcal{O}\left(\frac{\|\sqrt{\tilde{\nabla} F(x)}\|_2}{\mu}\log(1/\varepsilon)\right)$ where $\sqrt{\|\tilde{\nabla} F(x)\|_2}$ represents $\frac{1}{T}\sum_{i=0}^{T-1}\sqrt{\|\nabla F(x_i)\|_2}$. If F is smooth with smoothness parameter L, we have: $\|\nabla F(x)\|_2 \leq L\|x - x^\star\|_2$. For the smooth but non-strongly convex problem, if we assume $\lambda_t \leq \varepsilon$ for some ε for all t then, $\|\nabla F(x_t)\|_2^2 \in \mathcal{O}(1/T)$ in the worst case. Hence, the rate of for non-strongly convex smooth function will behave like $\mathcal{O}(1/T^{5/4})$.

G.5 An Adaptive Stochastic Trust Region Method

Related to the proximal point approach, sensitivity sampling can be used to obtain an adaptive stochastic trust region. In each iteration t, we approximately minimize a quadratic approximation to F over a ball, using local sensitivity sampling and directly applying the sensitivity score bound of Theorem G.2.0.1. At iteration t the center of the ball is at x_{t-1} and the radius is set to $r_t = \frac{\|\nabla F(x_{t-1})\|_2}{\lambda_t+\mu}$. We provide pseudocode in Algorithm 22 and a proof of a convergence bound in Appendix G.12. Here we just state the main result.

Theorem G.5.0.1. *For a given set of constants C_k, $\delta_k \in (0,1)$, and $\tilde{\varepsilon}_k = \delta_k \frac{\mu}{\lambda_k+\mu}$ which is an error tolerance for the quadratic approximation of the function $F_{\lambda_k, x_{k-1}}(x)$ for all k, if*

Algorithm 20 APPM with Local Sensitivity Sampling

input $x_0 \in \mathbb{R}^d$, ε_t, and μ.
Compute $\|\nabla F(x_0)\|_2$, $F(x_0)$, and C_0
for $t = 1 \dots$ T **do**
  Compute regularizer $\lambda_t \leftarrow \sqrt{\|\nabla f(x_{t-1})\|_2}$.
  Compute radius $r_t \leftarrow \frac{\|\nabla f(x_{t-1})\|_2}{\sqrt{\|\nabla f(x_{t-1})\|_2}+\mu}$.
  Get $\tilde{F}_{\lambda_t, x_{t-1}}$ via Taylor Expansion.
  Compute the local sensitivity for $F_{\lambda_t, x_{t-1}}$ using Theorem G.2.2.1.
  Local sensitivity based sampling of $F^s_{\lambda_t, x_{t-1}}(x)$ from $F_{\lambda_t, x_{t-1}}(x)$.
  $x_t \leftarrow \arg\min_{x \in B(r_t, x_{t-1})} F^s_{\lambda_t, x_{t-1}}(x)$.
  Compute $\|\nabla F(x_t)\|_2$.
end for
output x_T

λ_{k+1} is chosen of $\mathcal{O}(\sqrt{\|\nabla F(x_k)\|_2})$ then at iteration $k+1$ Algorithm 22 satisfies:

$$F(x_{k+1}) - F^\star \leq (1+2\varepsilon_{k+1}) \frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu} (F(x_k) - F^\star) + 2\varepsilon_{k+1} F^\star, \tag{G.9}$$

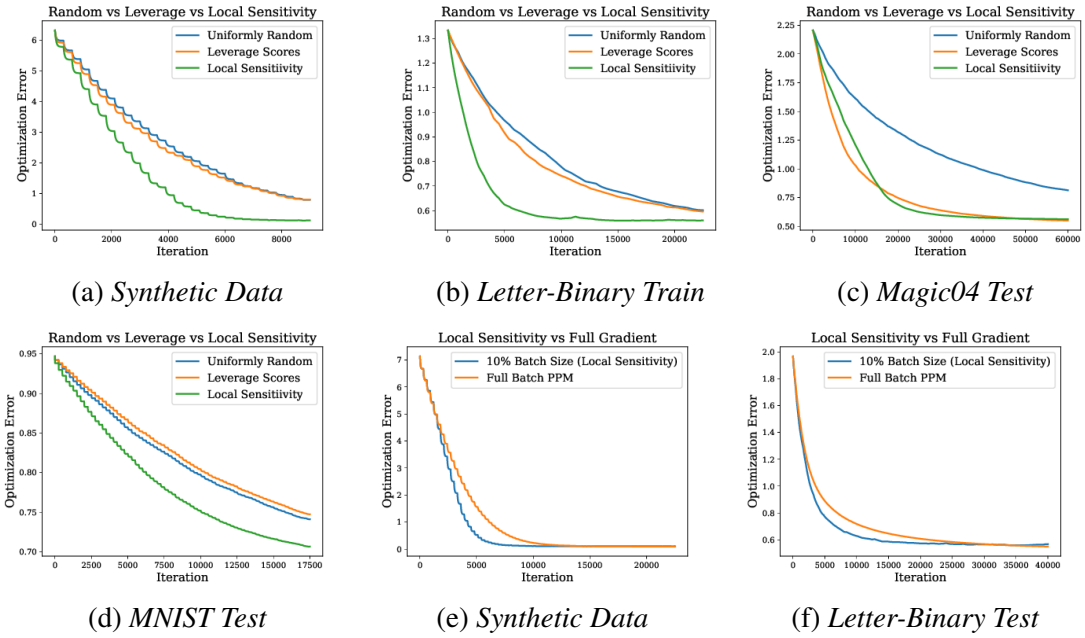
where $\varepsilon_{k+1} = 2\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)$, m and c are positive constants.

Comparing equation (G.9) in Theorem G.5.0.1 with the bound in Theorem G.4.1.1, we can see that we have obtained a similar recursive relation in both equations, and hence the trust region method will have a similar convergence rate to APPM in the presence of an *ε-multiplicative* oracle.

G.6 Experiments

We conclude by giving some initial experimental evidence to justify the performance of our proposed algorithm in practice. We provide the experiments for *Approximate Proximal Point Method with Local Sensitivity Sampling* (Algorithm 20). We run our algorithm on the following four datatsets[1] : (a) *Synthetic Data* (b) *Letter Binary* [70] (c) *Magic04* [28] and (d) *MNIST Binary* [128]. Prefix 'Train' or 'Test' denotes if the train or test split was used for the experiment. The *Synthetic Data* was generated by first generating a matrix A of size 3000×300 drawn from a 300 dimensional standard normal random variable. Then another vector x_0 of size 300 was fixed which is also drawn from a normal random variable to obtain $\hat{y} = Ax_0 + \eta$ where $\eta \sim 0.1 * \mathcal{N}(0,1)$. Finally, the classification label vector y was chosen as $\text{sign}(\hat{y})$. We perform all our experiments for

[1] Datasets can be downloaded from: manikvarma.org/code/LDKL/download.html.



(a) *Synthetic Data* (b) *Letter-Binary Train* (c) *Magic04 Test*

(d) *MNIST Test* (e) *Synthetic Data* (f) *Letter-Binary Test*

Figure G.2: (a-d) Local sensitivity sampling vs. uniform random sampling and leverage score sampling on four datasets: (a) *Synthetic Data* (3000 points), (b) *Letter Binary Train* (12000 points), (c) *Magic04 Test* (4795 points), and (d) *MNIST Test* (10000 points). (e-f) Local Sampling Method is compared with Full Batch Gradient for (e) *Synthetic* and (f) *Letter Binary Test*.

logistic regression with an ℓ_2^2 regularization parameter of 0.001. For the experiments plotted in the Figure G.2, we have considered a fixed sample size of 100 data points for every iteration of the proximal algorithm. In the first four subfigures of Figure G.2, we compare compare local sensitivity sampling with two base lines: uniform random sampling and sampling using the leverage scores of the data matrix A. On the horizontal axis, we report the total number of iterations which is the number of times the sampling oracle is called (outer loop in Algorithm 20) multiplied by number of times the gradient call to solve the optimization problem given in Line 9 in Algorithm 20. We report the optimization error on vertical axis.

From the plots in Figures G.2a, G.2b, G.2c and G.2d, it is evident that our method outperforms uniform random sampling with a large margin on the synthetic and real datasets. It also often performs much better than leverage score sampling. Since the local sensitively approximations of Theorems G.2.0.1 and G.2.2.1 are the leverage scores of a matrix with essentially the same dimensions as A, these methods have the same order of computational cost.

We perform a second set of experiments to compare our sampling technique with full batch gradient iteration for each proximal point iteration on *Synthetic* and *Letter Binary Test* which we plot in Figures G.2e and G.2f. We can see in Figures G.2e and G.2f that our

sampling method outperforms the full gradient just with 10% of total points. In both plots, the sampling method needs just half of the number of iterations taken by full gradient to saturate to similar value.

In both of the experiments, we set the number of inner loop iteration (number of calls to the gradient oracle for solving Line 9 in Algorithm 20) in advance to let the optimization error saturate for that particular outer loop; however the plots demonstrate that it can be set to a much smaller number or can be set adaptively to achieve gains of multiple folds.

G.7 Conclusion

In this work, we study how the elegant approach of function approximation via sensitivity sampling can be made practical. We overcome two barriers: (1) the difficulty of approximating the sensitivity scores and (2) the high sample complexities required by theoretical bounds. We handle both by considering a *local* notion of sensitivity, which we can efficiently approximate and bound. We demonstrate that this notion can be combined with methods that globally optimize a function via iterative local optimizations, including proximal point and trust region methods.

Our work leaves open a number of questions. Most importantly, since local sensitivity approximation incurs some computational overhead (a leverage score computation along with some derivative computations), we believe it will be especially useful for functions that are difficult to optimize, e.g., non-strongly-convex functions. Understanding how our theory extends and how our method performs in practice on such functions would be very interesting. It would be especially interesting to compare performance to related approaches, such as quasi-Newton and other trust region approaches.

Proofs for Main Results

G.8 Leverage Scores as Sensitivities of Quadratic Functions

We here start by stating Lemma G.8.0.1 and giving its proof. This lemma is helpful in proving Theorem G.2.0.1. Lemma G.8.0.1 is a relatively well known characterization of the leverage scores of a matrix, see e.g, [17]; however for completeness we give a proof here.

Lemma G.8.0.1 (Leverage Scores as Sensitivities)**.** *For any $C \in \mathbb{R}^{n\times p}$ with i^{th} row c_i,*

$$\ell_i^{\lambda}(C) = \max_{\{z\in\mathbb{R}^p:\|z\|>0\}} \frac{[Cz]_i^2}{\|Cz\|_2^2+\lambda\|z\|_2^2} = c_i^T(C^TC+\lambda I)^{-1}c_i.$$

Proof. Write $\sigma(z) = \frac{[Cz]_i^2}{\|Cz\|_2^2+\lambda\|z\|_2^2}$, $f(z) = [Cz]_i^2 = (c_i^Tz)^2$, $g(z) = \|Cz\|_2^2+\lambda\|z\|_2^2 = z^T(C^TC+\lambda I)z$. We can compute the gradient of $\sigma(z)$ as:

$$\nabla_j\sigma(z) = \frac{\nabla_j f(z)\cdot g(z) - \nabla_j g(z)\cdot f(z)}{g(z)^2}.$$

At the minimium this must equal 0 and so since $g(z) > 0$ for z with $\|z\|_2 > 0$, we must have $\nabla f(z)\cdot g(z) - \nabla g(z)\cdot f(z) = 0$. We have $\nabla f(z) = 2c_i^Tz\cdot c_i$ and $\nabla g(z) = 2(C^TC+\lambda I)z$. We thus have at optimum:

$$c_i\cdot\left(2c_i^Tz\cdot z^T(C^TC+\lambda I)z\right) - 2(C^TC+\lambda I)z\cdot(c_i^Tz)^2 = 0.$$

Dividing by $2(c_i^Tz)^2$ we must have:

$$-c_i\cdot\frac{z^T(C^TC+\lambda I)z}{c_i^Tz} = (C^TC+\lambda I)z.$$

For this to hold we must have $(C^TC+\lambda I)z$ equal to a multiple of c_i and so $z = \alpha\cdot(C^TC+\lambda I)^{-1}c_i$ for some α. Note that the value of α does not change the value of $\sigma(z)$ since it simply scales the numerator and denominator in the same way. So we have that

$$z^\star = \operatorname*{arg\,max}_{\{z\in\mathbb{R}^d:\|z\|_2>0\}} \frac{[Cz]_i^2}{\|Cz\|_2^2+\lambda\|z\|_2^2} = (C^TC+\lambda I)^{-1}z_i.$$

Plugging in we have:

$$\max_{\{z\in\mathbb{R}^d:\|z\|_2>0\}} \frac{[Cz]_i^2}{\|Cz\|_2^2+\lambda\|z\|_2^2} = \frac{\left(c_i^T(C^TC+\lambda I)^{-1}c_i\right)^2}{c_i^T(C^TC+\lambda I)^{-1}(C^TC+\lambda I)(C^TC+\lambda I)^{-1}c_i}$$
$$= c_i^T(C^TC+\lambda I)^{-1}c_i,$$

which completes the proof. □

Proof of Theorem G.2.0.1. Letting $z = x-y$ and $\eta = \min_{x\in B(r,y)} \tilde{F}_{\lambda,y}(x)$ we can write:

$$\tilde{F}_{\lambda,y}(x) = \underbrace{\left[\frac{1}{2}\|H_y^{1/2}Az + H_y^{-1/2}\alpha_y\|^2 + \lambda\|z\|^2 + \gamma(z+y)\right]}_{:G(z)=\sum_{i=1}^n g_i(z)+\lambda\|z\|^2+\gamma(z+y)} + \underbrace{\left[F(y) - \frac{1}{4}\|H^{-1/2}\alpha_y\|^2\right]}_{\Delta=\sum_{i=1}^n \Delta_i}. \tag{G.10}$$

where $g_i(z) = \frac{1}{2}(H_y^{1/2}Az + H_y^{-1/2}\alpha_y)_i^2$ and $\Delta_i = f_i(a_i^Ty) - \frac{1}{4}\left(H_y^{-1/2}\alpha_y\right)_i^2$. Noting that $G(z)$ is nonnegative, we can write the sensitivity as:

$$\sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) = \max_{\{z:\|z\|\le r\}} \frac{g_i(z)+\Delta_i}{G(z)+\Delta} = \max_{\{z:\|z\|<r\}} \left[\frac{g_i(z)}{G(z)}\cdot\frac{G(z)}{G(z)+\Delta} + \frac{\Delta_i}{G(z)+\Delta}\right]$$
$$\le \max_{z\in\mathbb{R}^d}\left[\frac{g_i(z)}{G(z)}\cdot\frac{G(z)}{G(z)+\Delta}\right] + \frac{f_i(a_i^Ty)}{\eta} \tag{G.11}$$

since $G(z)+\Delta = \tilde{F}_{\lambda,y}(y+z) \ge \eta$ for $\eta = \min_{x\in B(r,y)}\tilde{F}_{\lambda,y}(x)$ and since $f_i(a_i^Ty)\ge\Delta_i$. When $\Delta\ge 0$, $\frac{G(z)}{G(z)+\Delta}\le 1$. When $\Delta<0$:

$$\frac{G(z)}{G(z)+\Delta} = 1 - \frac{\Delta}{G(z)+\Delta} = 1 - \frac{\Delta}{\tilde{F}_{\lambda,y}(x)} \le 1 - \frac{\Delta}{\eta}.$$

Overall we have:

$$\sigma_{\tilde{F}_{\lambda,y},\mathcal{W}_\eta}(a_i) \le \max\left(1, 1-\frac{\Delta}{\eta}\right)\cdot\max_{\{z:z+y\in\mathcal{W}_\eta\}}\left[\frac{g_i(z)}{G(z)}\right] + \frac{f_i(a_i^Ty)}{\eta}. \tag{G.12}$$

Letting $\delta = \min_{x\in B(r,y)}\gamma(x) = \min_{z:\|z\|\le r}\gamma(z+y)$, $C\in\mathbb{R}^{n\times d+1}$ be the matrix $[H_y^{1/2}A, \frac{1}{\delta}H_y^{-1/2}\alpha_y]$ and $\bar{z} = [z,-\delta]$ we have:

$$\frac{g_i(z)}{G(z)} = \frac{(C\bar{z})_i^2}{\|C\bar{z}\|^2+\lambda\|z\|^2+\gamma(z+y)} = \frac{(C\bar{z})_i^2}{\|C\bar{z}\|^2+\lambda\|\bar{z}\|^2-\delta+\gamma(z+y)} \tag{G.13}$$

We can bound this ratio using Lemma G.8.0.1. Specifically, since $\gamma(z+y)-\delta\ge 0$ the

ratio by $\ell_i^{\lambda}(C)$. Plugging back into (G.12) we have:

$$\sigma_{\tilde{F}_{\lambda,y},\mathcal{W}_\eta}(a_i) \leq \max\left(1, 1-\frac{\Delta}{\eta}\right)\cdot \ell_i^{\lambda}(C) + \frac{f_i(a_i^T y)}{\eta},$$

which completes the proof. □

G.9 Local Sensitivity Bound via Quadratic Approximation

We next prove Theorem G.2.2.1, which bounds the local sensitivities of a function in terms of the sensitivities of a quadratic approximation to that function, which can in term be bounded using the leverage scores of an appropriate matrix (Theorem G.2.0.1).

Theorem' G.2.2.1. *Consider $F_{\lambda,y}$ as in Defs. G.1.0.1 and G.2.0.2, $y \in \mathcal{X}$, radius r, and $\alpha = \min_{x\in B(r,y)} F_{\lambda,y}(x)$. We have:*

$$\sigma_{F_{\lambda,y},B(r,y)}(a_i) \leq \sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) + \min\left(\frac{C_i r}{6n\lambda}, \frac{C_i r^3}{6n\alpha}\right),\ \forall\, i \in [n].$$

Proof. From the local quadratic approximation, we have :

$$F_{\lambda,y}(x) = \tilde{F}_{\lambda,y}(x) + B_y(x)\|x-y\|^3, \text{ where } \tilde{F}_{\lambda,y}(x) = \frac{1}{n}\sum_{i=1}^{n} \tilde{f}_i(a_i^\top x) + \gamma(x) + \lambda\|x-y\|^2.$$

From the previous Theorem G.2.0.1, we have a bound on the sensitivity for quadratic approximation,

$$\sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i) = \sup_{x\in B(r,y)} \frac{\frac{1}{n}\tilde{f}_i(a_i^\top x)}{\tilde{F}_{\lambda,y}(x)}$$

We can bound the local sensitivity of the true function $F_{\lambda,y}$ by:

$$\sigma_{F_{\lambda,y},B(r,y)}(a_i) = \sup_{x\in B(r,y)} \frac{\frac{1}{n} f_i(a_i^\top x)}{F_{\lambda,y}(x)} = \sup_{x\in B(r,y)} \frac{\frac{1}{n}\left[\tilde{f}_i(a_i^\top x) + B_y^{(i)}(x)\|x-y\|^3\right]}{F_{\lambda,y}(x)}$$

We have assumed that $B_y(x) = \frac{1}{n}\sum_{i=1}^n B_y^{(i)}(x)$ is positive for $x \in B(r,y)$ and that $B_y^{(i)}(x) \leq$

$\frac{1}{6}C_i$ for all i. This gives:

$$\begin{aligned}\sigma_{F_{\lambda,y},B(r,y)}(a_i) &= \sup_{x\in B(r,y)} \frac{\frac{1}{n}\left[\tilde{f}_i(a_i^\top x) + B_y^{(i)}(x)\|x-y\|^3\right]}{F_{\lambda,y}(x)} \\ &\leq \underbrace{\sup_{x\in B(r,y)} \frac{\frac{1}{n}\left[\tilde{f}_i(a_i^\top x)\right]}{F_{\lambda,y}(x)}}_{:=\text{ term 1}} + \underbrace{\sup_{x\in B(r,y)} \frac{C_i}{6n}\frac{\|x-y\|^3}{F_{\lambda,y}(x)}}_{:=\text{term 2}}.\end{aligned}$$

For term 1 we have:

$$\sup_{x\in B(r,y)} \frac{\frac{1}{n}\left[\tilde{f}_i(a_i^\top x)\right]}{F_{\lambda,y}(x)} = \sup_{x\in B(r,y)} \frac{\frac{1}{n}\left[\tilde{f}_i(a_i^\top x)\right]}{\tilde{F}_{\lambda,y}(x) + B_y(x)\|x-y\|^3} \leq \sigma_{\tilde{F}_{\lambda,y},B(r,y)}(a_i),$$

where the inequality comes from assumption that $B_y(x) > 0$ for $x \in B(r,y)$. For term 2 we simply bound $F_{\lambda,y}(x) \geq \alpha := \min_{x\in B(r,y)} F_{\lambda,y}(x)$ or alternatively, $F_{\lambda,y}(x) \geq \lambda\|x-y\|^2$ giving:

$$\frac{C_i}{6n}\frac{\|x-y\|^3}{F_{\lambda,y}(x)} \leq \min\left(\frac{C_i r}{6n\lambda}, \frac{C_i r^3}{6n\alpha}\right),$$

which completes the theorem. □

G.10 Constrained Penalized Connection

Proof of Lemma G.3.1.1. Given, $x^\star = \arg\min_{x\in\mathbb{R}^d} F(x)$. We assume that F is a convex function. From KKT conditions, if $x^\star$ does not lie inside the ball than the optimal solution will exist on the boundary of the ball. Hence, the inequality in the equation can be replaced with the equality given that $x^\star$ doesn't lie inside the ball represented by the equations $\|x-y\|^2 = r^2$. The optimization problem then becomes:

$$x^\star_{r,y} = \arg\min_{x\in r^d} F(x) \text{ such that } \|x-y\|^2 = r^2 \tag{G.14}$$

The Lagrangian of equation (G.14) is: $L(x,\nu) = F(x) + \frac{\nu}{2}\left(\|x-y\|^2 - r^2\right)$. First order optimality condition for the above equation implies $\nabla F(x^\star_{r,y}) + \nu^\star(x^\star_{r,y} - y) = 0 \Rightarrow\ x^\star_{r,y} - y = \frac{-1}{\nu^\star}\nabla F(x^\star_{r,y})$. Now from the constrained we have, $\|\frac{-1}{\nu^\star}\nabla F(x^\star_{r,y})\| = r\ \Rightarrow\ \nu^\star = \frac{\|\nabla F(x^\star_{r,y})\|}{r}$. Hence, it is clear from the above arguement that $x^\star_{r,y}$ also optimize the following optimiza-

tion problem:

$$x^{\star}_{r,y} = \arg\min_{x \in \mathbb{R}^d} \left[F(x) + \frac{\|\nabla F(x^{\star}_{\hat{R},y})\|}{2r} \|x-y\|^2 \right] \tag{G.15}$$

□

Proof of Lemma G.3.1.2. As we have:

$$F_{\lambda,y}(x) = F(x) + \lambda \|x-y\|^2$$

From the property of strongly convex function:

$$\|\nabla F_{\lambda,y}(y)\| = \|\nabla F(y)\| \geq (\mu + 2\lambda)\|y - x^{\star}_{\lambda,y}\| \tag{G.16}$$

Now from the first order optimality of $F_{\lambda,y}$, we have:

$$\nabla F_{\lambda,y}(x^{\star}_{\lambda,y}) = \nabla F(x^{\star}_{\lambda,y}) + 2\lambda(x^{\star}_{\lambda,y} - y) = 0$$

Hence,

$$\|\nabla F(x^{\star}_{\lambda,y})\| = 2\lambda \|x^{\star}_{\lambda,y} - y\| \tag{G.17}$$

From the equations (G.16) and (G.17), we have:

$$\|\nabla F(x^{\star}_{\lambda,y})\| \leq \frac{2\lambda}{\mu + 2\lambda} \|\nabla F(y)\|$$

From the equation (G.15), we know that

$$R = \frac{\|\nabla F(x^{\star}_{\lambda,y})\|}{2\lambda} \leq \frac{\|\nabla F(y)\|}{2\lambda + \mu}$$

If the optimal point $x^{\star}$ of the function F lie in the ball then the radius will be further less. □

Corollary G.10.0.1. *After running one step of line 4 of the Algorithm 19 for the parameters x_{t-1}, λ_t, ε_t and μ, we have the following bound:*

$$\|x_t - x_{t-1}\| \leq \sqrt{\frac{2\varepsilon_t}{2\lambda_t + \mu} F(x_{t-1})} + \frac{\|\nabla F(x_{t-1})\|}{2\lambda_t + \mu}$$

$$\|x_t - x_{t-1}\| \geq r^{\star}_t - \sqrt{\frac{2\varepsilon_t}{2\lambda_t + \mu} F(x_{t-1})}$$

where $r_t^\star = \|x_{t-1} - x_{\lambda_t}^\star\|$.

Proof. As from Lemma G.3.1.2, we have

$$\|x_{2\lambda_t, x_{t-1}}^\star - x_{t-1}\| \leq \frac{\|\nabla F(x_{t-1})\|}{2\lambda_t + \mu}.$$

Let us denote $\|x_{\lambda_t, x_{t-1}}^\star - x_{t-1}\|$ as r_t. Now, let us try to bound $\|x_t - x_{\lambda_t, x_{t-1}}^\star\|$. From the strong convexity and aproximation argument:

$$\|x_t - x_{\lambda_t, x_{t-1}}^\star\|^2 \leq \frac{2}{2\lambda_t + \mu}\left(F_{\lambda_t, x_{t-1}}(x_t) - f_{\lambda_t, x_{t-1}}^\star\right) \text{Ł} \leq \frac{2\varepsilon_t}{2\lambda_t + \mu} f_{\lambda_t, x_{t-1}}^\star$$

Now we can apply strong convexity argument one more time.

$$f_{\lambda_t, x_{t-1}}^\star \leq F(x_{t-1}) - \frac{2\lambda_t + \mu}{2} r_t^2$$

Hence finally we have:

$$\|x_t - x_{\lambda_t, x_{t-1}}^\star\|^2 \leq \frac{2\varepsilon_t}{2\lambda_t + \mu} F(x_{t-1}) - \varepsilon_t r_t^2 \tag{G.18}$$

Hence finally:

$$\begin{aligned} r_t - \sqrt{\frac{2\varepsilon_t}{2\lambda_t + \mu} F(x_{t-1})} \leq \|x_t - x_{t-1}\| &\leq \sqrt{\frac{2\varepsilon_t}{2\lambda_t + \mu} F(x_{t-1})} + r_t \\ &\leq \sqrt{\frac{2\varepsilon_t}{2\lambda_t + \mu} F(x_{t-1})} + \frac{\|\nabla F(x_{t-1})\|}{2\lambda_t + \mu} \end{aligned}$$

□

G.11 Approximate Proximal Point Method

The following Lemma from G.11.0.1 is useful in proving the Theorem G.4.1.1.

Lemma G.11.0.1 (Lemma 2.7 [73]). *For all* $y \in \mathbb{R}^d$ *and* $\lambda \geq 0$*:*

$$F(x_{\lambda, y}^\star) - F^\star \leq F_{\lambda, y}^\star - F^\star \leq \frac{2\lambda}{\mu + 2\lambda}\left(F(y) - F^\star\right).$$

Proof of Theorem G.4.1.1. Let us assume that $x_{\lambda, x}^\star = \arg\min_{y \in \mathbb{R}^d} F_{\lambda, x}(y)$, then from the

Lemma G.11.0.1,

$$\begin{aligned} F^\star_{\lambda,x} - F^\star &\leq \frac{2\lambda}{\mu+2\lambda}\left(F(x)-F^\star\right) \\ \Rightarrow\ F(x^\star_{\lambda,x}) - F^\star &\leq \frac{2\lambda}{\mu+2\lambda}\left(F(x)-F^\star\right) \end{aligned} \tag{G.19}$$

Last equation comes from the fact that $F^\star_{\lambda,x} = F(x^\star_{\lambda,x}) + \lambda\|x^\star - x\|^2$.

We know that

$$F_{\lambda,y}(x^\star_{\lambda,y}) \leq f\left(\mathcal{P}_{F_{\lambda,y}}(x)\right) \leq (1+\varepsilon)F_{\lambda,y}(x^\star_{\lambda,y})\ \forall\ y \in \mathbb{R}^d.$$

We can get the upper bound on the true minimizer using this black-box oracle in terms of the approximate solution. We have:

$$F^\star_{\lambda_T,x_{t-1}} \leq F_{\lambda_T,x_{t-1}}(x_t) \tag{G.20}$$

From Lemma G.11.0.1 and black-box oracle , for any $t \in [T]$ we have

$$\begin{aligned} F_{\lambda_t,x_{t-1}}(x_t) - F^\star &= F_{\lambda_t,x_{t-1}}(x_t) - F^\star_{\lambda_t,x_{t-1}} + F^\star_{\lambda_t,x_{t-1}} - F^\star \\ &\leq \varepsilon_t\ F^\star_{\lambda,x_{t-1}} + \frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) \\ &\leq \varepsilon_t\ F_{\lambda_t,x_{t-1}}(x_t) + \frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) \end{aligned} \tag{G.21}$$

which leads us to

$$\begin{aligned} &(1-\varepsilon_t)F_{\lambda_t,x_{t-1}}(x_t) - F^\star \leq \frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) \\ \Rightarrow\ &(1-\varepsilon_t)F_{\lambda_t,x_{t-1}}(x_t) - (1-\varepsilon_t)F^\star \leq \frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) + \varepsilon_t F^\star \\ \Rightarrow\ &F_{\lambda_t,x_{t-1}}(x_t) - F^\star \leq \frac{1}{1-\varepsilon_t}\frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) + \frac{\varepsilon_t}{1-\varepsilon_t}F^\star \end{aligned} \tag{G.22}$$

Now since,

$$F_{\lambda_t,x_{t-1}}(x_t) = F(x_t) + \lambda_t\|x_t - x_{t-1}\|^2 \geq F(x_t)$$

Hence, finally we have:

$$\begin{aligned} F(x_t) - F^\star &\leq \frac{1}{1-\varepsilon_t}\frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) + \frac{\varepsilon_t}{1-\varepsilon_t}F^\star \\ &\leq (1+2\varepsilon_t)\frac{2\lambda_t}{\mu+2\lambda_t}\left(F(x_{t-1})-F^\star\right) + 2\varepsilon_t F^\star \end{aligned} \tag{G.23}$$

whenever $\varepsilon_t \leq 1/2$. Now we can do recursion on the equation (G.23):

$$F(x_T) - F^\star \leq \underbrace{\left[\prod_{t=1}^{T}(1+2\varepsilon_t)\frac{2\lambda_t}{\mu+2\lambda_t}\right]}_{\text{:linear rate}}(F(x_0)-F^\star) + F^\star \underbrace{\left[\sum_{t=1}^{T}2\varepsilon_t \prod_{j=t+1}^{T}(1+2\varepsilon_j)\frac{2\lambda_j}{\mu+2\lambda_j}\right]}_{:=\delta} \tag{G.24}$$

□

Algorithm 21 Proximal-Point Method

1: **input** $x_0 \in \mathbb{R}^d$, $\lambda_t > 0 \; \forall \, t \in [T]$.
2: **for** $t = 1 \ldots$ T **do**
3: $\quad x^\star_{\lambda_t, x_{t-1}} \leftarrow \arg\min F(x) + \lambda_t \|x - x_{t-1}\|^2$
4: $\quad x_t \leftarrow x^\star_{\lambda_t, x_{t-1}}$
5: **end for**
6: **output** x_T

Lemma G.11.0.2 (Proposition 3.1.6 [243]). *Let F be lower semi-continuous convex function then for any x in the domain and for any $t \geq 1$ following relation holds for iterates in Algorithm 21:*

$$\frac{1}{\lambda_t}\left(F(x) - F(x^\star_{\lambda_t, x_{t-1}})\right) \geq \|x_{t-1} - x^\star_{\lambda_t, x_{t-1}}\|^2 + \|x - x^\star_{\lambda_t, x_{t-1}}\|^2 - \|x - x_{t-1}\|^2.$$

In the next lemma, we characterize the result provided in lemma G.11.0.2 for the ε-approximate oracle.

Lemma G.11.0.3. *Let F be lower semi-continuous convex function then for $x^\star$, the minimizer of F and for any $t \geq 1$ and $\varepsilon \leq 1/2$, following relation holds for iterates in Algorithm 19:*

$$\begin{aligned}\frac{1}{\lambda_t}\left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star\right) &\leq \frac{2}{\lambda_t}\left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star\right) + \frac{\varepsilon_t}{(1-\varepsilon_t)\lambda_t}F^\star \\ &\leq 2\|x^\star - x_{t-1}\|^2 - 2\|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 + \frac{\varepsilon_t}{(1-\varepsilon_t)\lambda_t}F^\star\end{aligned}$$

Proof. We have $x_t = P_{f_{\lambda_t, x_{t-1}}}(x)$ as defined in line 3 of Algorithm 19 where $\mathcal{P}_f$ is *multiplicative* ε_t-oracle.
From the oracle we know that $F_{\lambda, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) \leq F_{\lambda, x_{t-1}}(x_t) \leq (1+\varepsilon_t)F_{\lambda, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}})$.
Next we use the result from Lemma G.11.0.2 where we use $x = x^\star = \arg\min_x F(x)$. We

denote $F^\star$ with $F(x^\star)$.

$$\begin{aligned}\frac{1}{\lambda_t}\left(F^\star - F(x^\star_{\lambda_t,x_{t-1}})\right) &\geq \|x_{t-1} - x^\star_{\lambda_t,x_{t-1}}\|^2 + \|x^\star - x^\star_{\lambda_t,x_{t-1}}\|^2 - \|x^\star - x_{t-1}\|^2 \\ &\geq \|x_{t-1} - x^\star_{\lambda_t,x_{t-1}}\|^2 + \|x^\star - x_t + x_t - x^\star_{\lambda_t,x_{t-1}}\|^2 - \|x^\star - x_{t-1}\|^2\end{aligned} \tag{G.25}$$

The last equation essentially tells us the following:

$$\frac{1}{\lambda_t}\left(F^\star - F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}})\right) \geq \|x^\star - x^\star_{\lambda_t,x_{t-1}}\|^2 - \|x^\star - x_{t-1}\|^2 \tag{G.26}$$

From the ε_t-oracle we do have:

$$\left(F_{\lambda_t,x_{t-1}}(x_t) - F_{\lambda,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}})\right) \leq \varepsilon_t F_{\lambda,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}})$$

Hence

$$\begin{aligned}\frac{1}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x_t) - F^\star\right) &= \frac{1}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x_t) - F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) + F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right) \\ &= \frac{1}{\lambda_t}\left[\left(F_{\lambda_t,x_{t-1}}(x_t) - F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}})\right) + \left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right)\right] \\ &\leq \frac{1}{\lambda_t}\left[\varepsilon_t F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) + \left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right)\right] \\ &\leq \frac{1}{\lambda_t}\left[\varepsilon_t F_{\lambda_t,x_{t-1}}(x_t) + \left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right)\right]\end{aligned} \tag{G.27}$$

From equations (G.26) and (G.27), we have:

$$\begin{aligned}&\frac{1}{\lambda_t}\left((1-\varepsilon_t)F_{\lambda_t,x_{t-1}}(x_t) - F^\star\right) \leq \frac{1}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right) \\ \Rightarrow\ &\frac{1(1-\varepsilon_t)}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x_t) - F^\star\right) \leq \frac{1}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right) + \frac{\varepsilon_t}{\lambda_t}F^\star \\ \Rightarrow\ &\frac{1}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x_t) - F^\star\right) \leq \frac{1}{(1-\varepsilon_t)\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right) + \frac{\varepsilon_t}{(1-\varepsilon_t)\lambda_t}F^\star\end{aligned}$$

If $\varepsilon_t \leq 1/2$, then from equations (G.26) and (G.27), we have:

$$\frac{1}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x_t) - F^\star\right) \leq \frac{2}{\lambda_t}\left(F_{\lambda_t,x_{t-1}}(x^\star_{\lambda_t,x_{t-1}}) - F^\star\right) + \frac{\varepsilon_t}{(1-\varepsilon_t)\lambda_t}F^\star$$

$$\leq 2\|x^\star - x_{t-1}\|^2 - 2\|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 + \frac{\varepsilon_t}{(1-\varepsilon_t)\lambda_t} F^\star$$

□

Lemma G.11.0.4. *For a lower semi-continuous convex function F at any and for any $t \geq 1$ and $\varepsilon \leq 1/2$, following relation holds for iterates after T iterations in Algorithm 19:*

$$\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star \right) \leq \frac{2}{(1-\varepsilon)} \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \frac{3\varepsilon}{((1-\varepsilon))\lambda_t} F^\star$$

Proof. We know that:

$$\frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) \leq \|x^\star - x_{t-1}\|^2 - \|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 \tag{G.28}$$

We can however sum the equation (G.28) for $t = 1$ till T and we get:

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) &\leq \sum_{t=1}^{T} \left[\|x^\star - x_{t-1}\|^2 - \|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 \right] \\
&= \|x^\star - x_0\|^2 + \sum_{t=1}^{T-1} \left[\|x^\star - x_t\|^2 - \|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 \right] \\
&\qquad - \|x^\star - x^\star_{\lambda_t, x_{T-1}}\|^2 \\
&\leq \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \left[\|x^\star - x_t\|^2 - \|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 \right]
\end{aligned} \tag{G.29}$$

In equation (G.29), we can use Corollary G.10.0.1,

$$\|x^\star - x_t\|^2 - \|x^\star - x^\star_{\lambda_t, x_{t-1}}\|^2 \leq \frac{\varepsilon_t}{\lambda_t} F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}).$$

Hence,

$$\begin{aligned}
&\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) \leq \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \frac{\varepsilon_t}{\lambda_t} F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) \\
\Rightarrow &\sum_{t=1}^{T} \frac{1}{\lambda_t} \left((1-\varepsilon_t) F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) \leq \|x^\star - x_0\|^2 \\
\Rightarrow &\sum_{t=1}^{T} \frac{(1-\varepsilon_t)}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) \leq \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \frac{\varepsilon_t}{\lambda_t} F^\star
\end{aligned}$$

Now, if we choose $\varepsilon_t = \varepsilon$ for all t then we have:

$$\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) \leq \frac{1}{(1-\varepsilon)} \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \frac{\varepsilon}{((1-\varepsilon))\lambda_t} F^\star \tag{G.30}$$

From the previous lemma G.11.0.3, we have

$$\frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star \right) \leq \frac{2}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) + \frac{\varepsilon_t}{(1-\varepsilon_t)\lambda_t} F^\star \tag{G.31}$$

Summing up the equation (G.31) for $t = 1$ to T and for $\varepsilon_t = \varepsilon$, we have:

$$\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star \right) \leq \sum_{t=1}^{T} \frac{2}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) + \sum_{t=1}^{T} \frac{\varepsilon}{(1-\varepsilon)\lambda_t} F^\star \tag{G.32}$$

Now from equations (G.30) and (G.32),

$$\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star \right) \leq \sum_{t=1}^{T} \frac{2}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x^\star_{\lambda_t, x_{t-1}}) - F^\star \right) + \sum_{t=1}^{T} \frac{\varepsilon}{(1-\varepsilon)\lambda_t} F^\star \tag{G.33}$$

$$\leq \frac{2}{(1-\varepsilon)} \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \frac{3\varepsilon}{((1-\varepsilon))\lambda_t} F^\star \tag{G.34}$$

□

Proof of Theorem G.4.1.2. From the previous Lemma G.11.0.4, we have:

$$\sum_{t=1}^{T} \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star \right) \leq \frac{2}{(1-\varepsilon)} \|x^\star - x_0\|^2 + \sum_{t=1}^{T} \frac{3\varepsilon}{((1-\varepsilon))\lambda_t} F^\star$$

We assume that $F(x_t) \leq F(x_{t-1})$ for all t. This is fine to assume as we can always do the resampling if failed once. And also:

$$\frac{1}{\lambda_t} \left(F(x_t) - F^\star \right) \leq \frac{1}{\lambda_t} \left(F_{\lambda_t, x_{t-1}}(x_t) - F^\star \right) \quad \text{for all } t.$$

Hence,

$$F(x_T) - F^\star \leq \frac{2}{(1-\varepsilon)} \frac{\|x^\star - x_0\|^2}{\sum_{t=1}^{T} \frac{1}{\lambda_t}} + \frac{3\varepsilon}{1-\varepsilon} F^\star.$$

□

G.12 Adaptive Stochastic Trust Region Method

Algorithm 22 Adaptive Stochastic Trust Region Method

input $x_0 \in \mathbb{R}^d$, ε_0, μ and $m > 0$.
Compute $\|\nabla F(x_0)\|$, $F(x_0)$ and C_0
for $t = 1 \dots$ T **do**
  Compute regularizer λ_t using $\|\nabla f(x_{t-1})\|$, λ_t and μ.
  Compute radius r_t using $\|\nabla f(x_{t-1})\|$, $f(x_{t-1})$ and C_{t-1}.
  Computer error parameter ε_t using λ_t and μ, the strong convexity of F.
  Get $\tilde{F}_{\lambda_t, x_{t-1}}$ via Taylor Expansion.
  Compute the sensitivity for $\tilde{F}_{\lambda_t, x_{t-1}}$ using Theorem G.2.0.1.
  Local sensitivity based sampling of $\tilde{F}^s_{\lambda_t, x_{t-1}}(x)$ from $\tilde{F}_{\lambda_t, x_{t-1}}(x)$.
  $x_t \leftarrow \arg\min_{x \in B(r_t, x_{t-1})} F^s_{\lambda_t, x_{t-1}}(x)$.
  Compute $\|\nabla F(x_t)\|$, $F(x_t)$ and C_t.
end for
output x_T

We here now provide the detailed statemnt of Theorem G.5.0.1 and then provide the proof for it.

Theorem' G.2.2.1. *For a given set of constants C_k, δ_k and $\tilde{\varepsilon}_k = \delta_k \frac{\mu}{\lambda_k + \mu}$ which is error tolerance for the square approximation of the function $F_{\lambda_k, x_{k-1}}(x)$ for all $k \in [T]$, if λ_{k+1} is chosen as :*

$$2\lambda_{k+1} = \max\left(\sqrt{\frac{4C_k\|\nabla F(x_k)\|^3}{\frac{1}{4c^2}\|\nabla F(x_k)\|^2 + 4\tilde{\delta}_{k+1}\mu\frac{F(x_k)}{3m}}} - \mu, \mu\right),$$

then with probability $\geq 1/2$ the following holds:

$$F(x_{k+1}) - F^\star \leq (1 + 2\varepsilon_{k+1})\frac{2\lambda_{k+1}}{2\lambda_{k+1} + \mu}(F(x_k) - F^\star) + 2\varepsilon_{k+1}F^\star, \tag{G.35}$$

where $\varepsilon_{k+1} = 2\tilde{\varepsilon}_{k+1}\left(1 + \frac{1}{m}\right)$ $\forall$ k, m and c are positive constants.

Proof of Theorem G.5.0.1. Let us first reiterate the notations:

$$\tilde{F}_{\lambda_{k+1}, x_k}(x) = f(x_k) + (x - x_k)^\top A^\top \alpha_{x_k} + \frac{\gamma}{2}\|x\|^2 + (x - x_k)^\top A^\top H_{x_k} A(x - x_k) + \lambda\|x - x_k\|^2.$$

and $F_{\lambda_{k+1}, x_k}(x) = \tilde{F}_{\lambda_{k+1}, x_k}(x) + B_{x_k}(x)\|x - x_k\|^3$. We can write $\tilde{F}_{\lambda_{k+1}, x_k}(x) = \frac{1}{n}\sum_{i=1}^n \tilde{f}_i(x^T a_i)$ where $\tilde{f}_i(x^T a_i)$ is the quadratic approximation of $f_i(x^T a_i)$ around the point x_k. We also

define the upper bound on the radius $r_{k+1} = \frac{\|\nabla F(x_k)\|}{2\lambda_{k+1}+\mu}$. Contribution in B_{x_k} comes from each term f_i *i.e.* $B_{x_k}(x) = \frac{1}{n}\sum_{i=1}^{n} B_{x_k}^{(i)}(x)$. Let us assume that x_{k+1} is the point, we get after minimizing the subset after sampling from the sensitivity of the quadratic approximation. To make proof simpler in this section, we assume $C_k^{(i)}$ as the upper bound on the absolute value of $B_{x_k}^{(i)}(x)\ \forall\ i \in [n]$ in the ball $B(x_k, r_{k+1})$ *i.e.* $C_k^{(i)} \cdot r_{k+1}^3 \geq \max_{x\in B(x_k,r_{k+1})} \left|f_i(x^T a_i) - \tilde{f}_i(x^T a_i)\right|\ \forall\ i \in [n]$ where $C_k^{(i)}$ is a positive real number. We have $C_k = \frac{1}{n}\sum_{i=1}^{n} C_k^{(i)}$.
As we have already defined for all x:

$$\left|\tilde{F}_{\lambda_{k+1},x_k}(x) - F_{\lambda_{k+1},x_k}(x)\right| \leq C_k\|x-x_k\|^3.$$

So if $\tilde{F}^s_{\lambda_{k+1},x_k}(x)$ is sampled by sensitivities with error parameter $\tilde{\varepsilon}_{k+1}$ we have by triangle inequality:

$$\begin{aligned}\left|\tilde{F}^s_{\lambda_{k+1},x_k}(x) - F_{\lambda_{k+1},x_k}(x)\right| &\leq \left|\tilde{F}^s_{\lambda_{k+1},x_k}(x) - \tilde{F}_{\lambda_{k+1},x_k}(x)\right| + \left|\tilde{F}_{\lambda_{k+1},x_k}(x) - F_{\lambda_{k+1},x_k}(x)\right| \\ &\leq C_k\|x-x_k\|^3 + \tilde{\varepsilon}_{k+1}\tilde{F}_{\lambda_{k+1},x_k}(x) \\ &\leq C_k\|x-x_k\|^3 + \frac{\tilde{\varepsilon}_{k+1}}{1-\tilde{\varepsilon}_{k+1}} \cdot \tilde{F}^s_{\lambda_{k+1},x_k}(x). \end{aligned} \tag{G.36}$$

Hence, with very high probability, we do have :

$$\begin{aligned} F_{\lambda_{k+1},x_k}(x) &\leq C_k\|x-x_k\|^3 + \frac{\tilde{\varepsilon}_{k+1}}{1-\tilde{\varepsilon}_{k+1}} \cdot \tilde{F}^s_{\lambda_{k+1},x_k}(x) + \tilde{F}^s_{\lambda_{k+1},x_k}(x) \\ &= C_k\|x-x_k\|^3 + \frac{1}{1-\tilde{\varepsilon}_{k+1}}\tilde{F}^s_{\lambda_{k+1},x_k}(x) \end{aligned} \tag{G.37}$$

Now, we would like to show that letting $x_k^s = \arg\min_{x\in B(x_k,r_{k+1})} \tilde{F}^s_{\lambda_{k+1},x_k}(x)$, the error can still be controlled.

If, we let x_k^s be the minimizer of $\tilde{F}^s_{\lambda_{k+1},x_k}(x)$ and $x^\star_{\lambda_{k+1},x_k}$ be the minimizer of $F_{\lambda_{k+1},x_k}(x)$. We assume that $\frac{r_{k+1}}{c} \leq \|\tilde{x}_k^s - x_k\| \leq r_{k+1}$ and $\|x^\star_{\lambda_{k+1},x_k} - x_k\| \leq r_{k+1}$ for some positive real constant $c > 1$. We have :

$$\begin{aligned} F_{\lambda_{k+1},x_k}(x_k^s) &\leq \frac{1}{1-\tilde{\varepsilon}_{k+1}}\tilde{F}^s_{\lambda_{k+1},x_k}(x) + C_k\|x_k^s - x_k\|^3 \\ &\leq \frac{1}{1-\tilde{\varepsilon}_{k+1}}F^s_{\lambda_{k+1},x_k}(x^\star_{\lambda_{k+1},x_k}) + C_k\|x_k^s - x_k\|^3 \end{aligned} \tag{G.38}$$

where the second line follows the fact that x_k^s minimizes $\tilde{F}^s_{\lambda_{k+1},x_k}(x)$.

Hence, if we set $\tilde{\varepsilon}_{k+1} \leq 1/2$ plugging back everything together:

$$F_{\lambda_{k+1},x_k}(x_k^s) \leq (1+4\tilde{\varepsilon}_{k+1})F^\star_{\lambda_{k+1},x_k} + 4C_k r_{k+1}^3. \tag{G.39}$$

where in the last line we use that both $\|\tilde{x}_k^s - x_k\| \leq r_{k+1}$ and $\|x^\star_{\lambda_{k+1},x_k} - x_k\| \leq r_{k+1}$.

We have from Lemma G.11.0.1 that:

$$F^\star_{\lambda_{k+1},x_k} \leq \frac{2\lambda_{k+1}}{\mu+2\lambda_{k+1}}(F(x_k)-F^\star) + F^\star.$$

Plugging this bound into (G.39) gives:

$$F_{\lambda_{k+1},x_k}(x_k^s) \leq \frac{(1+4\tilde{\varepsilon}_{k+1})2\lambda_{k+1}}{\mu+2\lambda_{k+1}}(F(x_k)-F^\star) + F^\star + 4C_k r_{k+1}^3. \tag{G.40}$$

Now consider if we make the update $x_k^s = x_{k+1}$. Then we have using the simple bound that $F(x) \leq F_{\lambda_{k+1},x_k}(x)$ for all x:

$$\begin{aligned}
& F(x_{k+1}) = F_{\lambda_{k+1},x_k}(x_{k+1}) - \lambda_{k+1}\|x_{k+1}-x_k\|^2 \\
\Rightarrow\ & F(x_{k+1}) \leq \frac{(1+4\tilde{\varepsilon}_{k+1})2\lambda_{k+1}}{\mu+2\lambda_{k+1}}(F(x_k)-F^\star) + F^\star + 4C_k r_{k+1}^3 - \lambda_{k+1}\|x_{k+1}-x_k\|^2 \\
& \leq \frac{(1+4\tilde{\varepsilon}_{k+1})2\lambda_{k+1}}{\mu+2\lambda_{k+1}}(F(x_k)-F^\star) + F^\star + 4C_k r_{k+1}^3 - \frac{\lambda_{k+1}}{c^2} r_{k+1}^2
\end{aligned}$$

In the last line we have used $\|x_{k+1}-x_k\| \geq \frac{r_{k+1}}{c}$. Now, we do want to choose our parameters such that the following holds for some positive constant $m > 0$:

$$4C_k r_{k+1}^3 - \frac{\lambda_{k+1}}{c^2} r_{k+1}^2 \leq \frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu}\frac{4\tilde{\varepsilon}_{k+1}}{m}(F(x_k)-F^\star) + 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)F^\star \tag{G.41}$$

We provide the condition on λ in the next lemma:

Now if the condition given in equation (G.41) holds then the following recursion holds:

$$F(x_{k+1})-F^\star \leq \left(1+4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)\right)\frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu}(F(x_k)-F^\star) + 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)F^\star \tag{G.42}$$

We can compare the recursion equations given in equations (G.43) and (G.23). If we choose $\varepsilon_{k+1} = 2\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)$, then we have:

$$F(x_{k+1})-F^\star \leq (1+2\varepsilon_{k+1})\frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu}(F(x_k)-F^\star) + 2\varepsilon_{k+1}F^\star \tag{G.43}$$

which also confirms coreset conditions for the original function F.

□

Lemma G.12.0.1. *For a given set of constants $C_k^{(i)} \geq |B_{x_k}^{(i)}(x)|$, $x \in B(x_k, r_{k+1})$ such that $C_k = \frac{1}{n}\sum_{i=1}^{n} C_k^{(i)}$, and $\varepsilon_k = \delta_k \frac{\mu}{2\lambda_k+\mu}$ for $\delta_k \in (0,1/2)$ and $\forall\, k \in [T]$, we have ,*

$$4C_k r_{k+1}^3 - \frac{\lambda}{c^2} r_{k+1}^2 \leq \frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu} \frac{4\tilde{\varepsilon}_{k+1}}{m}(F(x_k) - F^\star) + 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)F^\star$$

is satisfied if for positive constants c and m:

$$2\lambda_{k+1} = \max\left(\sqrt{\frac{4C_k\|\nabla F(x_k)\|^3}{\frac{1}{4c^2}\|\nabla F(x_k)\|^2 + 4\tilde{\delta}_{k+1}\mu\frac{F(x_k)}{3m}}} - \mu, \mu\right).$$

Proof. We need to ensude the following condition:

$$4C_k r_{k+1}^3 - \frac{\lambda}{c^2} r_{k+1}^2 \leq \frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu} \frac{4\tilde{\varepsilon}_{k+1}}{m}(F(x_k) - F^\star) + 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)F^\star \tag{G.44}$$

Let us assume that there exist a positive real number θ_{k+1}.

- Consider the case when $F(x_k) \geq \theta_{k+1}F^\star$. Hence to ensure the condition given in equation (G.44), we can just ensure that the following holds:

$$4C_k r_{k+1}^3 - \frac{\lambda_{k+1}}{c^2} r_{k+1}^2 \leq \frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu} \frac{4\tilde{\varepsilon}_{k+1}}{m}\left(1-\frac{1}{\theta_{k+1}}\right)F(x_k) \tag{G.45}$$

- Consider the case when $F(x_k) \leq \theta_{k+1}F^\star$. Then, to ensure the condition given in equation (G.44), we can just ensure that the following holds:

$$4C_k r_{k+1}^3 - \frac{\lambda_{k+1}}{c^2} r_{k+1}^2 \leq 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)\frac{F(x_k)}{\theta_{k+1}} \tag{G.46}$$

In equations (G.45) and (G.46), we use $\theta_{k+1} = 1 + (m+1)\frac{2\lambda_{k+1}+\mu}{2\lambda_{k+1}}$ then we get the following condition to be satisfied:

$$\begin{aligned}
&4C_k r_{k+1}^3 - \frac{\lambda_{k+1}}{c^2} r_{k+1}^2 \leq 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)\frac{F(x_k)}{\theta_{k+1}} \\
\Rightarrow\ &4C_k r_{k+1}^3 \leq \frac{\lambda_{k+1}}{c^2} r_{k+1}^2 + 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)\frac{F(x_k)}{\theta_{k+1}} \\
\Rightarrow\ &4C_k \frac{\|\nabla F(x_k)\|^3}{(2\lambda_{k+1}+\mu)^3} \leq \frac{1}{2c^2}\frac{2\lambda_{k+1}}{2\lambda_{k+1}+\mu}\frac{\|\nabla F(x_k)\|^2}{2\lambda_{k+1}+\mu} + 4\tilde{\varepsilon}_{k+1}\left(1+\frac{1}{m}\right)\frac{F(x_k)}{\theta_{k+1}}
\end{aligned} \tag{G.47}$$

Now we assume that $2\lambda_k \geq \mu \ \forall \ k \Rightarrow \frac{2\lambda_k}{2\lambda_k+\mu} \geq \frac{1}{2}$ and $\tilde{\varepsilon}_{k+1} = \tilde{\delta}_{k+1}\frac{\mu}{2\lambda_{k+1}+\mu}$. Hence the condition given in the equation (G.47) is satisfied when:

$$4C_k\frac{\|\nabla F(x_k)\|^3}{(2\lambda_{k+1}+\mu)^3} \leq \frac{1}{4c^2}\frac{\|\nabla F(x_k)\|^2}{2\lambda_{k+1}+\mu} + 4\tilde{\delta}_{k+1}\frac{\mu}{2\lambda_{k+1}+\mu}\left(1+\frac{1}{m}\right)\frac{F(x_k)}{\theta_{k+1}}$$

$$\Rightarrow \ 2\lambda_{k+1}+\mu \geq \sqrt{\frac{4C_k\|\nabla F(x_k)\|^3}{\frac{1}{4c^2}\|\nabla F(x_k)\|^2 + 4\tilde{\delta}_{k+1}\mu\left(1+\frac{1}{m}\right)\frac{F(x_k)}{\theta_{k+1}}}}$$

Now in the above equation we put the value of $\theta_{k+1} = 1+(m+1)\frac{2\lambda_{k+1}+\mu}{2\lambda_{k+1}} \leq 2m+3$. We also use the fact that $m+1 \geq \frac{1}{3}(2m+3)$. That means the other conditions on λ_{k+1} are satisfied when

$$2\lambda_{k+1}+\mu \geq \sqrt{\frac{4C_k\|\nabla F(x_k)\|^3}{\frac{1}{4c^2}\|\nabla F(x_k)\|^2 + 4\tilde{\delta}_{k+1}\mu\frac{F(x_k)}{3m}}}$$

Hence, given

$$2\lambda_{k+1} = \max\left(\sqrt{\frac{4C_k\|\nabla F(x_k)\|^3}{\frac{1}{4c^2}\|\nabla F(x_k)\|^2 + 4\tilde{\delta}_{k+1}\mu\frac{F(x_k)}{3m}}} - \mu, \mu\right),$$

the conditions mentioned in the lemma are satisfied. □

Appendix H

Explicit Regularization of Stochastic Gradient Methods through Duality

Anant Raj[1], Francis Bach[2]

1 – MPI for Intelligent Systems, Tübingen

2 – Inria, Ecole Normale Supérieure, PSL Research University, Paris.

Abstract

We consider stochastic gradient methods under the interpolation regime where a perfect fit can be obtained (minimum loss at each observation). While previous work highlighted the implicit regularization of such algorithms, we consider an explicit regularization framework as a minimum Bregman divergence convex feasibility problem. Using convex duality, we propose randomized Dykstra-style algorithms based on randomized dual coordinate ascent. For non-accelerated coordinate descent, we obtain an algorithm which bears strong similarities with (non-averaged) stochastic mirror descent on specific functions, as it is is equivalent for quadratic objectives, and equivalent in the early iterations for more general objectives. It comes with the benefit of an explicit convergence theorem to a minimum norm solution. For accelerated coordinate descent, we obtain a new algorithm that has better convergence properties than existing stochastic gradient methods in the interpolating regime. This leads to accelerated versions of the perceptron for generic ℓ_p-norm regularizers, which we illustrate in experiments.

H.1 Introduction

With the recent advancement in machine learning and hardware research, the size and capacity of training models for machine learning tasks have been consistently increasing.

For many models which are widely used in practice, e.g., deep neural networks [81] and non-parametric regression models [25, 136], the training process achieves zero error, which means that such models are expressive enough to interpolate the training data completely. Hence, it is important to understand the interpolation regime to improve the training and prediction of such complex and over-parameterized models used in machine learning.

It is a well known fact that regularization, either explicit or implicit, plays a crucial role in achieving better generalization. While Tikhonov regularization is amongst the most famous form of regularization [80, 260] for linear or non-linear problems, several other methods can induce regularization in form of computational regularization when training machine learning models [207, 227, 268]. Apart from explicitly induced regularization in machine learning models, optimization algorithms like (stochastic) gradient descent which is widely used in practice while training large machine learning models, also induce implicit regularization in the obtained solution. In many cases, (stochastic) gradient descent converges to minimum Euclidean norm solutions. Recent series of papers [15, 84, 117, 225] present result about introducing implicit regularization/bias by (stochastic) gradient descent in different set of convex and non-convex problems.

In this paper, we address the following question: instead of relying on implicit regularization properties of stochastic algorithms, can we introduce an *explicit* regularization/bias while training over-parameterized models in the interpolation regime?

In optimization terms, the interpolation regime corresponds to the minimization of an average of finitely many functions of the form

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta)$$

with respect to $\theta \in \mathbb{R}^d$, where there is a global minimizer of F, which happens to be a global minimizer of *all* functions f_i, for $i \in \{1,\dots,n\}$ (instead of only minimizing their average). In the interpolation regime, we are thus looking for a point $\theta \in \mathbb{R}^d$ in the intersection of all sets of minimizers

$$\mathcal{K}_i = \arg\min_{\eta \in \mathbb{R}^d} f_i(\eta),$$

for all $i \in \{1,\dots,n\}$.

We can thus explicitly regularize the problem by solving the following optimization problem:

$$\min_{\theta \in \mathbb{R}^d} \psi(\theta) \text{ such that } \forall i \in \{1,\dots,n\},\ \theta \in \mathcal{K}_i, \tag{H.1}$$

where ψ is a regularization function (typically a squared norm). In the reformulated problem given in Eq. (H.1), explicit regularization can be induced in the solution via the

structure of the function ψ. Note also that the above problem can be seen as problem of generalized projection onto sets, which are convex if the original functions f_i's are convex, which we assume throughout this paper.

To address the problem defined in Eq. (H.1), we use the tools from convex duality and accelerated randomized coordinate ascent, which result in Dykstra-style projection algorithms [32, 75, 270]. In this paper, we make the following contributions:

(a) We provide a generic inequality going from dual guarantees in function values to primal guarantees in terms of Bregman divergences of iterates.

(b) For non-accelerated coordinate ascent, we obtain an algorithm which bears strong similarities with (non-averaged) stochastic mirror descent on specific functions f_i's. Our algorithm comes with the benefit of an explicit convergence theorem to a minimum value of the regularizer.

(c) For accelerated coordinate ascent, we obtain a new algorithm that has better convergence properties than existing stochastic gradient methods in the interpolating regime. While we indeed use the classical accelerated randomized coordinate descent algorithm to get accelerated rates, we show that we do not need any of the strong assumption that previous attempts at acceleration were needing (e.g.,Vaswani *et al.* [253]) for SGD in interpolation regime.

(d) This leads to accelerated versions of the perceptron for generic ℓ_p-norm regularizers (this is already an improvement for the ℓ_2-regularizer).

H.1.1 Related work

Stochastic gradient methods. First order stochastic gradient based iterative approaches [54, 60, 114, 165, 259] are the most efficient methods to perform optimization for machine learning problems with large datasets. There has been a large amount of work done in the area of stochastic first order optimization methods [see, e.g., 158, 165, 191, 192, and references therein] since the original stochastic approximation approach was proposed by Robbins and Monro [206].

Primal SGD in the interpolation regime. To address the optimization problem in the interpolation regime, Vaswani *et al.* [253] provide faster convergence rates for first order stochastic methods in the Euclidean geometry. They propose a strong growth condition, and a more widely applicable weak growth condition, under which stochastic gradient descent algorithm achieves fast convergence rate while using constant learning rate (a side contribution of our paper is to extend the latter algorithm to stochastic mirror descent). Vaswani *et al.* [254] propose to use line-search to set the step-size while training over-parameterized models which can fit completely to data. Several other works propose to use constant learning rate for stochastic gradient methods [20, 37, 140, 147] while training

extremely expressive models which interpolate. However, all of the above mentioned works are primal-based algorithms.

Dysktra's projection algorithms. Dykstra-type projection algorithms [32, 75] are simple modifications of the classical alternating projections methods [85, 255] to project on the intersection of convex sets. A key interpretation is the connection between Dykstra's algorithm and block coordinate ascent [22, 23, 242], which we use in this paper. Chambolle *et al.* [38] provide accelerated rates for Dykstra projection algorithm when projecting on the intersection of two sets.

Coordinate descent. Coordinate descent has a long history in the optimization literature [248, 249, 251]. Rates for accelerated randomized coordinate descent were first proved by Nesterov [171]. Since then, various extensions of the accelerated coordinate descent including proximal accelerated coordinate descent and non-uniform sampling have been proposed by Allen-Zhu *et al.* [11], Hendrikx *et al.* [88], Lin *et al.* [139], Nesterov and Stich [177]. Dual coordinate ascent can also be used to solve regularized empirical risk minimization problem [216, 218]. We recover some of their results as a by-product in this paper.

Perceptron. The perceptron is one of the oldest machine learning algorithms [27, 156]. Since then, there has been a lot of work on theoretical and empirical foundations of perceptron algorithms [69, 214, 246], in particular, with related extensions to ours, to ℓ_p-norm perceptron through mirror maps [83, 115]. However, none of the above mentioned work forces structure to the optimal solution in an explicit way.

H.2 Optimization Algorithms for Finite Data

We consider the finite data setting, that is, we will give bounds on training objectives (or distances to the minimum norm interpolator on the training set). We thus consider the problem:

$$\min_{\theta \in \mathbb{R}^d} \Psi(\theta) \text{ such that } \forall i \in \{1, \dots, n\},\ x_i^\top \theta \in \mathcal{Y}_i, \tag{H.2}$$

where:

- Regularizer / mirror map: $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a differentiable μ-strongly convex function with respect to some norm $\|\cdot\|$ (which is not in general the ℓ_2-norm). We will consider in this paper the associated Bregman divergence [33] defined as

$$D_\Psi(\theta, \eta) = \psi(\theta) - \psi(\eta) - \psi'(\eta)^\top (\theta - \eta).$$

- Data: $x_i \in \mathbb{R}^{d\times k}$, $\mathcal{Y}_i \subset \mathbb{R}^k$ are closed convex sets, for $i \in \{1,\dots,n\}$.
- Feasibility / interpolation regime: we make the assumption that there exists $\theta \in \mathbb{R}^d$ such that $\psi(\theta) < \infty$ and $\forall i \in \{1,\dots,n\}$, $x_i^\top \theta \in \mathcal{Y}_i$.

This is a general formulation that includes any set $\mathcal{K}_i$ like in the introduction (by having $k = d$, $x_i = I$, and $\mathcal{Y}_i = \mathcal{K}_i$), with an important particular case $k = 1$ (classical linear prediction).

In this paper, we consider primarily the ℓ_p-norm set-up, where $\psi(\theta) = \frac{1}{2}\|\theta\|_p^2$ for $p \in (1,2]$, which is $(p-1)$-strongly convex with respect to the ℓ_p-norm [19, 62]. The simplex with the entropy mirror map, which is 1-strongly convex with respect to the ℓ_1-norm, could also be considered.

H.2.1 From dual guarantees to primal guarantees

We can use Fenchel duality to obtain a dual problem for the problem given in Eq.(H.2). We will need the support function $\sigma_{\mathcal{Y}_i}$ of the convex set $\mathcal{Y}_i$, defined as, for $\alpha_i \in \mathbb{R}^k$ [30],

$$\sigma_{\mathcal{Y}_i}(\alpha_i) = \sup_{y_i \in \mathcal{Y}_i} y_i^\top \alpha_i.$$

We have, by Fenchel duality:

$$\min_{\theta\in\mathbb{R}^d} \psi(\theta) \text{ such that } \forall i \in \{1,\dots,n\},\ x_i^\top \theta \in \mathcal{Y}_i \tag{H.3}$$

$$= \min_{\theta\in\mathbb{R}^d} \psi(\theta) + \frac{1}{n}\sum_{i=1}^n \max_{\alpha_i\in\mathbb{R}^k} \Big\{ \alpha_i^\top x_i^\top \theta - \sigma_{\mathcal{Y}_i}(\alpha_i) \Big\}$$

$$= \max_{\forall i,\ \alpha_i\in\mathbb{R}^k} -\frac{1}{n}\sum_{i=1}^n \sigma_{\mathcal{Y}_i}(\alpha_i) - \psi^\star\Big(-\frac{1}{n}\sum_{i=1}^n x_i\alpha_i\Big), \tag{H.4}$$

with, at optimality,

$$\theta^\star = \theta(\alpha^\star) = \nabla\psi^\star\Big(-\frac{1}{n}\sum_{i=1}^n x_i\alpha_i\Big).$$

We denote by $G(\alpha)$ the dual objective function above. With our assumptions of feasibility and strong-convexity of ψ, there is a unique minimizer $\theta^\star \in \mathbb{R}^d$. The dual problem is bounded from above, and we assume that there exists a maximizer $\alpha^\star \in \mathbb{R}^{n\times k}$.

In this paper, we will consider dual algorithms to solve the problem disccused earlier in this section, that naturally leads to guarantees on $\text{gap}(\alpha) = G(\alpha^\star) - G(\alpha)$. Our first result is to provide some primal guarantees from $\theta(\alpha)$.

Proposition H.2.1.1. *With our assumption, for any $\alpha \in \mathbb{R}^{n\times k}$, we have:*

$$D_\Psi(\theta^\star, \theta(\alpha)) \leqslant \text{gap}(\alpha).$$

In the above statement, we also assume that ψ is differentiable everywhere, since Bregman divergences are well defined for differentiable functions. However, if we want to relax the above statement for a general function ψ which might not be differentiable, we would need to replace the term $D_\Psi(\theta^\star, \theta(\alpha))$ in Eq. (H.2.1.1) with $\psi(\theta^\star) - \psi(\theta(\alpha))) - \langle \partial\psi(\theta(\alpha))), \theta^\star - \theta(\alpha))\rangle$ where $\partial\psi(\theta(\alpha)))$ is a specific sub-gradient of ψ at point $\theta(\alpha)$. In the proof of Proposition H.2.1.1, we simply use the duility structure of the problem with Fenchel-Young inequality. See the detailed proof in Appendix H.6.

This result relates primal rate of convergence and dual rate of convergence, and holds true irrespective of the algorithm used to optimize the dual objective. Using it, we can recover convergence guarantees for stochastic dual coordinate ascent (SDCA) [216] and accelerated SDCA [218]. Compared to their analysis, our result directly provides rates of convergence from existing results in coordinate descent, but in terms of primal iterates. Details are provided in Appendix H.8.

Overall, we limit our discussion to convex functions however there is no requirement of the linear model to be used. Generalization of Proposition 1 (Proposition 2) holds for general convex objective and can be extended to non-linear models without extra effort.

H.2.2 Randomized coordinate descent

This result relates primal rate of convergence and dual rate of convergence, and holds true irrespective of the algorithm used to optimize the dual objective. Using it, we can recover convergence guarantees for stochastic dual coordinate ascent (SDCA) [216] and accelerated SDCA [218]. Compared to their analysis, our result directly provides rates of convergence from existing results in coordinate descent, but in terms of primal iterates. Details are provided in Appendix H.8.

Overall, we limit our discussion to convex functions however there is no requirement of the linear model to be used. Generalization of Proposition 1 (Proposition 2) holds for general convex objective and can be extended to non-linear models without extra effort.

H.2.3 Randomized coordinate descent

Given our relationship between primal iterate sub-optimality and dual sub-optimality gap $\text{gap}(\alpha)$ for any dual variable α and its corresponding primal variable $\theta(\alpha)$, we can leverage good existing algorithms on the dual problem. One such well known method is randomized dual coordinate descent, where α and thus $\theta(\alpha)$ will be random.

The algorithm is initialized with $\alpha_i^{(0)} = 0$ for all $i \in \{1, \ldots, n\}$, and at step $t > 0$, an index $i(t) \in \{1, \ldots, n\}$ is selected uniformly (for simplicity) at random. The update for proximal randomized coordinate ascent [202] is obtained in the following lemma (whose proof is given in Appendix H.6.1).

Algorithm 23 Proximal Random Coordinate Ascent

Input: α_0, $\theta_0 \leftarrow \theta(\alpha_0)$ and $\mathbf{x}_i, \mathcal{Y}_i$ for $i \in [n]$.
Initialize: $z_0 \leftarrow \alpha_0$, $\theta_{z_0} \leftarrow \theta_0$, $v_0 \leftarrow \alpha_0$ and $\gamma_0 \leftarrow \frac{1}{n}$.
for $t = 1$ **to** T **do**
  Choose $i_t \in \{1, 2, \cdots, n\}$ randomly.
  $\beta_{(\text{prev})} = \alpha_{i(t)}^{(t-1)}$.
  $\zeta_t = \Pi_{\mathcal{Y}_i}\Big(\frac{L_{i(t)}}{n}\alpha_{i(t)}^{(t-1)} + x_{i(t)}^\top \theta_{t-1}\Big)$.
  $\alpha_{i(t)} = \alpha_{i(t)}^{(t-1)} + \frac{n}{L_{i(t)}} x_{i(t)}^\top \theta_{t-1} - \frac{n}{L_{i(t)}}\zeta_t$.
  $\Delta_\beta = \alpha_{i(t)} - \beta_{(\text{prev})}$.
  Update $\theta_{t+1} \leftarrow \theta(\alpha_{t+1})$ {Use $\Delta_\beta, x_{i(t)}$}.
end for
return θ_{T+1} and α_{T+1}.

Lemma H.2.3.1. *For any uniformly randomly selected coordinate $i(t)$ at time instance t, the update for randomized proximal coordinate ascent is equal to*

$$\alpha_{i(t)} = \alpha_{i(t)}^{(t-1)} + \frac{n}{L_{i(t)}} x_{i(t)}^\top \theta(\alpha^{(t-1)}) - \frac{n}{L_{i(t)}} \Pi_{\mathcal{Y}_i}\Big(\frac{L_{i(t)}}{n}\alpha_{i(t)}^{(t-1)} + x_{i(t)}^\top \theta(\alpha^{(t-1)})\Big),$$

where $\Pi_{\mathcal{Y}_i}$ is the orthogonal projection on $\mathcal{Y}_i$, and L_i is equal to $L_i = \frac{1}{\mu}\|x_i\|_{2\to\star}^2 = \frac{1}{\mu}\sup_{\|\beta_i\|_2=1}\|x_i\beta_i\|_\star^2$.

Here, we implicitly assume that the individual projections on convex set $\mathcal{Y}_i$ for all $i \in \{1, \cdots, n\}$ are easy to compute, leading to Algorithm 23. For uniformly random selection of the datapoint $x_{i(t)}$ at time t, $L_i(t)$ can simply be replaced by $\max_i L_i$ in the algorithm.

Proximal randomized coordinate descent is a well studied problem [177, 202], and has a known rate of convergence for smooth objective functions. The set of optimal solutions of the dual problem in Equation (H.4) is denoted by $A^\star$ and $\alpha^\star$ is an element of it. Define,

$$\mathcal{R}(\alpha) = \max_y \max_{\alpha^\star \in A^\star} \{\|y - \alpha^\star\| \ : \ G(y) \geq G(\alpha)\}.$$

Since we assumed that ψ is μ-strongly convex, $\psi^\star$ is $(\frac{1}{\mu})$-smooth, and we get

$$\mathbb{E}\Big[D_\Psi(\theta^\star, \theta(\alpha^{(t)}))\Big] \leqslant \mathbb{E}\big[\text{gap}(\alpha^{(t)})\big] \leqslant \frac{\max_i L_i}{t}\frac{\max\{\|\alpha^\star\|^2, \mathcal{R}(0)^2\}}{n}, \tag{H.5}$$

where L_i is defined in Lemma H.2.3.1. The convergence rate given in Eq. (H.5) can further

be improved with non-uniform sampling based on the values L_i, and then $\max_i L_i$ can be replaced by $\frac{1}{n}\sum_{i=1}^n L_i$ [202]. However, taking inspirations from Cutkosky [50], Kavis *et al.* [111] the convergence for averaged iterate of coordinate descent when $\mathcal{Y}_i$ is a singleton set for all i can be obtained which only depends on $\|\alpha^\star\|$.

H.2.4 Relationship to least-squares

We now discuss an important case of the above formulation when $\mathcal{Y}_i$ is a singleton set, i.e., $\mathcal{Y}_i = \{y_i\}$. This problem has been addressed recently by Calatroni *et al.* [36] and we recover it as a special case of our general formulation.

We will make a link with least-squares in the interpolation regime, which can be written as a finite sum objective as follows,

$$\min \left[\frac{1}{2n} \sum_{i=1}^n \|y_i - x_i^\top \theta\|_2^2 = \frac{1}{2n} \sum_{i=1}^n d(x_i^\top \theta, \mathcal{Y}_i)^2 \right]. \tag{H.6}$$

It turns out that primal stochastic mirror descent with constant step-size applied to Eq. (H.6) and our formulation provided in Appendix H.2.1 are equivalent, as we now show.

Lemma H.2.4.1. *Consider the stochastic mirror descent updates using the mirror map ψ for the least-squares problem provided in Eq. (H.6). Then, the corresponding stochastic mirror descent updates converges to minimum ψ solution.*

Proof. Consider the primal-dual formulation given in Eq. (H.3) and Eq. (H.4), with $\mathcal{Y}_i = \{y_i\}$. The randomized dual coordinate ascent has the following update rule:

$$\alpha_{i(t)}^{(t)} = \alpha_{i(t)}^{(t-1)} + \frac{n}{L_{i(t)}} (x_{i(t)}^\top \theta(\alpha^{(t-1)}) - y_{i(t)}). \tag{H.7}$$

From the first order optimality condition, the update in Eq. (H.24) translates into, with $\theta^{(t)} = \theta(\alpha^{(t)})$,

$$\psi'(\theta^{(t)}) = \psi'(\theta^{(t-1)}) - \frac{1}{L_{i(t)}} x_{i(t)} (x_{i(t)}^\top \theta(\alpha^{(t-1)}) - y_{i(t)}),$$

which is exactly stochastic mirror descent on the least-squares objective with mirror map ψ. Hence the result. □

The rate of convergence can be obtained by the use of Eq. (H.5).

General case (beyond singletons). For any set $\mathcal{Y}_i$, if $\alpha_{i(t)}^{(t-1)} = 0$, for example, if $i(t)$ has never been selected, then, by Moreau's identity, we also get a stochastic mirror descent

Algorithm 24 Accelerated Proximal Coordinate Ascent (Dual Perceptron) [88, 139]

Input: α_0, $\theta_0 \leftarrow \theta(\alpha_0)$, x_i for $i \in [n]$ and $\mu = 0$.
Initialize: $z_0 \leftarrow \alpha_0$, $\theta_{z_0} \leftarrow \theta_0$, $v_0 \leftarrow \alpha_0$ and $\gamma_0 \leftarrow \frac{1}{n}$.
for $t = 1$ **to** $T-1$ **do**
  Choose $i_t \in \{1, 2, \cdots, n\}$ randomly.
  $r_t = 1 - \theta_{z_t}^\top x_{i_t}$
  $\alpha_{t+1} = u_{t+1} = \alpha_t + \frac{r_t}{n\gamma_t L_{i_t}}$.
  $\alpha_{i(t)}^{(t+1)} = \max(\alpha_{i(t)}^{(t+1)}, 0)$.
  Update $\theta_{t+1} \leftarrow \theta(\alpha_{t+1})$. (Algorithm 25)
  $\gamma_{t+1} = \frac{1}{2}\left(\sqrt{\gamma_t^4 + 4\gamma_t^2} - \gamma_t^2\right)$.
  $v_{t+1} = z_t + n\gamma_t(\alpha_{t+1} - \alpha_t)$.
  $z_{t+1} = (1-\gamma_{t+1})v_{t+1} + \gamma_{t+1}\alpha_{t+1}$.
  Update $\theta_{z_{t+1}} \leftarrow \theta(z_{t+1})$. (Algorithm 26
end for
return θ_{T+1} and α_{T+1}.

step for $\frac{1}{2n}\sum_{i=1}^n d(x_i^\top \theta, y_i)^2$. However, this is not true anymore when an index is selected twice.

H.2.5 Accelerated coordinate descent

In the previous sections, we discussed randomized coordinate dual ascent to optimize the problem in Eq. (H.3). We can also consider accelerated proximal randomized coordinate ascent [11, 88, 139]. For our problem, it leads to:

$$\mathbb{E}\Big[D_\Psi(\theta^\star, \theta(\alpha^{(t)}))\Big] \leqslant \mathbb{E}\Big[\text{gap}(\alpha^{(t)})\Big] \leqslant \frac{4\max_i L_i}{t^2}\left\{\frac{G(\alpha^\star) - G(0)}{\max_i L_i} + \frac{1}{2}\|\alpha^\star\|^2\right\}. \quad \text{(H.8)}$$

We will use the bound in Eq. (H.8) to analyze the general perceptron in the next section. We also provide the proximal accelerated randomized coordinate ascent algorithm [88, 139] with uniformly random sampling of coordinates to optimize the dual objective of ℓ_p-perceptron in Algorithm 25. However, the algorithm can easily be updated for the general case of Eqs. (H.3) and (H.4).

Note here that accelerated stochastic method for over-parametrized models in Algorithm 25 achieves Nesterov's fast rate without making explicit assumptions on the growth condition of the function and have the same computational overhead as that of primal SGD.

Algorithm 25 Update $\boldsymbol{\theta}_{t+1}$

Input: x_{i_t},α_{t+1} , $X^\top \alpha_t$, α_t and i_t.
$X^\top \alpha_{t+1} = X^\top \alpha_t + (\alpha_{i(t)}^{(t+1)} - \alpha_{i(t)}^{(t)}) x_{i_t}$.
Compute θ_{t+1} from $X^\top \alpha_{t+1}$.
return θ_{t+1} and $X^\top \alpha_{t+1}$.

Algorithm 26 Update $\boldsymbol{\theta}_{\mathbf{z}_{t+1}}$

Input: x_{i_t}, α_{t+1}, $X^\top \alpha_t$, $X^\top \alpha_{t+1}$, $X^\top z_t$, α_t, γ_t, γ_{t+1}.
$X^\top v_{t+1} = X^\top z_t + n\gamma_t X^\top (\alpha_{t+1} - \alpha_t)$.
$X^\top z_{t+1} = (1-\gamma_{t+1}) X^\top v_{t+1} + \gamma_{t+1} X^\top \alpha_{t+1}$.
Compute $\theta_{z_{t+1}}$ from $X^\top z_{t+1}$.
return $\boldsymbol{\theta}_{\mathbf{z}_{t+1}}$ and $X^\top z_{t+1}$.

H.2.6 Baseline: Primal Mirror Descent

We will compare our dual algorithms to existing primal algorithms. They correspond to the minimization of

$$F(\theta) = \frac{1}{2n} \sum_{i=1}^{n} d(x_i^\top \theta, y_i)^2. \tag{H.9}$$

Vaswani *et al.* [253] showed convergence of stochastic gradient descent for this problem. We extend their results to all mirror maps. Mirror descent with the mirror map ψ selects $i(t)$ at random and the iteration update is

$$\psi'(\theta^{(t)}) = \psi'(\theta^{(t-1)}) - \gamma x_{i(t)} (\Pi_{\mathcal{Y}_i}(x_{i(t)}^\top \theta^{(t-1)}) - x_{i(t)}^\top \theta^{(t-1)}). \tag{H.10}$$

Note that we have already encountered it in Lemma H.2.4.1, for least-squares regression, where we provided a convergence rate on the final iterate.

In Theorem H.2.6.1 below, we prove an $O(1/t)$ convergence rate for stochastic mirror descent update with mirror map ψ, for a constant step-size and the average iterate, directly extending the result of Vaswani *et al.* [253] to all mirror maps.

Theorem H.2.6.1. *Consider the stochastic mirror descent update in Eq. (H.10) for the optimization problem in Eq. (H.9) with $\gamma = \mu / \sup_i \|x_i\|_{2\to\star}^2$, the expected optimization error after t iterations the for averaged iterate $\bar{\theta}_t$ behaves as,*

$$0 \leqslant \mathbb{E}[F(\bar{\theta}^{(t)})] \leqslant \frac{\max_i L_i}{t} \psi(\theta^\star).$$

We provide the proof in Appendix H.6.3. The result is also applicable to general expectations and any form of convex objectives in the interpolation regime. We use this extension as one of our baseline in our experiments. In practice, as mentioned earlier, the

update for mirror descent in Eq. (H.10) is similar to randomized dual coordinate ascent update in Lemma H.2.3.1, in particular in early iterations (and not surprisingly, they behave similarly). Note here the difference in guarantees for the final iterates (which we get through a dual analysis) and the guarantees for the averaged iterate (which we get through a primal analysis).

H.3 ℓ_p-perceptrons

So far, we have discussed very general formulations for optimization problems in the interpolation regime. In this section, we discuss a specific problem which is widely used for linear binary classification, known as the perceptron algorithm, which is guaranteed to converge for linearly separable data. Here, we view the generalized ℓ_p-norm perceptron algorithm from the lens of our primal-dual formulation.

We consider $(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}$ for $i \in \{1, \cdots, n\}$, and the problem of minimizing $\psi(\theta)$ such that $\forall i, y_i x_i^\top \theta \geqslant 1$, which can be written as $\tilde{x}_i^\top \theta \geqslant 1$, where $\tilde{x}_i = y_i x_i$ for all $i \in \{1, \ldots, n\}$. For this section, we will be limiting ourselves to $\psi(\theta) = \frac{1}{2}\|\theta\|_p^2$ for $p \in (1, 2]$. We know that $\psi(\theta) = \frac{1}{2}\|\theta\|_p^2$ for $p \in (1, 2]$ is $(p-1)$-strongly convex with respect to the ℓ_p-norm. In this section, we denote $X \in \mathbb{R}^{n \times d}$ the data matrix $X = (\tilde{x}_1^\top; \tilde{x}_2^\top; \cdots; \tilde{x}_n^\top)$. Our generic primal optimization problem turns into:

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2}\|\theta\|_p^2 \text{ such that } X\theta \geqslant 1, \tag{H.11}$$

The dual problem is here

$$\max_{\alpha \in \mathbb{R}_+^n} -\frac{1}{2}\left\| \frac{-1}{n} \sum_{i=1}^n x_i \alpha_i \right\|_q^2 + \frac{1}{n} \sum_{i=1}^n \alpha_i, \tag{H.12}$$

where $\|\cdot\|_q$ is dual norm of $\|\cdot\|_p$, with $1/p + 1/q = 1$. At optimality, θ can be obtained from $X^\top \alpha$ as

$$\theta_j = \frac{1}{n}\|X^\top \alpha\|_q^{2-q} (X^\top \alpha)_j^{q-1},$$

where we define $u^{q-1} = |u|^{q-1}\text{sign}(u)$.

The function $\alpha \mapsto \frac{1}{2}\|X^\top \alpha\|_q^2$ is smooth, and the regular smoothness constant with respect to the i-th variable which is less than $L_i = \frac{1}{p-1}\|x_i\|_q^2$. We can apply here the results from Proposition H.2.1.1 to get the convergence in primal iterates for the the general ℓ_p-norm perceptron formulation in Eq. (H.11), while optimizing the dual function via accelerated coordinate ascent in Eq. (H.12).

Corollary H.3.0.1. *For the generalized ℓ_p-norm perceptron described in our primal-dual*

framework in Equations (H.11) *and* (H.12)*, we have*

$$\mathbb{E}\Big[\|\theta(\alpha)-\theta^\star\|_p\Big] \le \sqrt{\frac{2\mathbb{E}[\text{gap}(\alpha)]}{p-1}}.$$

Proof. The result comes from the application of Proposition H.2.1.1 in the generalized ℓ_p-norm perceptron from setting Eq. (H.11), with $D_{\frac{1}{2}\|\cdot\|_p^2}(\theta^\star,\theta) \ge \frac{p-1}{2}\|\theta-\theta^\star\|_p^2$. □

If we use accelerated randomized coordinate descent to optimize dual objective given in Eq. (H.12), then after t number of iterations, we get:

$$\mathbb{E}\Big[\|\theta_t-\theta^\star\|_p\Big] \le \frac{2\sqrt{2}\max_i\|x_i\|_q}{\sqrt{(p-1)t}}\sqrt{\frac{G(\alpha^\star)-G(0)}{\max_i\|x_i\|_q}+\frac{1}{2}\|\alpha^\star\|^2}, \tag{H.13}$$

where $\theta_t=\theta(\alpha_t)$.

Mistake bound. Since, we have the bound on the distance between primal iterate to its optimum, we can simply derive the mistake bound for our algorithm which we prove in Appendix H.7.

Lemma H.3.0.2. *For the generalized ℓ_p-norm perceptron described in our primal-dual framework in Equations* (H.11) *and* (H.12)*, we make no mistakes on training data on average after*

$$t > \frac{2\sqrt{2}R^2}{\sqrt{p-1}}\sqrt{\frac{G(\alpha^\star)-G(0)}{R}+\frac{1}{2}\|\alpha^\star\|^2}$$

steps where $R=\max_i\|x_i\|_q$ and $\|\cdot\|_q$ is the dual norm of $\|\cdot\|_p$.

The accelerated coordinate descent algorithm to solve the ℓ_p-perceptron is given in Algorithm 25. More details about the relationship between primal and dual variables, as well as dual ascent update for random coordinate descent for general ℓ_p-norm perceptron, e.g., the dual problem in Eq. (H.12), is given in Appendix H.7. Mistake bounds for the classical ℓ_p-perceptron are also recalled in Appendix H.7.

Baseline: primal mirror descent. We consider the finite sum minimization with stochastic mirror descent update and mirror map $\psi=\frac{1}{2}\|\cdot\|_p^2$ as discussed in Section H.2.6, that is, the finite sum minimization in Eq. (H.9) with $f_i(\theta)=\frac{1}{2}(1-\theta^\top x_i)_+^2$.

Corollary H.3.0.3. *Consider the finite sum minimization of $f(\theta)=\frac{1}{2n}\sum_{i=1}^n(1-\theta^\top x_i)_+^2$ via stochastic mirror descent with mirror map $\psi(\cdot)=\frac{1}{2}\|\cdot\|_p^2$, then on average, the proportion of mistakes on the training set is less than $\sqrt{\frac{\|\theta^\star\|_p^2R^2}{(p-1)t}}$ where $R=\max_i\|x_i\|_q$.*

Proof. The proof comes directly from Theorem H.2.6.1 and from the fact that the proportion of mistakes on the training set is less than the square root of the excess risk. □

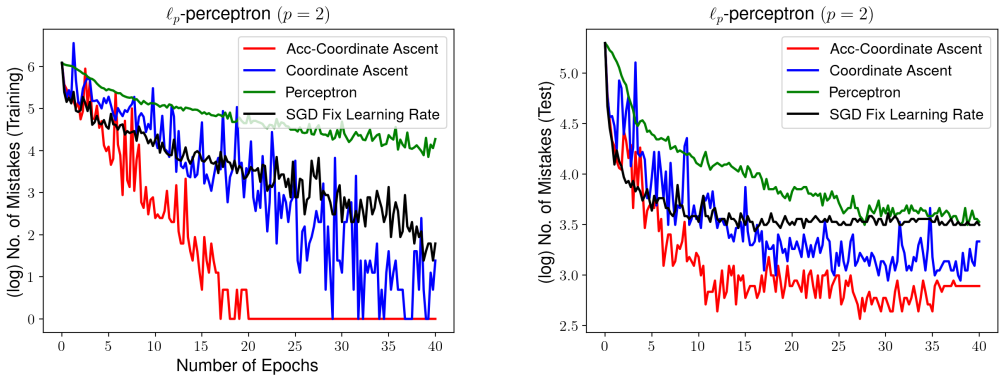
Similar bounds on the proportion of mistakes can also be obtained while optimizing $f(\theta) = \frac{1}{n}\sum_{i=1}^{n}(1 - x_i^\top \theta)_+$ via stochastic mirror descent with mirror map $\frac{1}{2}\|\cdot\|_p^2$. However, while tuning the step size, it requires the knowledge of $\|\theta^\star\|_p$, hence we do not include it in our base line.

We can compare the minimum number of iterations required to achieve no further mistakes while training in Lemma H.3.0.2 and Corollary H.3.0.3 to get the conditions on optimal primal and dual optimal variables under which our method (which has a better dependence in the number of iterations t) performs better than the baseline. We discuss these in the Appendix H.7. In our empirical evaluationin Section H.4, dual accelerate coordinate ascent significantly outperforms primal mirror descent.

Special Case of ℓ_1-perceptron. Our goal in this specific case is to solve the following sparse problem,

$$\theta_0 = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{2}\|\theta\|_1^2 \text{ such that } X\theta \geqslant 1. \tag{H.14}$$

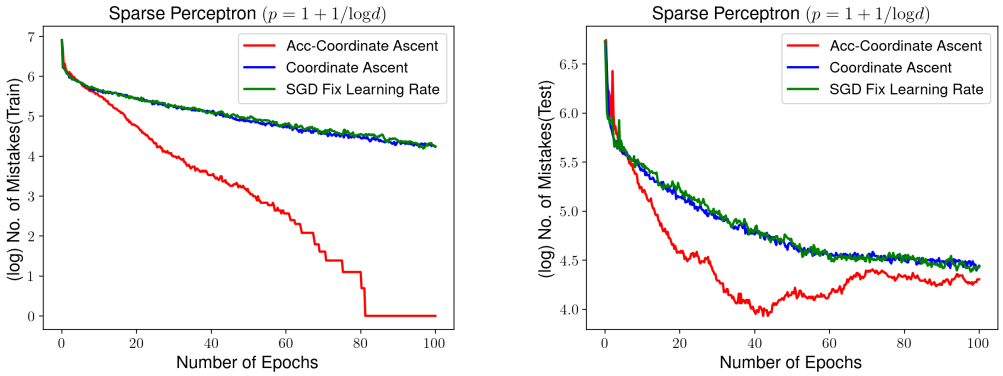
$\|\cdot\|_1$ is not strongly convex, hence we can not fit this problem to our formulation. However, following Duchi *et al.* [62], we solve the problem in (H.11) with $p = 1 + \frac{1}{\log d}$ where d is the dimension.



(a) Number of mistakes on the training test (in log scale).

(b) Number of mistakes on the test (in log scale).

Figure H.1: Experimental results for ℓ_2-perceptron



(a) Number of mistakes on the training (in log scale).

(b) Number of mistakes on the test (in log scale).

Figure H.2: Experimental results for sparse perceptron.

H.4 Experiments

In this section, we provide empirical evaluation for the methods discussed in this paper with the ℓ_p-perceptron. We generate data from a Gaussian distribution in dimension $d = 2000$, which we describe below. We consider two settings of p for our experiments, $p = 2$ which is usual perceptron, and $p = 1 + \frac{1}{\log d}$, which is the sparse perceptron setting.

Data generation. We generate $n = 1000$ inputs $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$ with $d = 2000$ from a Gaussian distribution centered at 0 and covariance matrix Σ which is a diagonal matrix. Similarly, we generate a random $d = 2000$ prediction vector θ sampled again from the normal distribution.

For ℓ_2-perceptron, the i-th eigenvalue for Σ is $1/i^{3/2}$ and for sparse perceptron i-th eigenvalue for Σ, is $1/i$. We compute the prediction vector y_i for x_i as follows, $y_i = \text{sign}(x_i^\top \theta + b)$ where we fix $b = 0.005$. We also remove those pair of (x_i, y_i) from the data for which we have $x_i^\top \theta + b \leq 0.1$. We generate 1000 train examples and 1000 test examples for both settings. For the sparse perceptron case, we make the prediction vector θ sparse by randomly choosing 50 entries to be non zero. We then compute the prediction vector similar to the ℓ_p-perceptron case, $y_i = \text{sign}(x_i^\top \theta + b)$ where we fix $b = 0.005$ and remove those pair of (x_i, y_i) from the data for which we have, $x_i^\top \theta + b \leq 0.1$.

Baseline. For the ℓ_2-perceptron, we compare accelerated coordinate descent and randomized coordinate descent with the perceptron and primal SGD [253]. For the sparse perceptron, we compare the accelerated coordinate descent and randomized coordinate descent with extension of primal SGD to stochastic mirror descent case (discussed in section H.2.6 with $f_i = \frac{1}{2}(1 - x_i^\top \theta)_+^2$) with mirror map $\psi(\cdot) = \frac{1}{2}\|\cdot\|_p^2$ where $p = 1 + \frac{1}{\log d}$.

Note that we compare to non-averaged SGD (for which we provide a new proof), which works significantly better than averaged SGD.

Comparisons for the ℓ_2-perceptron and sparse perceptron are given in Figures H.1 and H.2 respectively.

We can make the following observations:

(a) From both the training plots (Figure H.1a and Figure H.2a), it is clear that we gain significantly in training performance over primal SGD and the perceptron if we optimize the dual with accelerated randomized coordinate ascent method, which supports our theoretical claims made in Section H.3.

(b) For testing errors, we also see gains for our accelerated perceptron, which is not supported by theoretical arguments. This gives motivation to further study this algorithm for general expectations.

(c) Note that in the semi-log plots, we observe an affine behavior of the training errors, highlighting exponential convergence. This can be explained by a strongly convex dual problem (since the matrix $XX^\top$ is invertible), and could be quantified using usual convergence rates for coordinate ascent for strongly-convex objectives.

H.5 Conclusion

In this paper, we proposed algorithms that are explicitly regularizing solutions of an interpolation problem. This is done through a dual approach, and, with acceleration, it improves over existing algorithms. Several natural questions are worth exploring: (1) Can we explicitly characterize linear convergence in the dual (like observed in experiments), with or without regularization? (2) How are our algorithms performing beyond the interpolation regime, where the dual become unbounded but some primal information can typically be recovered in Dykstra-style algorithms [23]? (3) Can we extend our approach to saddle-point formulations such as proposed by Kundu *et al.* [118]? Can we prove any improvement in the general population regime, where we aim at bounds on testing data?

In a recent series of papers [64, 190], it was shown that strong duality holds for different neural network architecture and there might be a possibility to extend our approach for those architectures which are beyond the convexity assumptions. It is a promising future research direction and needs to be investigated further.

Proofs for Main Results

H.6 Primal-Dual Structure

Apart from the notations discussed in the main paper, we would further use the following notation for data matrix $X \in \mathbb{R}^{n\times d}$ such that $X = \left[x_1^\top; \cdots; x_n^\top\right]$. We consider the following general primal and its corresponding dual problem which appear very frequently in machine learning domain.

$$\min_{\theta\in\mathbb{R}^d} \left[\mathcal{O}_P(\theta) := \psi(\theta) + \frac{1}{n}\sum_{i=1}^n \phi_i(x_i^\top\theta)\right] \tag{H.15}$$

$$\max_{\alpha\in\mathbb{R}^n} \left[\mathcal{O}_D(\alpha) := -\psi^*\left(-\frac{1}{n}X^\top\alpha\right) - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i)\right]. \tag{H.16}$$

Here, we assume that $\phi : \mathbb{R}^d \to \mathbb{R}$ and $\psi : \mathbb{R}^d \to \mathbb{R}$ are smooth convex function for all i. We have the following first order optimality conditions for the equivalent problems given in Equations (H.15) and (H.16):

$$\begin{aligned} &x_i^\top\theta \in \partial\phi_i^*(\alpha_i), \\ &\theta \in \partial\psi^*\left(-\frac{1}{n}X^\top\alpha\right), \end{aligned} \quad \text{and} \quad \begin{aligned} &\alpha_i \in \partial\phi_i(x_i^\top\theta), \\ &-\frac{1}{n}X^\top\alpha \in \partial\psi(\theta). \end{aligned} \tag{H.17}$$

From the duality, $\theta(\alpha) = \partial\psi^*\left(-\frac{1}{n}\sum_{i=1}^n \alpha_i x_i\right)$. We can recall Fenchel's Inequality: For any convex function f, the inequality $f(x) + f^*(\theta) \geq x^\top\theta$ holds for all $x \in \text{dom}(f)$ and $\theta \in \text{dom}(f^*)$. Equality holds if the following is satisfied $\theta \in \partial f(x)$.

From Fenchel's inequality, we have:

Proposition H.6.0.1. *Consider the general primal dual problem given in equations (H.15) and (H.16), dual sub-optimlaity gap $gap(\alpha) = [\mathcal{O}_D(\alpha^\star) - \mathcal{O}_D(\alpha)]$ at some α provides the upper bound on the Bregman divergence of ψ between $\theta^\star$ and $\theta(\alpha)$ i.e. $D_\Psi(\theta^\star, \theta(\alpha)) \leqslant \text{gap}(\alpha)$.*

Proof. The Bregman divergence with respect to mirror map ψ is

$$D_\Psi(x,y) = \psi(x) - \psi(y) - \langle\nabla\psi(y), x - y\rangle.$$

Now, we have:

$$\text{gap}(\alpha) = -\psi^*\left(-\frac{1}{n}X^\top\alpha^\star\right) + \psi^*\left(-\frac{1}{n}X^\top\alpha\right) - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i). \tag{H.18}$$

In the proof we would again use Fenchel's inequality which we used in the proof of previous theorem. From the optimality condition, we know that $-\frac{1}{n}X^\top \alpha \in \partial \psi(\theta(\alpha))$. Hence,

Hence,

$$
\begin{aligned}
\text{gap}(\alpha) &= -\psi^*\left(-\frac{1}{n}X^\top \alpha^\star\right) + \psi^*\left(-\frac{1}{n}X^\top \alpha\right) - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= -\left(-\left\langle \frac{1}{n}X^\top \alpha^\star, \theta^\star \right\rangle - \psi(\theta^\star)\right) + \left(-\left\langle \frac{1}{n}X^\top \alpha, \theta(\alpha) \right\rangle - \psi(\theta(\alpha))\right) \\
&\qquad - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= \psi(\theta^\star) - \psi(\theta(\alpha)) + \left\langle \frac{1}{n}X^\top \alpha^\star, \theta^\star \right\rangle - \left\langle \frac{1}{n}X^\top \alpha, \theta(\alpha) \right\rangle \\
&\qquad - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= \psi(\theta^\star) - \psi(\theta(\alpha)) + \left\langle \frac{1}{n}X^\top \alpha^\star, \theta^\star \right\rangle - \left\langle \frac{1}{n}X^\top \alpha, \theta(\alpha) \right\rangle \\
&\qquad - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= \psi(\theta^\star) - \psi(\theta(\alpha)) + \left\langle \frac{1}{n}X^\top \alpha^\star, \theta^\star \right\rangle + \left\langle \frac{1}{n}X^\top \alpha, \theta^\star \right\rangle - \left\langle \frac{1}{n}X^\top \alpha, \theta^\star \right\rangle \\
&\qquad - \left\langle \frac{1}{n}X^\top \alpha, \theta(\alpha) \right\rangle - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= \psi(\theta^\star) - \psi(\theta(\alpha)) - \left\langle \frac{1}{n}X^\top \alpha, \theta(\alpha) - \theta^\star \right\rangle + \left\langle \frac{1}{n}X^\top \alpha^\star - \frac{1}{n}X^\top \alpha, \theta^\star \right\rangle \\
&\qquad - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= \underbrace{\psi(\theta^\star) - \psi(\theta(\alpha)) - \langle \nabla\psi(\theta(\alpha)), \theta^\star - \theta(\alpha)\rangle}_{:= D_\Psi(\theta^\star, \theta(\alpha))} + \left\langle \frac{1}{n}X^\top \alpha^\star - \frac{1}{n}X^\top \alpha, \theta^\star \right\rangle \\
&\qquad - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= D_\Psi(\theta^\star, \theta(\alpha)) + \left\langle \frac{1}{n}\alpha^\star - \frac{1}{n}\alpha, X\theta^\star \right\rangle - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i) \\
&= D_\Psi(\theta^\star, \theta(\alpha)) + \frac{1}{n}\sum_{i=1}^n (\alpha_i^\star - \alpha_i)\cdot x_i^\top \theta^\star - \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^n \phi_i^*(\alpha_i)
\end{aligned}
$$

$$= D_\Psi(\theta^\star, \theta(\alpha)) - \frac{1}{n}\sum_{i=1}^{n}(\alpha_i - \alpha_i^\star)\cdot\nabla\phi^*(\alpha_i^\star) - \frac{1}{n}\sum_{i=1}^{n}\phi_i^*(\alpha_i^\star) + \frac{1}{n}\sum_{i=1}^{n}\phi_i^*(\alpha_i)$$

$$= D_\Psi(\theta^\star, \theta(\alpha)) + \frac{1}{n}\sum_{i=1}^{n} D_{\phi_i^*}(\alpha_i, \alpha_i^\star) \geq D_\Psi(\theta^\star, \theta(\alpha)). \tag{H.19}$$

□

After we provide the general result in Proposition H.6.0.1, we now provide the proof for proposition H.2.1.1 below. The result in statement is a useful result and can be useful in several ways. For example, the guarantees for SDCA [216, 218]. We provide the details in the Appendix H.8.

Proof of Proposition H.2.1.1. We can just use the result in Proposition H.6.0.1 to prove Proposition H.2.1.1. Let's recall once again the primal dual formulation of the problem which we have in Equation (H.3) and Equation (H.4).

$$\min_{\theta\in\mathbb{R}^d} D_\psi(\theta, \theta^{(0)}) \text{ such that } \forall i \in \{1,\ldots,n\},\ x_i^\top\theta \in \mathcal{Y}_i \tag{H.20}$$

$$= \min_{\theta\in\mathbb{R}^d} \psi(\theta) + \frac{1}{n}\sum_{i=1}^{n}\max_{\alpha_i\in\mathbb{R}^k}\Big\{\alpha_i^\top x_i^\top\theta - \sigma_{\mathcal{Y}_i}(\alpha_i)\Big\}$$

$$= \max_{\forall i,\ \alpha_i\in\mathbb{R}^k} -\frac{1}{n}\sum_{i=1}^{n}\sigma_{\mathcal{Y}_i}(\alpha_i) - \psi^\star\Big(-\frac{1}{n}\sum_{i=1}^{n}x_i\alpha_i\Big) \tag{H.21}$$

$$= \max_{\alpha\in\mathbb{R}^{n\times k}} G(\alpha),$$

Let $\mathcal{K}_i$ represents that set for all θ such that $x_i^\top\theta \in \mathcal{Y}_i$ and the indicator function $\iota_{\mathcal{K}_i}$ for a convex set $\mathcal{K}_i$ for all $\in \{1,\ldots,n\}$ is defined as $\iota_{\mathcal{K}_i}(x_i^\top\theta) = 0$ if $x_i^\top\theta \in \mathcal{Y}_i$ and $\iota_{\mathcal{K}_i}(x_i^\top\theta) = +\infty$, otherwise for all $\in \{1,\ldots,n\}$. We can write Equation (H.20) in the form of generalized equation given in Equation (H.15) considering $\phi_i(x_i^\top\theta) = \iota_{\mathcal{K}_i}(x_i^\top\theta)$. It is easy to see that $\phi_i^*(\alpha_i) = \sigma_{\mathcal{Y}_i}(\alpha_i)$. Hence, now the statement follows from Proposition H.6.0.1. □

H.6.1 Coordinate Descent Update: Proof of Lemma H.2.3.1

We have:

$$\alpha_{i(t)}^{(t)} = \arg\max_{\alpha_{i(t)}} -\frac{1}{n}\sigma_{\mathcal{Y}_{i(t)}}(\alpha_i) + \frac{1}{n}\nabla\psi^\star\Big(-\frac{1}{n}\sum_{i=1}^{n}x_i\alpha_i^{(t-1)}\Big)^\top x_{i(t)}[\alpha_{i(t)} - \alpha_{i(t)}^{(t-1)}]$$
$$- \frac{L_{i(t)}}{2n^2}\|\alpha_i - \alpha_{i(t)}^{(t-1)}\|_2^2$$

$$= \arg\max_{\alpha_{i(t)}} -\frac{1}{n}\sigma_{y_{i(t)}}(\alpha_i) + \frac{1}{n}\theta(\alpha^{(t-1)})^\top x_{i(t)}[\alpha_{i(t)} - \alpha_{i(t)}^{(t-1)}] - \frac{L_{i(t)}}{2n^2}\|\alpha_i - \alpha_{i(t)}^{(t-1)}\|_2^2$$

$$= \arg\min_{\alpha_{i(t)}} \sigma_{y_{i(t)}}(\alpha_i) + \frac{L_{i(t)}}{2n}\|\alpha_i - \alpha_{i(t)}^{(t-1)} - \frac{n}{L_{i(t)}}x_{i(t)}^\top \theta(\alpha^{(t-1)})\|_2^2. \tag{H.22}$$

The minimization problem in Equation (H.22) can be written as follows:

$$\begin{aligned}
&\min_{\alpha_{i(t)}} \left[\sigma_{y_{i(t)}}(\alpha_i) + \frac{L_{i(t)}}{2n}\|\alpha_i - \alpha_{i(t)}^{(t-1)} - \frac{n}{L_{i(t)}}x_{i(t)}^\top \theta(\alpha^{(t-1)})\|_2^2\right] \\
=&\min_{\alpha_{i(t)}} \left[\sigma_{y_{i(t)}}(\alpha_i) - \sup_z \left[(\alpha_i - \alpha_{i(t)}^{(t-1)} - \frac{n}{L_{i(t)}}x_{i(t)}^\top \theta(\alpha^{(t-1)}))^\top z + \frac{n}{2L_{i(t)}}\|z\|^2\right]\right] \\
=&\sup_{z\in\mathcal{Y}_{i(t)}} \left[-\frac{n}{2L_{i(t)}}\|z\|^2 + z^\top\left(\frac{n}{L_{i(t)}}x_{i(t)}^\top \theta(\alpha^{(t-1)}) + \alpha_{i(t)}^{(t-1)}\right)\right]
\end{aligned} \tag{H.23}$$

The above maximization problem has a solution at $z^\star = \Pi_{y_{i(t)}}\left(x_{i(t)}^\top \theta(\alpha^{(t-1)}) + \frac{L_{i(t)}}{n}\alpha_{i(t)}^{(t-1)}\right)$. However, $z^\star$ is also the solution of the following optimization formulation:

$$z^\star = \arg\max_z \left[(\alpha_i - \alpha_{i(t)}^{(t-1)} - \frac{n}{L_{i(t)}}x_{i(t)}^\top \theta(\alpha^{(t-1)}))^\top z + \frac{n}{2L_{i(t)}}\|z\|^2\right]$$

Comparing both the value of $z^\star$, we get the following update in $\alpha_{i(t)}$ in alternative form

$$\alpha_{i(t)} = \alpha_{i(t)}^{(t-1)} + \frac{n}{L_{i(t)}}x_{i(t)}^\top \theta(\alpha^{(t-1)}) - \frac{n}{L_{i(t)}}\Pi_{\mathcal{Y}_i}\left(\frac{L_{i(t)}}{n}\alpha_{i(t)}^{(t-1)} + x_{i(t)}^\top \theta(\alpha^{(t-1)})\right),$$

where $\Pi_{\mathcal{Y}_i}$ is the orthogonal projection on $\mathcal{Y}_i$.

H.6.2 Implicit Regularization of Stochastic Mirror Descent : Proof of Lemma H.2.4.1

Proof. Consider the primal-dual formulation given in Eq. (H.3) and Eq. (H.4), with $\mathcal{Y}_i = \{y_i\}$. The randomized dual coordinate ascent has the following update rule:

$$\alpha_{i(t)}^{(t)} = \alpha_{i(t)}^{(t-1)} + \frac{n}{L_{i(t)}}(x_{i(t)}^\top \theta(\alpha^{(t-1)}) - y_{i(t)}). \tag{H.24}$$

From the first order optimality condition, the update in Eq. (H.24) translates into, with $\theta^{(t)} = \theta(\alpha^{(t)})$,

$$\psi'(\theta^{(t)}) = \psi'(\theta^{(t-1)}) - \frac{1}{L_{i(t)}} x_{i(t)}(x_{i(t)}^\top \theta(\alpha^{(t-1)}) - y_{i(t)}),$$

which is exactly stochastic mirror descent on the least-squares objective with mirror map ψ. Hence the result. □

H.6.3 Mirror Descent: [Proof of Theorem H.2.6.1]

The convergence rate does depend on $\psi(\theta^\star)$ but this is not an explicit regularization. The proof goes as follows:

Mirror descent with the mirror map ψ selects $i(t)$ at random and the iteration is

$$\psi'(\theta^{(t)}) = \psi'(\theta^{(t-1)}) - \gamma x_{i(t)}(\Pi_{\mathcal{Y}_i}(x_{i(t)}^\top \theta^{(t-1)}) - x_{i(t)}^\top \theta^{(t-1)}).$$

Following the proof of Flammarion and Bach [67], we have for any $\theta \in \mathbb{R}^d$:

$$\begin{aligned} D_\psi(\theta, \theta^{(t)}) &= D_\psi(\theta, \theta^{(t)}) - D_\psi(\theta^{(t)}, \theta^{(t-1)}) + \gamma f_t'(\theta^{(t-1)})^\top(\theta^{(t)} - \theta) \\ &\leqslant D_\psi(\theta, \theta^{(t)}) - \frac{\mu}{2}\|\theta^{(t)} - \theta^{(t-1)}\|^2 + \gamma f_t'(\theta^{(t-1)})^\top(\theta^{(t-1)} - \theta) \\ &\quad + \gamma \|f_t'(\theta^{(t-1)})\|_\star \|\theta^{(t-1)} - \theta^{(t)}\| \\ &\leqslant D_\psi(\theta, \theta^{(t)}) - \gamma f_t'(\theta^{(t-1)})^\top(\theta^{(t-1)} - \theta) + \frac{\gamma^2}{2\mu}\|f_t'(\theta^{(t-1)})\|_\star^2. \end{aligned}$$

For $\theta = \theta^\star$ and using $\mathbb{E}\big[\|f_t'(\theta^{(t-1)})\|_\star^2\big] \leqslant \sup_i \|x_i\|_{2\to\star}^2 \big[f(\theta) - f(\theta^\star)\big]$, we get and taking expectations, we get:

$$\big(1 - \gamma \frac{\|x_i\|_{2\to\star}^2}{2\mu}\big)\mathbb{E}\big[f(\theta^{(t-1)}) - f(\theta^\star)\big] \leqslant \frac{1}{\gamma}\Big(\mathbb{E}[D_\psi(\theta^\star, \theta^{(t)})] - \mathbb{E}\big[D_\psi(\theta^\star, \theta^{(t-1)})\big]\Big).$$

Thus, with $\gamma = \mu / \sup_i \|x_i\|_{2\to\star}^2$, we get

$$\mathbb{E}\big[f(\theta^{(t-1)}) - f(\theta^\star)\big] \leqslant \frac{2}{\gamma}\Big(\mathbb{E}[D_\psi(\theta^\star, \theta^{(t)})] - \mathbb{E}\big[D_\psi(\theta^\star, \theta^{(t-1)})\big]\Big).$$

This leads to

$$\mathbb{E}\big[f(\bar{\theta}_t) - f(\theta^\star)\big] \leqslant \frac{2}{\gamma t} D_\psi(\theta^\star, \theta^{(0)}).$$

H.7 ℓ_p-perceptron

In this section, we provide proofs for the claims made in Section H.3.

We start with the proof of Lemma H.3.0.2.

Proof. For all i, $x_i^\top \theta^\star \geq 1$. Hence,

$$\begin{aligned} x_i^\top \theta_t &= x_i^\top \theta_t - x_i^\top \theta^\star + x_i^\top \theta^\star = x_i^\top \theta^\star - x_i^\top (\theta^\star - \theta_t) \\ &\geq 1 - x_i^\top (\theta^\star - \theta_t) \geq 1 - \|x_i\|_q \|\theta_t - \theta^\star\|_p \\ &\geq 1 - R\|\theta_t - \theta^\star\|_p. \end{aligned}$$

Assuming $\alpha_0 = 0$, from Equation (H.13), we have

$$\mathbb{E}\Big[\|\theta_t - \theta^\star\|_p\Big] \leq \frac{2\sqrt{2}\max_i \|x_i\|_q}{\sqrt{(p-1)t}} \sqrt{\frac{G(\alpha^\star) - G(0)}{\max_i \|x_i\|_q} + \frac{1}{2}\|\alpha^\star\|^2}$$

Now for on average for no mis-classification for all $i \in \{1, \cdots, n\}$,

$$1 \geq R\mathbb{E}\Big[\|\theta_t - \theta^\star\|_p\Big] \Rightarrow t \geq \frac{2\sqrt{2}R^2}{\sqrt{p-1}} \sqrt{\frac{G(\alpha^\star) - G(0)}{R} + \frac{1}{2}\|\alpha^\star\|^2}. \tag{H.25}$$

□

Mistake Bound ℓ_p-primal perceptron. If we apply mirror descent with the mirror map $\psi = \frac{1}{2}\|\cdot\|_p^2$ to the minimization of $\frac{1}{n}\sum_{i=1}^n (1 - \theta^\top x_i)_+$, then the iteration is

$$\psi'(\theta_t) = \psi'(\theta_{t-1}) - \gamma 1_{1 - \theta_{t-1}^\top x_{i(t)} > 0} x_{i(t)},$$

and we have

$$\frac{1}{n}\sum_{i=1}^n (1 - \bar{\theta}_t^\top x_i)_+ \leqslant \frac{\|\theta_\star\|_p^2}{2\gamma t} + \gamma \frac{\max_i \|x_i\|_q^2}{2(p-1)}.$$

The best γ is equal to $\gamma = \frac{\|\theta_\star\|_p}{\max_i \|x_i\|_q} \frac{\sqrt{p-1}}{\sqrt{t}}$, which does depend on too many things, and leads to a proportion of mistakes on the training set less than

$$\frac{\|\theta_\star\|_p \max_i \|x_i\|_q}{\sqrt{p-1}\sqrt{t}}.$$

H.7.1 Update for Random Coordinate Descent

We have:

$$\begin{aligned} & \min_{\theta\in\mathbb{R}^d} \frac{1}{2}\|\theta\|_p^2 \text{ such that } X\theta \geqslant 1 \\ = & \min_{\theta\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^n} \frac{1}{2}\|\theta\|_p^2 + \alpha^\top(1 - X\theta) \\ = & \max_{\alpha\in\mathbb{R}^n} -\frac{1}{2}\|X^\top\alpha\|_q^2 + \alpha^\top 1, \end{aligned}$$

where, at optimality, θ can be obtained from $X^\top\alpha$ as

$$\theta_j = \|X^\top\alpha\|_q^{2-q}(X^\top\alpha)_j^{q-1},$$

where we define $u^{q-1} = |u|^{q-1}\text{sign}(u)$.

The function $\frac{1}{2}\|X^\top\alpha\|_p^2$ is smooth, and the regular smoothness constant with respect to the i-th variable which is less than

$$L_i = \frac{1}{p-1}\|x_i\|_q^2.$$

A dual coordinate ascent step corresponds to choosing $i(t)$ and replacing $(\alpha_{t-1})_{i(t)}$ by

$$(\alpha_t)_i = \max\Big\{0, (\alpha_{t-1})_{i(t)} + \frac{1}{L_{i(t)}}\big(1 - \|X^\top\alpha_{t-1}\|_q^{2-q}\sum_{j=1}^d [(X^\top\alpha_{t-1})_j]^{q-1}X_{i(t)j}\Big\},$$

which can be interpreted as:

$$(\alpha_t)_i = \max\Big\{0, (\alpha_{t-1})_{i(t)} + \frac{1}{L_{i(t)}}\big(1 - \theta_{t-1}^\top x_{i(t)}\big)\Big\}.$$

H.7.2 ℓ_2-perceptron

The primal problem has the following dual form under the interpolation regime

$$\max_{\alpha\geq 0, \alpha\in\mathbb{R}^n} \alpha^\top 1 - \frac{1}{2}\|X\alpha\|^2.$$

We denote S_v as the set of support vectors *i.e.* S_v is the set of indices where $\alpha_j^\star \neq 0$. Hence, we also have $\tilde{x}_j^\top\theta^\star = 1$ for $j \in S_v$. α_{S_v} denotes the vector of non-zero entries in α. Correspondingly, X_{S_v} denotes the feature matrix for support vectors. From the first order

suboptimality condition we have,

$$\theta(\alpha) = \frac{1}{n} X\alpha.$$

We also know that for support vectors, $y_i \cdot x_i^\top \theta^\star = \tilde{x}_i^\top \theta^\star = 1$ for all $i \in S_v$. Also $\theta^\star = \frac{1}{n} X_{S_v} \alpha^\star_{S_v}$. Hence,

$$\frac{1}{n} X_{S_v}^\top X_{S_v} \alpha^\star_{S_v} = 1 \Rightarrow \alpha^\star_{S_v} = n(X_{S_v}^\top X_{S_v})^{-1} 1.$$

From Lemma H.3.0.2, we should have $t \geq \frac{2\sqrt{2}R^2}{\sqrt{p-1}} \sqrt{\frac{G(\alpha^\star)-G(0)}{R} + \frac{1}{2}\|\alpha^\star\|^2}$, for no training mistakes.

We now use Corollary H.3.0.3 to get mistake bound on the perceptron. To have no mistakes on average, the proportion of mistakes should be less than $1/n$. Hence,

$$\frac{R\|\theta^\star\|}{\sqrt{t}} \leq \frac{1}{n} \Rightarrow t \geq R^2 \|\theta^\star\|^2 n^2. \tag{H.26}$$

We already have $\alpha^\star_{S_v} = n(X_{S_v}^\top X_{S_v})^{-1} 1$.

$$\theta^\star = \frac{1}{n} X\alpha^\star = \frac{1}{n} X_{S_v} \alpha^\star_{S_v} = X_{S_v}(X_{S_v}^\top X_{S_v})^{-1} 1.$$

Finally we have the following:

$$\begin{aligned} \|\alpha^\star\| &= \|\alpha^\star_{S_v}\| = n\|(X_{S_v}^\top X_{S_v})^{-1} 1\| \\ \|\theta^\star\|^2 &= \|X_{S_v}(X_{S_v}^\top X_{S_v})^{-1} 1\|^2 = 1^\top (X_{S_v}^\top X_{S_v})^{-1} 1. \end{aligned} \tag{H.27}$$

Hence, one can compare the number of minimum iteration required by both the approaches.

H.8 (Accelerated) Stochastic Dual Coordinate Descent

Stochastic dual coordinate ascent [216] is a popular approach to optimize regularized empirical risk minimize problem. For this section, let $\phi_1, \cdots, \phi_n$ be a sequence of $\frac{1}{\gamma}$-smooth convex losses and let $\lambda > 0$ be a regularization parameter then consider following regularized empirical risk minimization problem:

$$\min_{\theta \in \mathbb{R}} \left[\mathcal{S}_P(\theta) := \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi_i(X_i^\top \theta) \right]. \tag{H.28}$$

Corresponding dual problem of the minimization problem given in equation (H.28) can be written similarly as:

$$\max_{\alpha \in \mathbb{R}^n} \left[\mathcal{S}_D(\alpha) := -\frac{\lambda}{2} \| \frac{1}{\lambda n} X^\top \alpha \|^2 - \frac{1}{n} \sum_{i=1}^{n} \phi_i^*(-\alpha_i) \right] \tag{H.29}$$

There is one to one relation between the smoothness constant and strong convexity parameter of primal and corresponding dual function. We prove the following result from Kakade *et al.* [105].

Theorem H.8.0.1 (Theorem 6, [105]). *Assume that f is a closed and convex function. Then f is β-strongly convex w.r.t. a norm $\|\cdot\|$ if and only if f^* is $\frac{1}{\beta}$-smooth w.r.t. the dual norm $\|\cdot\|_*$.*

From the above theorem it is clear that ϕ_i^* are γ-strongly convex. Hence the term $\frac{1}{n}\sum_{i=1}^n \phi_i^*(-\alpha_i)$ is $\frac{\gamma}{n}$ strongly convex. Similary coordinate wise smoothness $L_i = \frac{\|x_i\|^2}{\lambda n^2}$. Now, just as a direct implication of the result provided in Proposition H.6.0.1, we have the convergence result for SDCA [216] and accelerated stochastic dual coordinate ascent [218] which we provide in Corollary H.8.0.2 and Corollary H.8.0.3. For the next two results, we denote θ_k as $\theta(\alpha_k)$.

Corollary H.8.0.2 (Stochastic Dual Coordinate Ascent). *Consider the regularized empirical risk minimization problem given in equation (H.28), then if we run SDCA [216] algorithm starting from $\alpha_0 \in \mathbb{R}^n$ with a fix step size $1/\max_i L_i$ where $L_i = \frac{\|x_i\|^2}{\lambda n^2}$, primal iterate after k iterations converges as following:*

$$\frac{\lambda}{2}\|\theta_{k+1} - \theta^\star\|^2 \leq D(\alpha_{k+1}) \leq \left(1 - \frac{\gamma\lambda}{\max_i \|x_i\|^2}\right)^k (\mathcal{S}_D(\alpha_0) - \mathcal{S}_D(\alpha^\star)).$$

Proof. From Allen-Zhu *et al.* [11], it is clear that for μ-strongly convex and L_i-coordinate wise smooth convex function $\mathcal{S}_D(\alpha)$ where $\alpha \in \mathbb{R}^n$, randomized coordinate descent has the following convergence guarantee:

$$D(\alpha_{k+1}) \leq \left(1 - \frac{\mu}{n \max_i L_i}\right)^k (\mathcal{S}_D(\alpha_0) - \mathcal{S}_D(\alpha^\star)).$$

Here, $\mu = \frac{\gamma}{n}$. First part of the inequality directly comes from Proposition H.6.0.1 by the observation that here $\psi(\cdot) = \frac{\lambda}{2}\|\cdot\|^2$ and bregman divergence are always positive. □

Corollary H.8.0.3 (Accelerated Stochastic Dual Coordinate Ascent). *Consider the regularized empirical risk minimization problem given in equation (H.28), then if we run Accelerated SDCA [218] algorithm starting from $\alpha_0 \in \mathbb{R}^n$, we have following convergence*

rate for the primal iterates:

$$\frac{\lambda}{2}\|\theta_{k+1}-\theta^{\star}\|^2 \leq D(\alpha_{k+1}) \leq 2\left(1-\frac{\sqrt{\gamma\lambda}}{\sqrt{\max_i \|x_i\|^2}}\right)^k (\mathcal{S}_D(\alpha_0)-\mathcal{S}_D(\alpha^{\star})).$$

Proof. From Allen-Zhu *et al.* [11], it is clear that for μ-strongly convex and L_i-coordinate wise smooth convex function $\mathcal{S}_D(\alpha)$ where $\alpha \in \mathbb{R}^n$, accelerated randomized coordinate descent has the following convergence guarantee:

$$D(\alpha_{k+1}) \leq 2\left(1-\frac{\sqrt{\mu}}{n\sqrt{\max_i L_i}}\right)^k (\mathcal{S}_D(\alpha_0)-\mathcal{S}_D(\alpha^{\star})).$$

First part of the inequality directly comes from Proposition H.6.0.1 by the observation that here $\psi(\cdot)=\frac{\lambda}{2}\|\cdot\|^2$ and bregman divergence are always positive. Here $\mu=\frac{\gamma}{n}$ and $L_i=\frac{\|x_i\|^2}{\lambda n^2}$. □

Discussion. Let us denote duality gap at dual variable α as $\Delta(\alpha)$. From the definition of the duality gap $\Delta(\alpha)=\mathcal{S}_P(\theta(\alpha))-\mathcal{S}_D(\alpha)$. However, $\Delta(\alpha)$ is an upper bound on the primal sub-optimality gap as well on dual sub-optimality gap. The main difference in the analysis presented in our work with the works of Shalev-Shwartz and Zhang [216] and Shalev-Shwartz and Zhang [218] is that the we provide the guarantee in term of the iterate. However Shalev-Shwartz and Zhang [216] and Shalev-Shwartz and Zhang [218] provide convergence in terms of duality gap $\Delta(\alpha)$. Another main difference is that we use constant step size in each step and the output of our algorithm doesn't need averaging of the past iterates. Our analysis holds for the last iterate.

Notations

$f^\star$	Until and unless specified $f^\star = \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$.
$\mathbf{x}^\star$	$\arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$.
$\mathbf{e}_i$	Vector of appropriate dimension with all the entries set to 0 except i^{th} entry set to 1.
x_i	i^{th} entry of vector $\mathbf{x}$.
ML	Machine Learning
SGD	Stochastic Gradient Descent.
CD	Coordinate Descent
SCD	Steepest Coordinate Descecnt
RCD	Randomized Coordinate Descent
UCD	Uniformly Random Coordinate Descent
ASCD	Approximately Steepest Coordinate Descent
MP	Matching Pursuit
LMO	Linear Minimization Oracle
OLO	Online Linear Optimization
FTRL	Follow the Regularized Leader
AO-FTRL	Adaptive Optimistic Follow the Regularized Leader
MD	Mirror Descent
AO-MD	Adaptive Optimistic Mirror Descent

List of Figures

List of Algorithms

Bibliography

[1] Achlioptas, D. (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, **66**(4), 671 – 687. Special Issue on {PODS} 2001.

[2] Agarwal, N., Kakade, S., Kidambi, R., Lee, Y. T., Netrapalli, P., and Sidford, A. (2017). Leverage score sampling for faster accelerated regression and ERM. *arXiv:1711.08426*.

[3] Ahipaşaoğlu, S. D., Sun, P., and Todd, M. (2008). Linear Convergence of a Modified Frank–Wolfe Algorithm for Computing Minimum-Volume Enclosing Ellipsoids. *Optimization Methods and Software*, **23**(1), 5–19.

[4] Alain, G., Lamb, A., Sankar, C., Courville, A., and Bengio, Y. (2015). Variance Reduction in SGD by Distributed Importance Sampling. *arXiv.org*.

[5] Alaoui, A. and Mahoney, M. W. (2015). Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems 28 (NeurIPS)*, pages 775–783.

[6] Allen-Zhu, Z. (2017). Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205.

[7] Allen-Zhu, Z. and Hazan, E. (2016). Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707.

[8] Allen-Zhu, Z. and Orecchia, L. (2014). Linear Coupling of Gradient and Mirror Descent: A Novel, Simple Interpretation of Nesterov's Accelerated Method. *arXiv.org*.

[9] Allen-Zhu, Z. and Orecchia, L. (2017). Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. In C. H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67, pages 3:1–3:22.

[10] Allen-Zhu, Z. and Yuan, Y. (2016). Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In *International Conference on Machine Learning*, pages 1080–1089.

[11] Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. (2016a). Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119.

[12] Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. (2016b). Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling. In *ICML 2017 - Proceedings of the 34th International Conference on Machine Learning*, pages 1110–1119.

[13] Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. (2016c). Even faster accelerated coordinate descent using non-uniform sampling. In *ICML 2016 - Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *PMLR*, pages 1110–1119. PMLR.

[14] Altschuler, J., Bach, F., Rudi, A., and Weed, J. (2018). Massively scalable sinkhorn distances via the nystr\" om method. *arXiv preprint arXiv:1812.05189*.

[15] Arora, S., Cohen, N., Hu, W., and Luo, Y. (2019). Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pages 7411–7422.

[16] Atsushi Shibagaki, I. T. (2017). Stochastic Primal Dual Coordinate Method with Non-Uniform Sampling Based on Optimality Violations. *arXiv.org*.

[17] Avron, H., Kapralov, M., Musco, C., Musco, C., Velingker, A., and Zandieh, A. (2017). Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 253–262.

[18] Bachem, O., Lucic, M., and Krause, A. (2015). Coresets for nonparametric estimation-the case of DP-means. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*.

[19] Ball, K., Carlen, E. A., and Lieb, E. H. (1994). Sharp uniform convexity and smoothness inequalities for trace norms. *Inventiones mathematicae*, **115**(1), 463–482.

[20] Bassily, R., Belkin, M., and Ma, S. (2018). On exponential convergence of sgd in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*.

[21] Bauschke, H. H. and Combettes, P. L. (2011a). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer New York, New York, NY.

[22] Bauschke, H. H. and Combettes, P. L. (2011b). *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media.

[23] Bauschke, H. H. and Koch, V. R. (2015). Projection methods: Swiss army knives for solving feasibility and best approximation problems with halfspaces. *Contemp. Math*, **636**, 1–40.

[24] Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, **2**(1), 183–202.

[25] Belkin, M., Rakhlin, A., and Tsybakov, A. B. (2018). Does data interpolation contradict statistical optimality? *arXiv preprint arXiv:1806.09471*.

[26] Bishop, C. M. (2016). *Pattern Recognition and Machine Learning*. Springer-Verlag New York.

[27] Block, H.-D. (1962). The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, **34**(1), 123.

[28] Bock, R., Chilingarian, A., Gaug, M., Hakl, F., Hengstebeck, T., Jiřina, M., Klaschka, J., Kotrč, E., Savickỳ, P., Towers, S., *et al.* (2004). Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **516**(2-3), 511–528.

[29] Borwein, J. M. and Zhu, Q. (2005). *Techniques of Variational Analysis and Nonlinear Optimization*. Canadian Mathematical Society Books in Math, Springer New York.

[30] Boyd, S. and Vandenberghe, L. (2004a). *Convex optimization*. Cambridge university press.

[31] Boyd, S. P. and Vandenberghe, L. (2004b). *Convex optimization*. Cambridge University Press.

[32] Boyle, J. P. and Dykstra, R. L. (1986). A method for finding projections onto the intersection of convex sets in Hilbert spaces. In *Advances in Order Restricted Statistical Inference*, pages 28–47. Springer.

[33] Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, **7**(3), 200–217.

[34] Bubeck, S. (2014). *Theory of Convex Optimization for Machine Learning*.

[35] Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, **8**(3-4), 231–357.

[36] Calatroni, L., Garrigos, G., Rosasco, L., and Villa, S. (2019). Accelerated iterative regularization via dual diagonal descent. *arXiv preprint arXiv:1912.12153*.

[37] Cevher, V. and Vũ, B. C. (2019). On the linear convergence of the stochastic gradient method with constant step-size. *Optimization Letters*, **13**(5), 1177–1187.

[38] Chambolle, A., Tan, P., and Vaiter, S. (2017). Accelerated alternating descent methods for Dykstra-like problems. *Journal of Mathematical Imaging and Vision*, **59**(3), 481–497.

[39] Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The Convex Geometry of Linear Inverse Problems. *Foundations of Computational Mathematics*, **12**(6), 805–849.

[40] Chen, R., Menickelly, M., and Scheinberg, K. (2018). Stochastic optimization using a trust-region method and random models. *Mathematical Programming*, **169**(2), 447–487.

[41] Chen, S., Billings, S. A., and Luo, W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, **50**(5), 1873–1896.

[42] Chen, X., Lin, Q., and Pena, J. (2012). Optimal regularized dual averaging methods for stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 395–403.

[43] Chowdhury, A., Yang, J., and Drineas, P. (2018). An iterative, sketching-based framework for ridge regression. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 988–997.

[44] Clarkson, K. L. and Woodruff, D. P. (2017). Low-rank approximation and regression in input sparsity time. *Journal of the ACM (JACM)*, **63**(6), 54.

[45] Cohen, M. B., Lee, Y. T., Musco, C., Musco, C., Peng, R., and Sidford, A. (2015). Uniform sampling for matrix approximation. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science (ITCS)*.

[46] Cohen, M. B., Musco, C., and Musco, C. (2017). Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1758–1777. SIAM.

[47] Conn, A. R., Gould, N. I., and Toint, P. L. (2000). *Trust region methods*, volume 1. SIAM.

[48] Csiba, D. and Richtárik, P. (2016). Importance Sampling for Minibatches. *arXiv.org*.

[49] Csiba, D., Qu, Z., and Richtárik, P. (2015). Stochastic Dual Coordinate Ascent with Adaptive Probabilities. In *ICML 2015 - Proceedings of the 32th International Conference on Machine Learning*.

[50] Cutkosky, A. (2019). Anytime online-to-batch, optimism and acceleration. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1446–1454, Long Beach, California, USA. PMLR.

[51] d'Aspremont, A., Guzmán, C., and Jaggi, M. (2018). Optimal affine invariant smooth minimization algorithms. *SIAM Journal on Optimization (and arXiv:1301.0465)*.

[52] Davenport, M. A. and Wakin, M. B. (2010). Analysis of orthogonal matching pursuit using the restricted isometry property. *IEEE Transactions on Information Theory*, **56**(9), 4395–4401.

[53] Dawkins, B. (1991). Siobhan's problem: The coupon collector revisited. *The American Statistician*, **45**(1), 76–82.

[54] Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654.

[55] Dennis, Jr., J. and Torczon, V. (1991). Direct Search methods on parallel machines. *SIAM Journal on Optimization*, **1**(4), 448–474.

[56] Dennis, Jr, J. E. and Moré, J. J. (1977). Quasi-Newton methods, motivation and theory. *SIAM review*, **19**(1), 46–89.

[57] Dhillon, I. S., Ravikumar, P., and Tewari, A. (2011). Nearest Neighbor based Greedy Coordinate Descent. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*.

[58] Drineas, P., Mahoney, M. W., and Muthukrishnan, S. (2006). Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

[59] Drineas, P., Magdon-Ismail, M., Mahoney, M. W., and Woodruff, D. P. (2012). Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, **13**(December), 3475–3506.

[60] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, **12**, 2121–2159.

[61] Duchi, J., Jordan, M. I., and McMahan, B. (2013). Estimation, optimization, and parallelism when data is sparse. In *Advances in Neural Information Processing Systems*, pages 2832–2840.

[62] Duchi, J. C., Shalev-Shwartz, S., Singer, Y., and Tewari, A. (2010). Composite objective mirror descent. In *Conference on Learning Theory*, pages 14–26.

[63] El Halabi, M., Hsieh, Y.-P., Vu, B., Nguyen, Q., and Cevher, V. (2017). General proximal gradient method: A case for non-euclidean norms. Technical report.

[64] Ergen, T. and Pilanci, M. (2020). Training convolutional relu neural networks in polynomial time: Exact convex optimization formulations. *arXiv preprint arXiv:2006.14798*.

[65] Feldman, D. and Langberg, M. (2011). A unified framework for approximating and clustering data. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 569–578.

[66] Fercoq, O., Gramfort, A., and Salmon, J. (2015). Mind the duality gap: safer rules for the Lasso. In *ICML 2015 - Proceedings of the 32th International Conference on Machine Learning*, pages 333–342.

[67] Flammarion, N. and Bach, F. (2017). Stochastic composite least-squares regression with convergence rate o (1/n). *arXiv preprint arXiv:1702.06429*.

[68] Frank, M. and Wolfe, P. (1956). An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly*, **3**, 95–110.

[69] Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, **37**(3), 277–296.

[70] Frey, P. W. and Slate, D. J. (1991). Letter recognition using holland-style adaptive classifiers. *Machine learning*, **6**(2), 161–182.

[71] Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, **1**(2), 302–332.

[72] Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, **33**(1), 1–22.

[73] Frostig, R., Ge, R., Kakade, S., and Sidford, A. (2015). Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning*, pages 2540–2548.

[74] Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, **7**(3), 397–416.

[75] Gaffke, N. and Mathar, R. (1989). A cyclic projection algorithm via duality. *Metrika*, **36**(1), 29–54.

[76] Ghadimi, S. and Lan, G. (2013). Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, **23**(4), 2341–2368.

[77] Ghaoui, L. E., Viallon, V., and Rabbani, T. (2010). Safe Feature Elimination for the LASSO and Sparse Supervised Learning Problems. *arXiv.org*.

[78] Gillis, N. and Luce, R. (2018). A fast gradient method for nonnegative sparse regression with self-dictionary. *IEEE Transactions on Image Processing*, **27**(1), 24–37.

[79] Gillis, N., Kuang, D., and Park, H. (2015). Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, **53**(4), 2066–2078.

[80] Golub, G. H., Hansen, P. C., and O'Leary, D. P. (1999). Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, **21**(1), 185–194.

[81] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[82] Gribonval, R. and Vandergheynst, P. (2006). On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. *IEEE Transactions on Information Theory*, **52**(1), 255–261.

[83] Grove, A. J., Littlestone, N., and Schuurmans, D. (2001). General convergence results for linear discriminant updates. *Machine Learning*, **43**(3), 173–210.

[84] Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. (2018). Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*.

[85] Halperin, I. (1962). The product of projection operators. *Acta Sci. Math.(Szeged)*, **23**(1), 96–99.

[86] HaoChen, J. Z. and Sra, S. (2018). Random shuffling beats SGD after finite epochs. *arXiv preprint arXiv:1806.10077*.

[87] He, X. and Takáč, M. (2015). Dual Free Adaptive Mini-batch SDCA for Empirical Risk Minimization. *arXiv.org*.

[88] Hendrikx, H., Bach, F., and Massoulié, L. (2019). An accelerated decentralized stochastic proximal algorithm for finite sums. In *Advances in Neural Information Processing Systems*, pages 952–962.

[89] Hofmann, T., Lucchi, A., Lacoste-Julien, S., and McWilliams, B. (2015). Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pages 2305–2313.

[90] Holloway, C. A. (1974). An extension of the frank and Wolfe method of feasible directions. *Mathematical Programming*, **6**(1), 14–27.

[91] Holst, L. (1986). On birthday, collectors', occupancy and other classical urn problems. *International Statistical Review / Revue Internationale de Statistique*, **54**(1), 15–27.

[92] Hooke, R. and Jeeves, T. A. (1961). "Direct Search" solution of numerical and statistical problems. *Journal of the ACM*, **8**(2), 212–229.

[93] Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S. (2008). A Dual Coordinate Descent Method for Large-scale Linear SVM. In *the 25th International Conference on Machine Learning*, pages 408–415, New York, USA. ACM Press.

[94] Hu, C., Pan, W., and Kwok, J. T. (2009). Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pages 781–789.

[95] Huggins, J., Campbell, T., and Broderick, T. (2016). Coresets for scalable bayesian logistic regression. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*.

[96] Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *ICML*, pages 427–435.

[97] Jaggi, M. (2014). An Equivalence between the Lasso and Support Vector Machines. In *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 1–26. Chapman and Hall/CRC.

[98] Johnson, R. and Zhang, T. (2013a). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323.

[99] Johnson, R. and Zhang, T. (2013b). Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*.

[100] Johnson, T. and Guestrin, C. (2015). Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization. In *ICML 2015 - Proceedings of the 32th International Conference on Machine Learning*, pages 1171–1179.

[101] Johnson, T. B. and Guestrin, C. (2016). Unified methods for exploiting piecewise linear structure in convex optimization. In *Advances In Neural Information Processing Systems*, pages 4754–4762.

[102] Joulani, P., György, A., and Szepesvári, C. (2017). A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds. In *International Conference on Algorithmic Learning Theory, ALT*, pages 681–720.

[103] Joulani, P., György, A., and Szepesvári, C. (2020). A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, variance reduction, and variational bounds. *Theoretical Computer Science*, **808**, 108 – 138. Special Issue on Algorithmic Learning Theory.

[104] Joulani*, P., **Anant Raj***, György, A., and Szepesvari, C. (2020). A simpler approach to accelerated stochastic optimization: Iterative averaging meets optimism. In *ICML 2020- Proceedings of the 37th International Conference on Machine Learning*, PMLR.

[105] Kakade, S., Shalev-Shwartz, S., and Tewari, A. (2009a). On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, **2**(1).

[106] Kakade, S. M., Shalev-Shwartz, S., and Tewari, A. (2009b). On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. Technical report, Toyota Technological Institute - Chicago, USA.

[107] Källberg, L. and Larsson, T. (2014). Improved Pruning of Large Data Sets for the Minimum Enclosing Ball Problem. *Graphical Models*.

[108] Karimireddy, S. P., Koloskova, A., Stich, S. U., and Jaggi, M. (2019). Efficient greedy coordinate descent for composite problems. *AISTATS 2019, and arXiv:1810.06999*.

[109] Karmanov, V. G. (1974a). Convergence estimates for iterative minimization methods. *USSR Computational Mathematics and Mathematical Physics*, **14**(1), 1–13.

[110] Karmanov, V. G. (1974b). On convergence of a random search method in convex minimization problems. *Theory of Probability and its applications*, **19**(4), 788–794. (in Russian).

[111] Kavis, A., Levy, K. Y., Bach, F., and Cevher, V. (2019). Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In *Advances in Neural Information Processing Systems*, pages 6257–6266.

[112] Kerdreux, T., Pedregosa, F., and d'Aspremont, A. (2018). Frank-wolfe with subsampling oracle. In *ICML 2018 - Proceedings of the 35th International Conference on Machine Learning*.

[113] Kim, H. and Park, H. (2008). Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications*, **30**(2), 713–730.

[114] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[115] Kivinen, J. (2003). Online learning of linear classifiers. In *Advanced lectures on machine learning*, pages 235–257. Springer.

[116] Komiya, H. (1988). Elementary proof for sion's minimax theorem. *Kodai Math. J.*, **11**(1), 5–7.

[117] Kubo, M., Banno, R., Manabe, H., and Minoji, M. (2019). Implicit regularization in over-parameterized neural networks. *arXiv preprint arXiv:1903.01997*.

[118] Kundu, A., Bach, F., and Bhattacharya, C. (2018). Convex optimization over intersection of simple sets: improved convergence rate guarantees via an exact penalty approach. In *International Conference on Artificial Intelligence and Statistics*, pages 958–967.

[119] Lacoste-Julien, S. and Jaggi, M. (2015). On the Global Linear Convergence of Frank-Wolfe Optimization Variants. In *NIPS 2015*, pages 496–504.

[120] Lacoste-Julien, S., Schmidt, M., and Bach, F. (2012a). A simpler approach to obtaining an O(1/t) convergence rate for projected stochastic subgradient descent. *arXiv.org*.

[121] Lacoste-Julien, S., Schmidt, M., and Bach, F. (2012b). A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*.

[122] Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. (2013). Block-Coordinate Frank-Wolfe Optimization for Structural SVMs. In *ICML 2013 - Proceedings of the 30th International Conference on Machine Learning*.

[123] Lan, G. (2012). An optimal method for stochastic composite optimization. *Mathematical Programming*, **133**(1-2), 365–397.

[124] Lan, G. and Zhou, Y. (2018). An optimal randomized incremental gradient method. *Mathematical programming*, **171**(1-2), 167–215.

[125] Lan, G., Li, Z., and Zhou, Y. (2019). A unified variance-reduced accelerated gradient method for convex optimization. In *Advances in Neural Information Processing Systems*, pages 10462–10472.

[126] Langberg, M. and Schulman, L. J. (2010). Universal ε-approximators for integrals. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 598–607. SIAM.

[127] Le Roux, N., Schmidt, M., and Bach, F. (2012). A Stochastic Gradient Method with an Exponential Convergence Rate for Strongly-Convex Optimization with Finite Training Sets. *arXiv.org*.

[128] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., *et al.* (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.

[129] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. **401**(6755), 788–791.

[130] Lee, S. and Xing, E. P. (2014). Screening Rules for Overlapping Group Lasso. *arXiv*.

[131] Lee, Y. T. and Sidford, A. (2013). Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *FOCS '13 - Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 147–156.

[132] Lei, L. and Jordan, M. (2017). Less than a single pass: Stochastically controlled stochastic gradient. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 148–156. PMLR.

[133] Lei, L., Ju, C., Chen, J., and Jordan, M. I. (2017). Non-convex finite-sum optimization via SCSG methods. In *Advances in Neural Information Processing Systems*, pages 2345–2355.

[134] Levy, K. Y., Yurtsever, A., and Cevher, V. (2018). Online adaptive methods, universality and acceleration. In *Advances in Neural Information Processing Systems*, pages 6500–6509.

[135] Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, **5**, 361–397.

[136] Liang, T. and Rakhlin, A. (2018). Just interpolate: Kernel "ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*.

[137] Lin, H., Mairal, J., and Harchaoui, Z. (2015a). A Universal Catalyst for First-Order Optimization. *arXiv.org*.

[138] Lin, H., Mairal, J., and Harchaoui, Z. (2015b). A universal catalyst for first-order optimization. In *Advances in neural information processing systems*, pages 3384–3392.

[139] Lin, Q., Lu, Z., and Xiao, L. (2015c). An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, **25**(4), 2244–2273.

[140] Liu, C. and Belkin, M. (2018). Accelerating SGD with momentum for over-parameterized learning. *arXiv preprint arXiv:1810.13395*.

[141] Liu, J., Zhao, Z., Wang, J., and Ye, J. (2014). Safe Screening with Variational Inequalities and Its Application to Lasso. In *ICML 2014 - Proceedings of the 31st International Conference on Machine Learning*, pages 289–297.

[142] Locatello, F., Tschannen, M., Rätsch, G., and Jaggi, M. (2017a). Greedy algorithms for cone constrained optimization with convergence guarantees. In *NIPS - Advances in Neural Information Processing Systems 30*.

[143] Locatello, F., Khanna, R., Tschannen, M., and Jaggi, M. (2017b). A unified optimization view on generalized matching pursuit and frank-wolfe. In *AISTATS - Proc. International Conference on Artificial Intelligence and Statistics*.

[144] Locatello*, F., **Anant Raj***, Praneeth Karimireddy, S., Rätsch, G., Schölkopf, B., Stich, S., and Jaggi, M. (2018). On matching pursuit and coordinate descent. In *35th International Conference on Machine Learning (ICML)*, pages 3204–3213. PMLR.

[145] Lu, H., Freund, R. M., and Mirrokni, V. (2018). Accelerating greedy coordinate descent methods. In *ICML 2018 - Proceedings of the 35th International Conference on Machine Learning*.

[146] Lucic, M., Bachem, O., and Krause, A. (2016). Strong coresets for hard and soft Bregman clustering with applications to exponential family mixtures. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[147] Ma, S., Bassily, R., and Belkin, M. (2017). The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. *arXiv preprint arXiv:1712.06559*.

[148] Mahoney, M. W. (2011). Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, **3**(2), 123–224.

[149] Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, **41**(12), 3397–3415.

[150] Matoušek, J. and Gärtner, B. (2007). *Understanding and using linear programming*. Springer.

[151] Matoušek, J. (2008). On variants of the johnson–lindenstrauss lemma. *Random Structures & Algorithms*, **33**(2), 142–156.

[152] McMahan, H. B. (2011). Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 525–533.

[153] McMahan, H. B. (2017). A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, **18**(90), 1–50.

[154] Meir, R. and Rätsch, G. (2003). An introduction to boosting and leveraging. In *Advanced lectures on machine learning*, pages 118–183. Springer.

[155] Menard, S. (2018). *Applied logistic regression analysis*, volume 106. SAGE publications.

[156] Minsky, M. and Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry*. MIT press.

[157] Mohri, M. and Yang, S. (2016). Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics*, pages 848–856.

[158] Moulines, E. and Bach, F. R. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459.

[159] Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. P. (2018). On coresets for logistic regression. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*.

[160] Mutseniyeks, V. A. and Rastrigin, L. A. (1964). Extremal control of continuous multi-parameter systems by the method of random search. *Eng. Cybernetics*, **1**, 82–90.

[161] Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J. (2015). GAP Safe screening rules for sparse multi-task and multi-class models. In *NIPS 2015 - Advances in Neural Information Processing Systems 28*, pages 811–819.

[162] Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J. (2016). GAP Safe Screening Rules for Sparse-Group-Lasso. *arXiv*.

[163] Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J. (2017). Gap Safe screening rules for sparsity enforcing penalties. *JMLR*.

[164] Needell, D., Ward, R., and Srebro, N. (2014). Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz algorithm. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*, pages 1017–1025.

[165] Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009a). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, **19**(4), 1574–1609.

[166] Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009b). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, **19**(4), 1574–1609.

[167] Nemirovsky, A. and Yudin, D. (1983). *Problem complexity and method efficiency in optimization.* Wiley, Chichester, New York.

[168] Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate $O(1/K^2)$. *Soviet Mathematics Doklady*, **27**, 372–376.

[169] Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Springer US, Boston, MA.

[170] Nesterov, Y. (2012a). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, **22**(2), 341–362.

[171] Nesterov, Y. (2012b). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, **22**(2), 341–362.

[172] Nesterov, Y. (2012c). Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, **22**(2), 341–362.

[173] Nesterov, Y. (2013). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

[174] Nesterov, Y. (2015). Universal gradient methods for convex optimization problems. *Mathematical Programming*, **152**(1-2), 381–404.

[175] Nesterov, Y. (2018). *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing.

[176] Nesterov, Y. and Stich, S. U. (2017a). Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, **27**(1), 110–123.

[177] Nesterov, Y. and Stich, S. U. (2017b). Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, **27**(1), 110–123.

[178] Nguyen, H. and Petrova, G. (2014). Greedy strategies for convex optimization. *Calcolo*, pages 1–18.

[179] Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. (2017). SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621.

[180] Nutini, J., Schmidt, M. W., Laradji, I. H., Friedlander, M. P., and Koepke, H. A. (2015a). Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *ICML*, pages 1632–1641.

[181] Nutini, J., Schmidt, M., Laradji, I., Friedlander, M., and Koepke, H. (2015b). Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *ICML 2015 - Proceedings of the 32th International Conference on Machine Learning*, pages 1632–1641.

[182] Ogawa, K., Suzuki, Y., and Takeuchi, I. (2013). Safe Screening of Non-Support Vectors in Pathwise SVM Computation. In *ICML*, pages 1382–1390.

[183] Ogawa, K., Suzuki, Y., Suzumura, S., and Takeuchi, I. (2014). Safe Sample Screening for Support Vector Machines. *arXiv.org*.

[184] Osokin, A., Alayrac, J.-B., Lukasewitz, I., Dokania, P. K., and Lacoste-Julien, S. (2016). Minding the gaps for block frank-wolfe optimization of structured svms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 593–602. JMLR.org.

[185] Papa, G., Bianchi, P., and Clémençon, S. (2015). Adaptive Sampling for Incremental Optimization Using Stochastic Gradient Descent. *ALT 2015 - 26th International Conference on Algorithmic Learning Theory*, pages 317–331.

[186] Paquette, C., Lin, H., Drusvyatskiy, D., Mairal, J., and Harchaoui, Z. (2018). Catalyst for gradient-based nonconvex optimization. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 613–622. PMLR.

[187] Parikh, N., Boyd, S., *et al.* (2014). Proximal algorithms. *Foundations and Trends® in Optimization*, **1**(3), 127–239.

[188] Pena, J. and Rodriguez, D. (2015). Polytope conditioning and linear convergence of the frank-wolfe algorithm. *arXiv preprint arXiv:1512.06142*.

[189] Perekrestenko, D., Cevher, V., and Jaggi, M. (2017). Faster Coordinate Descent via Adaptive Importance Sampling. In A. Singh and J. Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 869–877, Fort Lauderdale, FL, USA. PMLR.

[190] Pilanci, M. and Ergen, T. (2020). Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. *arXiv preprint arXiv:2002.10553*.

[191] Polyak, B. T. (1990). New stochastic approximation type procedures. *Automat. i Telemekh*, **7**(98-107), 2.

[192] Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Fournal on Control and Pptimization*, **30**(4), 838–855.

[193] Qu, Z. and Richtárik, P. (2016). Coordinate descent with arbitrary sampling i: algorithms and complexity. *Optimization Methods and Software*, **31**(5), 829–857.

[194] Qu, Z., Richtárik, P., and Zhang, T. (2014). Randomized Dual Coordinate Ascent with Arbitrary Sampling. *arXiv.org*.

[195] Rakhlin, A. and Sridharan, K. (2013a). Online learning with predictable sequences. In *Conference on Learning Theory*, pages 993–1019.

[196] Rakhlin, S. and Sridharan, K. (2013b). Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074.

[197] Rätsch, G., Mika, S., Warmuth, M. K., *et al.* (2001). On the convergence of leveraging. In *NIPS*, pages 487–494.

[198] Reddi, S. J. (2017). *New Optimization Methods for Modern Machine Learning*. Ph.D. thesis, Stanford University.

[199] Reddi, S. J., Hefny, A., Sra, S., Póczos, B., and Smola, A. J. (2015). On variance reduction in stochastic gradient descent and its asynchronous variants. In *Advances in Neural Information Processing Systems*, pages 2647–2655.

[200] Reddi, S. J., Sra, S., Póczos, B., and Smola, A. (2016a). Fast incremental method for smooth nonconvex optimization. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1971–1977. IEEE.

[201] Reddi, S. J., Hefny, A., Sra, S., Póczos, B., and Smola, A. (2016b). Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pages 314–323.

[202] Richtárik, P. and Takáč, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, **144**(1-2), 1–38.

[203] Richtárik, P. and Takáč, M. (2016). On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, **10**(6), 1233–1243.

[204] Richtárik, P. and Takáč, M. (2016). Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, **156**(1), 433–484.

[205] Robbins, H. and Monro, S. (1951a). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, **22**(3), 400–407.

[206] Robbins, H. and Monro, S. (1951b). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

[207] Rudi, A., Camoriano, R., and Rosasco, L. (2015). Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665.

[208] Rudi, A., Calandriello, D., Carratino, L., and Rosasco, L. (2018). On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems*, pages 5672–5682.

[209] Schmidt, M., Babanezhad, R., Ahmed, M., Defazio, A., Clifton, A., and Sarkar, A. (2015). Non-Uniform Stochastic Average Gradient Method for Training Conditional Random Fields. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 819–828, San Diego, California, USA. PMLR.

[210] Schmidt, M., Le Roux, N., and Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, **162**(1-2), 83–112.

[211] Schölkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

[212] Seber, G. A. and Lee, A. J. (2012). *Linear regression analysis*, volume 329. John Wiley & Sons.

[213] Shalev-Shwartz, S. (2016). SDCA without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754.

[214] Shalev-Shwartz, S. and Singer, Y. (2005). A new perspective on an old perceptron algorithm. In *International Conference on Computational Learning Theory*, pages 264–278. Springer.

[215] Shalev-Shwartz, S. and Tewari, A. (2011). Stochastic Methods for l_1-regularized Loss Minimization. *JMLR*, **12**, 1865–1892.

[216] Shalev-Shwartz, S. and Zhang, T. (2013a). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, **14**(Feb), 567–599.

[217] Shalev-Shwartz, S. and Zhang, T. (2013b). Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *JMLR*, **14**, 567–599.

[218] Shalev-Shwartz, S. and Zhang, T. (2014). Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International conference on machine learning*, pages 64–72.

[219] Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2010). Pegasos: Primal Estimated Sub-Gradient Solver for SVM. *Mathematical Programming*, **127**(1), 3–30.

[220] Shibagaki, A., Karasuyama, M., Hatano, K., and Takeuchi, I. (2016). Simultaneous Safe Screening of Features and Samples in Doubly Sparse Modeling. In *ICML 2016 - Proceedings of the 33th International Conference on Machine Learning*, pages 1577–1586.

[221] Shrivastava, A. and Li, P. (2014). Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 2321–2329, Cambridge, MA, USA. MIT Press.

[222] Sion, M. (1958). On general minimax theorems. *Pacific Journal of Mathematics*, **8**(1), 171–176.

[223] Smith, V., Forte, S., Jaggi, M., and Jordan, M. I. (2015). L_1-Regularized Distributed Optimization: A Communication-Efficient Primal-Dual Framework. *arXiv cs.LG*.

[224] Song, C., Cui, S., Jiang, Y., and Xia, S.-T. (2017). Accelerated stochastic greedy coordinate descent by soft thresholding projection onto simplex. In *NIPS - Advances in Neural Information Processing Systems*, pages 4841–4850.

[225] Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, **19**(1), 2822–2878.

[226] Spielman, D. A. and Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing*, **40**(6), 1913–1926.

[227] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, **15**(1), 1929–1958.

[228] Stich, S. U. (2014). *Convex Optimization with Random Pursuit*. Ph.D. thesis, ETH Zurich. Nr. 22111.

[229] Stich, S. U., Müller, C. L., and Gärtner, B. (2013). Optimization of convex functions with random pursuit. *SIAM Journal on Optimization*, **23**(2), 1284–1309.

[230] Stich, S. U., Müller, C. L., and Gärtner, B. (2016). Variable metric random pursuit. *Mathematical Programming*, **156**(1), 549–579.

[231] Stich, S. U., **Anant Raj**, and Jaggi, M. (2017a). Approximate steepest coordinate descent. In *ICML 2017 - Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *PMLR*, pages 3251–3259.

[232] Stich, S. U., Raj, A., and Jaggi, M. (2017b). Approximate steepest coordinate descent. In *Proceedings of the 34rd International Conference on Machine Learning*.

[233] Stich, S. U., **Anant Raj**, and Jaggi, M. (2017c). Safe adaptive importance sampling. In *Advances in Neural Information Processing Systems*, pages 4381–4391.

[234] Strohmer, T. and Vershynin, R. (2008). A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, **15**(2), 262.

[235] Temlyakov, V. (2013). Chebushev Greedy Algorithm in convex optimization. *arXiv.org*.

[236] Temlyakov, V. (2014). Greedy algorithms in convex optimization on Banach spaces. In *48th Asilomar Conference on Signals, Systems and Computers*, pages 1331–1335. IEEE.

[237] **Anant Raj** and Bach, F. (2020). Explicit regularization of stochastic gradient methods through duality. *arXiv preprint arXiv:2003.13807 (Submitted to AISTATS 2021)*.

[238] **Anant Raj** and Stich, S. U. (2018). k-svrg: Variance reduction for large scale optimization. *arXiv preprint arXiv:1805.00982 (Manuscript)*.

[239] **Anant Raj**, Olbrich, J., Gärtner, B., Schölkopf, B., and Jaggi, M. (2016). Screening rules for convex problems. *Optimization for Machine Learning Workshop (OPT 2016), arXiv preprint arXiv:1609.07478*.

[240] **Anant Raj**, Musco, C., and Mackey, L. (2020). Importance sampling via local sensitivity. In *International Conference on Artificial Intelligence and Statistics*, pages 3099–3109. PMLR.

[241] Tibshirani, R. J. (2015). A general framework for fast stagewise algorithms. *Journal of Machine Learning Research*, **16**, 2543–2588.

[242] Tibshirani, R. J. (2017). Dykstra's algorithm, admm, and coordinate descent: Connections, insights, and extensions. In *Advances in Neural Information Processing Systems*, pages 517–528.

[243] Tichatschke, R. (2011). Proximal point methods for variational problems.

[244] Torczon, V. (1997). On the convergence of pattern search algorithms. *SIAM Journal on optimization*, **7**(1), 1–25.

[245] Tropp, J. A. (2004). Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, **50**(10), 2231–2242.

[246] Tsampouka, P. and Shawe-Taylor, J. (2005). Analysis of generic perceptron-like large margin classifiers. In *European Conference on Machine Learning*, pages 750–758. Springer.

[247] Tsang, I. W., Kwok, J. T., and Cheung, P.-M. (2005). Core Vector Machines: Fast SVM Training on Very Large Data Sets. *Journal of Machine Learning Research*, **6**, 363–392.

[248] Tseng, P. (1993). Dual coordinate ascent methods for non-strictly convex minimization. *Mathematical programming*, **59**(1-3), 231–247.

[249] Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, **109**(3), 475–494.

[250] Tseng, P. (2008). On accelerated proximal gradient methods for convex-concave optimization. Technical report.

[251] Tseng, P. and Bertsekas, D. P. (1987). Relaxation methods for problems with strictly convex separable costs and linear constraints. *Mathematical Programming*, **38**(3), 303–321.

[252] Tseng, P. and Yun, S. (2009). A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, **117**(1), 387–423.

[253] Vaswani, S., Bach, F., and Schmidt, M. (2018). Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. *arXiv preprint arXiv:1810.07288*.

[254] Vaswani, S., Mishkin, A., Laradji, I., Schmidt, M., Gidel, G., and Lacoste-Julien, S. (2019). Painless stochastic gradient: Interpolation, line-search, and convergence rates. *arXiv preprint arXiv:1905.09997*.

[255] Von Neumman, J. (1951). Functional operators ii, the geometry of orthogonal spaces. *Annals of Math. studies*, **22**.

[256] Wang, J., Lin, B., Gong, P., Wonka, P., and Ye, J. (2013). Lasso Screening Rules via Dual Polytope Projection. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*.

[257] Wang, J., Zhou, J., Liu, J., Wonka, P., and Ye, J. (2014). A Safe Screening Rule for Sparse Logistic Regression. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*, pages 1053–1061.

[258] Wang, J.-K. and Abernethy, J. D. (2018). Acceleration through optimistic no-regret dynamics. In *Advances in Neural Information Processing Systems*, pages 3824–3834.

[259] Ward, R., Wu, X., and Bottou, L. (2019). Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686.

[260] Weese, J. (1993). A regularization method for nonlinear ill-posed problems. *Computer Physics Communications*, **77**(3), 429–440.

[261] Wen, Z., Yin, W., Zhang, H., and Goldfarb, D. (2012). On the convergence of an active-set method for ℓ_1 minimization. *Optimization Methods and Software*, **27**(6), 1127–1146.

[262] Wright, S. J. (2015). Coordinate descent algorithms. *Mathematical Programming*, **151**(1), 3–34.

[263] Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, **2**(1), 224–244.

[264] Xiang, Z. J., Wang, Y., and Ramadge, P. J. (2014). Screening Tests for Lasso Problems. *arXiv.org*.

[265] Xiao, L. (2009). Dual averaging method for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems*, pages 2116–2124.

[266] Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, **11**(Oct), 2543–2596.

[267] Xu, P., Yang, J., Roosta-Khorasani, F., Ré, C., and Mahoney, M. W. (2016). Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*.

[268] Yao, Y., Rosasco, L., and Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, **26**(2), 289–315.

[269] Ye, H., Luo, L., and Zhang, Z. (2017). Approximate Newton methods and their local convergence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

[270] Zhang, J., Rivard, B., and Rogge, D. (2008). The successive projection algorithm (SPA), an algorithm with a spatial constraint for the automatic search of endmembers in hyperspectral data. *Sensors*, **8**(2), 1321–1342.

[271] Zhao, P. and Zhang, T. (2014). Stochastic Optimization with Importance Sampling. *arXiv.org*.

[272] Zhao, P. and Zhang, T. (2015). Stochastic optimization with importance sampling for regularized loss minimization. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1–9, Lille, France. PMLR.

[273] Zhu, R. (2016). Gradient-based sampling: An adaptive importance sampling for least-squares. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 406–414. Curran Associates, Inc.

[274] Zieliński, R. and Neumann, P. (1983). *Stochastische Verfahren zur Suche nach dem Minimum einer Funktion*. Akademie-Verlag, Berlin, Germany.

[275] Zimmert, J., de Witt, C. S., Kerg, G., and Kloft, M. (2015). Safe screening for support vector machines. In *NIPS Workshop on Optimization for Machine Learning*, pages 1–5.

[276] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**(2), 301–320.