CAA
2018

Martina Trognitz
https://dx.doi.org/10.15496/publikation-87761

Evaluation of Tools for Clustering of Archaeological Data

Martina Trognitz
Austrian Academy of Sciences
martina.trognitz@oeaw.ac.at

Abstract

Dedicated clustering programs, preferably with graphical user interfaces, are a useful alternative for those not very well acquainted with programming languages. In the last decades a variety of ready-to-use tools were developed, of which a freely availlable selection was evaluated in regard to their suitability for archaeological data, the number of functions and ease of use. This task was done with a dataset describing Aegean seals.

Keywords: clustering, clustering programs, evaluation of tools

Introduction

A common task in archaeology consists in subdividing large sets of seemingly similar artefacts, such as pots, stone tools or coins, into smaller groups of objects with distinct features. By classifying objects into groups, a typology is created. This process can be automated by the means of cluster analysis, an unsupervised machine-learning technique. With the increasing availability of computers from the sixties onwards, clustering was also tested and applied in the archaeological field (Baxter 1994: 2; Hodson 1970: 299-300).

In the last decades a variety of ready-to-use tools and programs for cluster analysis were developed, of which a selection was evaluated in regard to their suitability for archaeological data. This task was done with a dataset describing Aegean seals.

First, an introduction to cluster analysis is given in the next section. It is then followed by a section with a more in detail description of the data set, after which, the evaluated tools are presented. A summary of the evaluation of clustering programs is given in Table 1.

Cluster Analysis

In statistics, clustering is part of multivariate analysis methods (Drennan: 309), whereas it is also described as a part of machine learning or data mining (Kumar, Steinbach & Tan 2006: 6-7). With cluster analysis, a set of objects can be grouped into distinct groups of similar entities solely based on the objects' descriptive variables. It is thus also referred to as an unsupervised machine learning method or, more specifically, an unsupervised classification method (Kumar, Steinbach & Tan 2006: 490-491).

A vast number of clustering algorithms for different needs, aims, and data sets are available to choose from. Popular clustering algorithms include k-means and hierarchical clustering, which do also represent the two distinct clustering types: partitional and hierarchical. K-means belongs to the former group and divides a data set into groups without overlaps. Hierarchical clustering produces nested clusters, by linking smaller groups into superordinate groups (Kumar, Steinbach & Tan 2006: 491-492).

Though a great variety of clustering algorithms exists, they all have in common that similarity (or

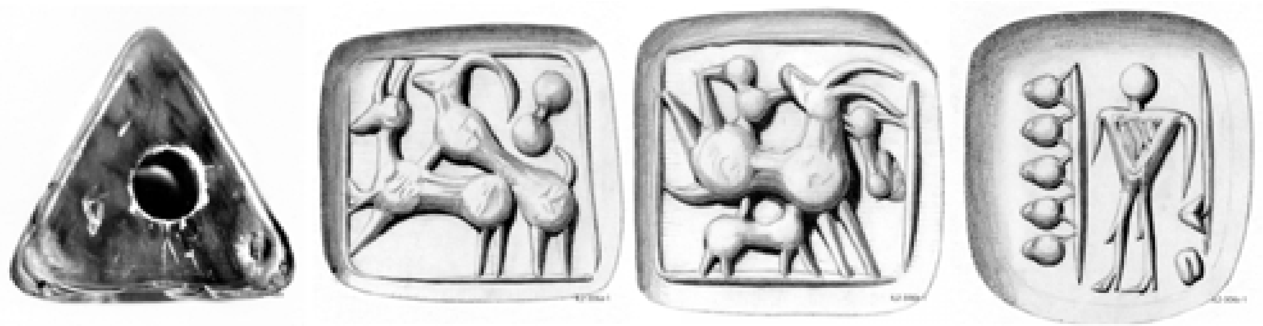


Figure 1. The three-sided prism CMS II,2 306 and drawings of its three sides.

dissimilarity) measures are used to calculate how similar (or different) two objects to be compared are (Kumar, Steinbach & Tan 2006: 65; Drennan 2009: 271). The higher a similarity coefficient is, the more alike two objects are.

Not all similarity measures are applicable to all data types. The Euclidean distance (Drennan 2009: 272-277), e.g., is not suitable for nominal values, whereas the Jaccard coefficient (Drennan 2009: 277-279) is. The Gower distance (Drennan 2009: 280) can be used for numeric, nominal, and binary (or dichotomous) values.

Data Set

The used data set describes 1033 Aegean seals with more than one face for sealing, i. e. multi-sided seals. The seals are small objects made of stone, bone or ivory and come in various shapes, such as lentoids, cylinders, cubes or triangular prisms. Most of the seals originate from Bronze Age Crete (Minoan seals) and mainland Greece (Mycenean seals). Their dating ranges from 3000 to 1100 BCE.

The information was harvested from the freely accessible database Arachne (http://arachne.uni-koeln.de) where all Aegean seals documented in the „Corpus der Minoischen und Mykenischen Siegel" (CMS) (http://cmsheidelberg.uni-hd.de/) are recorded. Each seal is described with about fifty attributes organised into eleven thematic groups: „identification", „provenience", „shape", „material & technique", „measurements & preservation", „general information about decoration", „stylistic classification", „ornaments", „characters", „figurative motifs excepting creatures" and „creatures".

The attributes contain numeric (interval and ratio), binary, ordinal, geographic and nominal values. With the help of the three-sided seal CMS II,2 306 in Figure 1, examples for each data type are given and illustrated in the following subsections.

Numeric Attributes

Numeric data types in the used data set belong to the thematic groups „measurements & preservation" and „creatures". They include the number of sides a seal has, dimensions (e. g. width or length) and a computed count of all creatures depicted on one seal. For the example in Figure 1, the values are 3, 1.1 cm and 1.2 cm, and 7, accordingly.

Binary Attributes

Originally the data set did not contain any binary attributes. During processing, a new attribute was introduced based on the attribute's values listing all used script characters. It denotes if script is present on a seal or not, thus only assuming either 0 (no script) or 1 (use of script). On CMS II,2 306 no script (=0) is used.

Ordinal Attributes

Ordinal attributes can be found in the thematic groups „provenience", „material & technique", and „stylistic classification". They are used to indicate a seal's dating using a relative chronology, such as MM II for CMS II,2 306, which stands for the second phase of the Middle Minoan period. Periods in the chronology of the Aegean Bronze Age are not of equal length, and absolute dates are still a matter of debate (Krzyszkowska 2005: 11).

A seal's material class in the original data set is given with nominal values, but its equivalents on the Moh's scale were introduced during processing. For the example seal, the value equals 2, which stands for soft stone.

Geographic Attributes

Geographic coordinates indicate where a seal was found and are given with longitude and latitude, e. g. 23.71622 and 37.97945.

Nominal Attributes

Nominal values are used for most of the attributes to describe a seal's shape, depicted creatures, objects or ornaments, and for listing the different engraved script characters. The shape of the example in Figure 1 is described with „Dreiseitiges Prisma“ (three-sided prism).

Evaluation of Clustering Tools

Although dedicated libraries for programming languages like Python or R exist, clustering programs, preferably with graphical user interfaces, are a convenient alternative for those not very well acquainted with programming languages.

Criteria

The selection presented here ranges from simple programs with single functionalities to full-blown statistic environments able to set up custom workflows. The most significant selection criterion was free availability, which is why any software not free of charge was excluded from the evaluation.

The evaluation focused on overall functionalities and handling of programs as opposed to performance and clustering results. The latter is a subject of its own and would require a data set with already known clusters. An example of how the evaluation of clustering algorithms might be executed is shown in the works of Hodson, Sneath & Doran (1966) and Hodson (1970), which also includes a small data set of Iron Age fibulae.

Tests were run on a computer with Xubuntu 16.4 and partially on a virtual machine with Windows XP. Because the k-means algorithm is supported by all programs, this was used to test importing, parameter setting, actual clustering, visualisation and exporting of results to get a feeling for the programs.

The considered programs, are presented in order of their latest version with the oldest first. A tabular summary of each program's features is given in Table I. Each row represents evaluation results for one program. The first column displays the latest version of the software and the year that it was released. In the next three columns the supported operating systems are listed.

Of all tested programs, only CLUTO did not have a graphical user interface (GUI) because the respective package could not be executed. The columns installation, technical, manual, and training are all concerned with documentation: Are installation guidelines available and useful? Is there extensive technical documentation? Does there exist a manual and is further training material provided?

How large and active the community using a specific tool is, is a good indicator of how easy it is to get help or alternative tutorials. This is indicated in a separate column.

The actual functionalities provided by each program are indicated with the import and export formats, the supported data types (numerical, ordinal or nominal), the number of algorithms and similarity measures available and visualisation as well as validation capabilities of the results.

Cluster 3.0

Cluster 3.0[1], a program originally written by Michael Eisen at Stanford University, is a cluster program for analysing genome datasets which uses the C Clustering Library 1.54.

It offers a GUI, but can also be used on the command line. Three clustering algorithms are provided, of which the first one, hierarchical clustering, offers four variants, thus effectively offering six algorithms. Also, principal component analysis (PCA) is available. Eight similarity measures are provided and described in detail in the manual (four variants of the Pearson correlation, Spearman rank, Kendall's τ, Euclidean distance, and the city-block distance).

Cluster 3.0 can only process numerical data and

[1] http://bonsai.hgc.jp/~mdehoon/software/cluster/software.htm; Manual is available at http://bonsai.hgc.jp/~mdehoon/software/cluster/cluster3.pdf

this has to be provided in a tab-delimited txt-file. After the import values can be further filtered and adjusted. ‚Genes' refers to the rows, while ‚Arrays' takes into account the data in a column.

Results are stored in different text-based output files. When using k-means for 'genes', a cdt and a kgg file are created. In the cdt file, the rows of the original file are rearranged according to the clusters the objects were assigned to. Some additional columns and rows are inserted, but the meaning of those is not described in the manual. The kgg file contains the object identifier and the cluster number it was assigned to.

The software lacks a general overview of cluster results, e.g. how many objects each cluster has. A visualisation of results is only available for results achieved with hierarchical clustering and with the TreeView software.[2]

CLUTO

CLUTO[3], the CLUstering TOolkit, was developed for the clustering of documents by George Karypis at the University of Minnesota. It can be used to cluster low and high-dimensional datasets.

It is command line based, but with gCluto (2003), which unfortunately did not work on Xubuntu 16.04, a GUI is offered as well. Furthermore, the program can be used as a stand-alone C or C++ library. CLUTO offers the vcluster and scluster programs to cluster data in k clusters. With vcluster each object is treated as a vector in high-dimensional space, while with scluster clustering is done by calculating the similarity space between the objects.

Of the six clustering algorithms provided, four are partitional and the remaining two are agglomerative (or hierarchical). The user has four similarity measures to choose from, as well as seven different criterion functions for finding the clusters.

Data has to be provided either as a MatrixFile for vcluster or as a GraphFile for scluster.

The manual provides information on how to format these files. Many examples for executing CLUTO are also provided in the manual. CLUTO can only process numerical data.

Results are directly displayed in the command line, and multiple options exist to fine-tune the output. The output also contains internal and external cluster quality statistics, which also serve as validation or evaluation of the clusters; the former is based on the clustering criterion function, and the latter is based on external measures provided by the user.

Two files representing the result are created as well. The actual clustering outcome is stored in the clustering solution file <orig-filename>.clustering.<k>, which contains n lines with a single number representing the cluster an object belongs to. The visualised tree is stored in <orig-lename>.cltree.<k> and contains 2k - 1 lines containing the parents of the nodes with further values describing similarities. Visualisations of the results can be exported in various graphic file formats.

ELKI

ELKI[4] - Environment for Developing KDD-Applications Supported by Index-Structures is a JAVA-based program developed at the Data Science Lab of LMU Munich with research on clustering in focus.

Data can be imported in its own ELKI format, as arff or as libSVM. Custom import settings allow accommodating other file formats. The GUI provides a tabular view to select and set the wished parameters for clustering. This creates a command, which can then be executed from within the interface. Upon execution, a visualisation window with an overview of graphs is displayed. The graphs can be explored individually. Exporting data is only possible into a set of txt files compatible with GNUPlot.

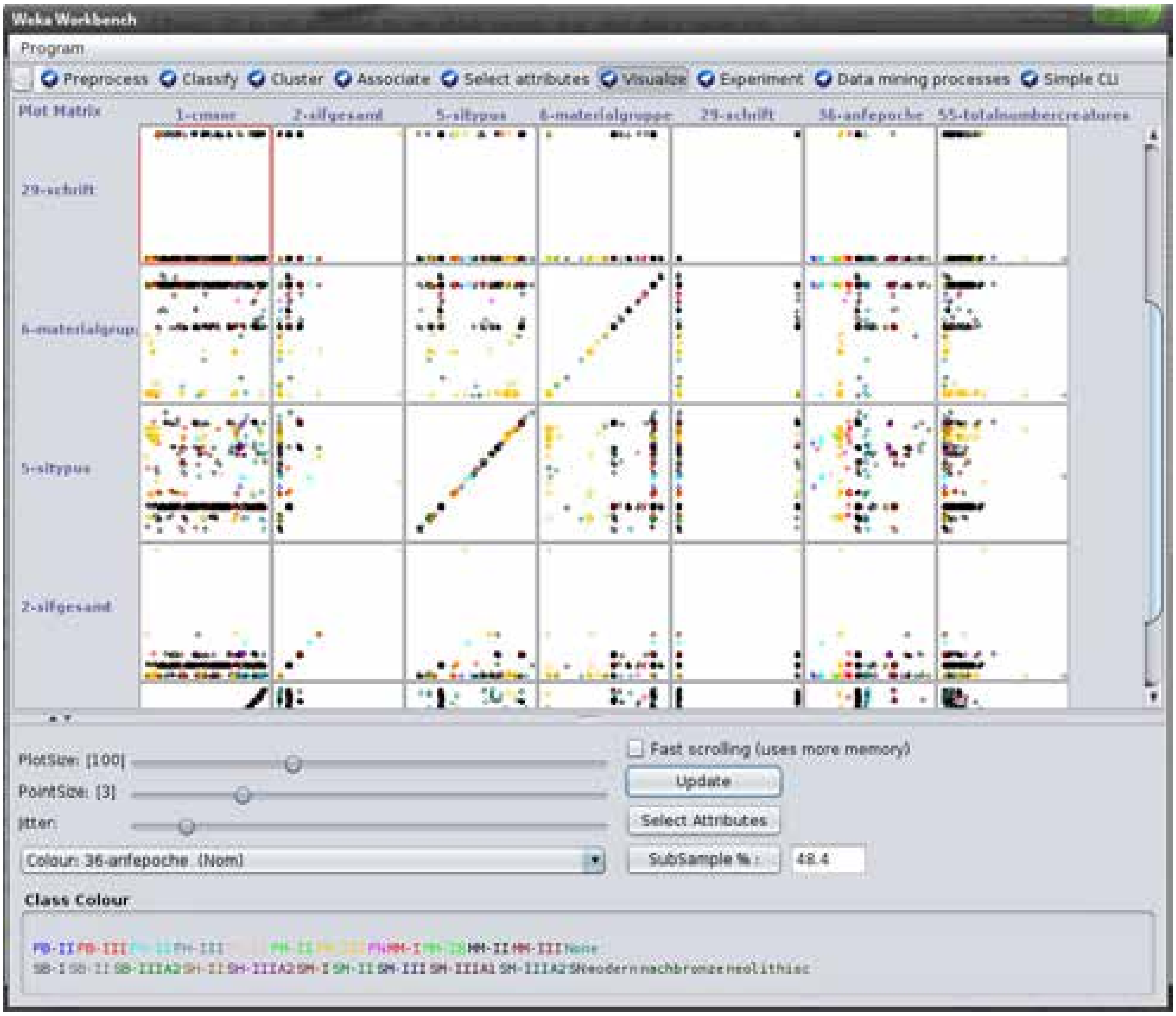
The extensive documentation offers a few tutorials to allow for a quick start. But for the proper use and full understanding, good theoretical knowledge about cluster analysis and its related concepts is required.

Algorithms and distance measures can be extended, which would also allow to process ordinal and nominal values.

[2] https://jtreeview.sourceforge.net/

[3] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

[4] https://elki-project.github.io/



☐ Fast scrolling (uses more memory)

PlotSize: [100]

PointSize: [3]

Jitter:

Colour: 36-anfepoche (Nom)

SubSample % : 48.4

Figure 2. EMultiple scatterplots for different attribute pairs in Weka.

jMinHep

jMinHep[5] provides three clustering algorithms and is a JAVA-based program developed by S. Chekanov. It is part of the DataMelt environment[6] but is also available as a standalone application.

Data has to be imported in a slightly modified arff file format. Arff files can be converted from a csv file with online tools[7] and for jMinHep the separating commas have to be replaced by spaces. Only numeric values can be processed and all other data types should be removed from the import file to avoid errors.

Results are visualised as a graph either representing single data points or density. These graphs can be exported as pdf. A more informative result with data points and the clusters assigned, can be opened by clicking on “show result” and saved as a txt file.

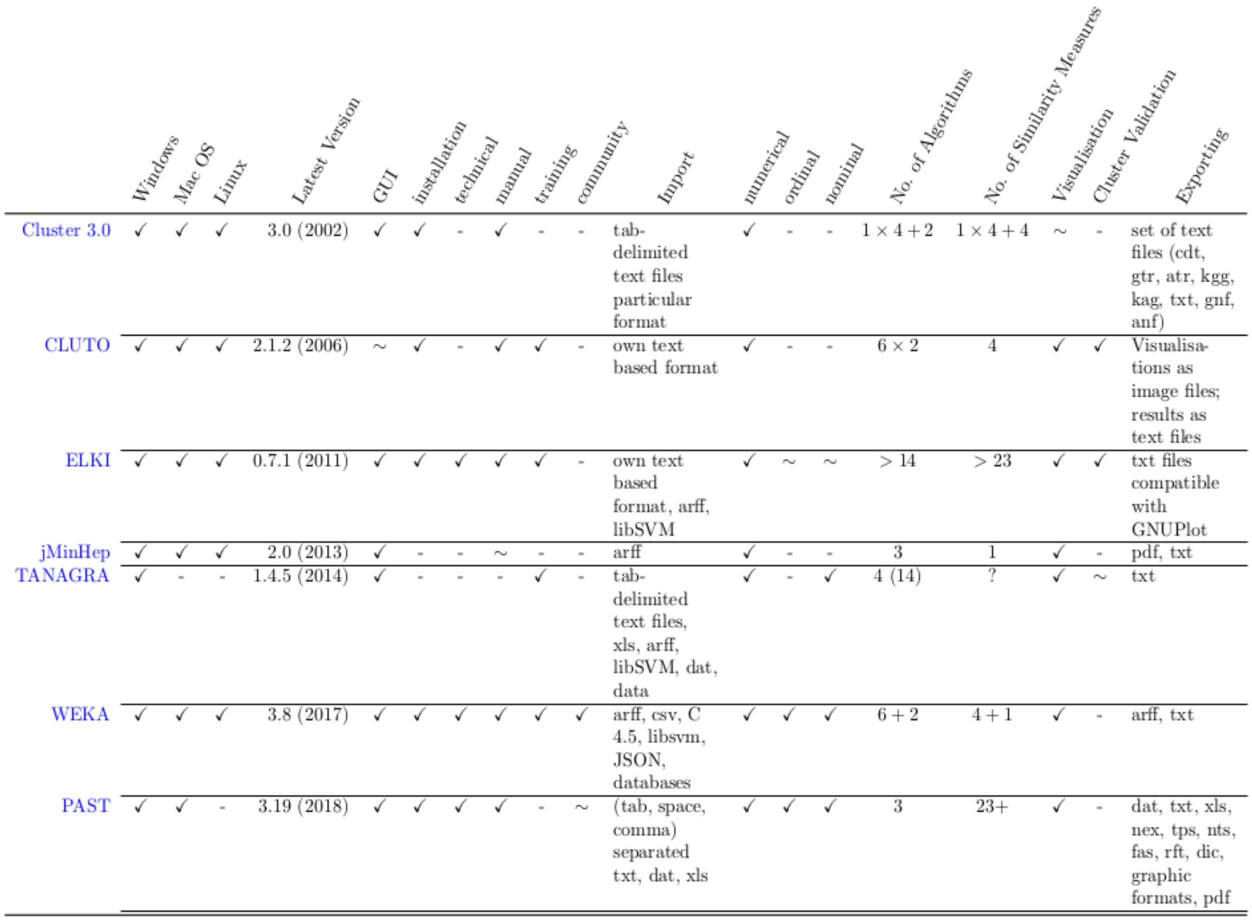
The application is not supported anymore and documentation about it on DataMelt has limited access.[8]

[5] http://jwork.org/jminhep/

[6] https://datamelt.org/

[7] E. g. https://ikuz.eu/csv2arff/

[8] https://handwiki.org/wiki/DMelt:AI/Data_Clustering



	Windows	Mac OS	Linux	Latest Version	GUI	installation	technical	manual	training	community	Import	numerical	ordinal	nominal	No. of Algorithms	No. of Similarity Measures	Visualisation	Cluster Validation	Exporting
Cluster 3.0	✓	✓	✓	3.0 (2002)	✓	✓	-	✓	-	-	tab-delimited text files particular format	✓	-	-	$1 \times 4 + 2$	$1 \times 4 + 4$	~	-	set of text files (cdt, gtr, atr, kgg, kag, txt, gnf, anf)
CLUTO	✓	✓	✓	2.1.2 (2006)	~	✓	-	✓	✓	-	own text based format	✓	-	-	6×2	4	✓	✓	Visualisations as image files; results as text files
ELKI	✓	✓	✓	0.7.1 (2011)	✓	✓	✓	✓	✓	-	own text based format, arff, libSVM	✓	~	~	> 14	> 23	✓	✓	txt files compatible with GNUPlot
jMinHep	✓	✓	✓	2.0 (2013)	✓	-	-	~	-	-	arff	✓	-	-	3	1	✓	-	pdf, txt
TANAGRA	✓	-	-	1.4.5 (2014)	✓	-	-	-	✓	-	tab-delimited text files, xls, arff, libSVM, dat, data	✓	-	✓	4 (14)	?	✓	~	txt
WEKA	✓	✓	✓	3.8 (2017)	✓	✓	✓	✓	✓	✓	arff, csv, C 4.5, libsvm, JSON, databases	✓	✓	✓	$6 + 2$	$4 + 1$	✓	-	arff, txt
PAST	✓	✓	-	3.19 (2018)	✓	✓	✓	✓	-	~	(tab, space, comma) separated txt, dat, xls	✓	✓	✓	3	23+	✓	-	dat, txt, xls, nex, tps, nts, fas, rft, dic, graphic formats, pdf

Table I: Summary of the evaluation of clustering programs (2018).

TANAGRA

Tanagra[9] was developed by Ricco Rakotomalala. It is a data mining software providing different data mining methods from exploratory data analysis, statistical learning, and machine learning.

The software runs on Windows (the author claims it can also be executed with WINE on Linux) and comes with a GUI on which the user can create a stream diagram with several components for different tasks. Numeric and nominal data can be imported from tab-separated txt files or xls, arff, libSVM, dat or data files.

Testing clustering with only numerical values was successful, and although in theory clustering of nominal values should be possible it could not be achieved in practice. For clustering 14 options, which presumably represent variants of a total of four clustering algorithms are available. No similarity measure can be chosen.

Results can only be exported as a txt file and be visualised in the program. Visualisations cannot be exported and the lack of systematic documentation does not allow to fully understand what the program is capable of.

WEKA

Weka[10], the Waikato Environment for Knowledge Analysis, was developed at the University of Waikato as a workbench for machine learning.

It is a Java-based software application which offers full functionality via the command line interface. A large set of functions is also available via a GUI. The GUI offers five environments for creating stream diagrams for custom data workflows, including a command-line interface.

Testing was done by using the explorer environ-

[9] https://tanagra-machine-learning.blogspot.com/ and http://data-mining-tutorials.blogspot.co.at/search/label/Tanagra and http://tutoriels-data-mining.blogspot.com/

[10] https://www.cs.waikato.ac.nz/ml/weka/index.html

ment, but available algorithms and functionalities also apply to the remaining environments. Data can be imported in arff, csv, C4.5, libSVM and JSON format. Further editing, adjustment, filtering and transforming of data are supported. Weka can process numeric, nominal, ordinal, binary, dates and string data types. It is also possible to work with data from a remote database.

Six different algorithms (and two additional variants) and four (plus a user-defined filter) similarity measures are available for hierarchical and k-means clustering.

The general output of the clustering process is displayed in the GUI, which also provides a list of scatterplots for a quick overview, as shown in Figure 2. Furthermore, concise information about the functions is presented to the user when hovering over the respective buttons. Data with cluster assignments, can be exported in arff format. Visualisations cannot be exported.

PAST

PAST (PAleontological STatistics)[11] is a statistics software package developed by Øyvind Hammer at the Natural History Museum and the University of Oslo. The software offers a GUI literature references for the available methods and measures. Tabular data can be imported as a tab-, space- or comma-separated txt file, as dat, or as xls. Further operations and transformations are available, although a function is missing to convert nominal values in string format into a numeric format. For each column, a data type (unspecified, ordinal, nominal, binary, string, and group) can be selected. This step has to be done to enable the clustering of mixed data types.

Three clustering algorithms can be selected, and except for k-means a selection from 23 similarity measures is possible. The user can also provide a custom similarity measure or select different measures for mixed data types.

Results are displayed in a separate view which allows exporting in the nexus format. Visualisations of the resulting trees can be exported in various graphic formats. Exporting results from k-means is limited, and no cluster validation is provided.

Not Tested

There exist more tools, which are worth a look. For example, an open-source alternative of SPSS capable of doing k-means clustering called PSPP[12] exists. A similar system is SOFA - Statistics Open For All[13] or KNIME[14].

Some GUI programs are based on Python or R, such as Orange[15] or Rattle[16], which were specifically made for data mining.

And finally, an online tool should also be mentioned: clustVis[17]. It cannot cluster data, but it can be used for principal component analysis.

Conclusion

Though an archaeological dataset typically consists of numerical, nominal and spatial data, most clustering applications require the data to be described with numerical values.

From the seven evaluated programs Weka and especially PAST offer all necessary functionalities for processing archaeological data out of the box and easy understandable interfaces. For the more advanced users ELKI also poses a useful application, due to the possibility to use custom algorithms and similarity measures.

If one wants to have full control, more options, and complete flexibility using respective packages in Python or R might be the better way to go.

The article was finalised in 2018. For the publication in the proceedings the links were checked and updated. As of March 2023, new releases for ELKI, Weka, and PAST were available.

[11] https://www.nhm.uio.no/english/research/resources/past/

[12] https://www.gnu.org/software/pspp/

[13] http://www.sofastatistics.com/home.php

[14] https://www.knime.com/

[15] https://orangedatamining.com/

[16] https://rattle.togaware.com/

[17] https://biit.cs.ut.ee/clustvis/

References

Baxter, M J 1994 *Exploratory Multivariate Analysis in Archaeology.* Edinburgh: University Press.

Drennan, R D 2009 *Statistics for Archaeologists: A Common Sense Approach*. Boston, MA: Springer. DOI: 10.1007/978-1-4419-0413-3

Hodson, F R 1970 Cluster Analysis and Archaeology: Some New Developments and Applications. *World Archaeology*, 1(3): 299-320. Available at https://www.jstor.org/stable/124057 [Last accessed 23 March 2023].

Hodson, F R, Sneath, P H A and Doran, J E 1966 Some experiments in the numerical analysis of archaeological data. *Biometrika*, 53(3-4): 311-324. DOI: 10.1093/biomet/53.3-4.311

Krzyszkowska, O 2005 *Aegean Seals: An Introduction.* London: Inst. of Classical Studies, School of Advanced Study, Univ. of London.

Kumar, V, Steinbach, M and Tan, P-N 2006 *Introduction to Data Mining.* Boston: Pearson Addison Wesley.

Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, Blondel, M, Prettenhofer, P, Weiss, R, Dubourg, V, Vanderplas, J, Passos, A, Cournapeau, D, Brucher, M, Perrot, M and Duchesnay, E 2011 Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* 12: 2825-2830. Available at https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf [Last accessed 23 March 2023]