CUBO A Mathematical Journal
Vol.11, N$^{\underline{o}}$02, (139–149). May 2009

A General Purpose Platform for Data Clustering Analysis

Haiyan Qiao and Brandon Edwards
Department of Computer Science and Engineering
California State University, San Bernardino
San Bernardino, CA 92407, USA
email: `hqiao@csusb.edu`

ABSTRACT

Grouping objects into meaningful sets - clustering - is an important procedure in many fields of social sciences. Yet, clustering analysis is a difficult problem because many factors come into play in devising a well tuned clustering technique for a given clustering problem. Therefore, an easy-to-use clustering analysis tool is needed. In this paper, we have designed a general purpose clustering analysis platform, which integrates the different clustering algorithms and provides the clustering results in both textual and visual display. This platform will assist the users without any computing or programming background in the selection of appropriate clustering algorithm, improve the quality of clustering, and be extendable to the evaluation of clustering results.

RESUMEN

Agrupación de objetos en conjuntos meaningful-clustering es un importante procedimiento en muchos campos de las ciencias sociales. Análisis de clustering es un problema difícil pues varios factores entran en juego en idear una bien afinada técnica de clustering para un determinado problema de clustering. Por lo tanto, una análisis de clustering de fácil nanejo es necesário. En este artículo, designamos una plataforma para propositos generales de análisis de clustering, la cual integra los diferentes algoritmos de clustering y prove los resultados de clustering mostrando a la vez textual y visual. Esta plataforma asiste a los usuarios sin cualquer formación computacional o de programación en la seleción de algoritmos de clustering apropriados, perfeccionando la calidad del clustering y extendible para evaluación de resultados de clustering.

Key words and phrases: *Data clustering, clustering validation, clustering platform.*

Math. Subj. Class.: *62H30, 91C20.*

1 Introduction

With the increased advancement of computers and technologies for data collection and data storage, the task of finding patterns and discovering knowledge through data analysis techniques, e.g., data clustering, is getting exceedingly important. Data clustering is to group data into clusters so that the objects within the same cluster are similar to each other, while the objects in different clusters are dissimilar to one another. For example, social network analysis, i.e., the identification and reorganization of community by certain types of interdependency among people, such as values, visions, ideas, financial exchange, friends, kinship, dislike, conflict, trade, web links, sexual relations, disease transmission, or airline routes [1], plays an important role on the study of social sciences. In the analysis of complex survey data, cluster analysis is usually the crucial step to discover patterns that will be used in the decision-makings such as the target audience for a product, the development of new products etc. Data clustering analysis is also an essential technique for behavior studies and social-economic studies.

Although data clustering is an important procedure in many fields of the social sciences, clustering analysis is a difficult problem. Many factors such as effective similarity measures, algorithms and initial parameters come into play in devising a well tuned clustering technique for a given clustering problem. Moreover, it is well known that no clustering method can be universally applicable to all sorts of data structures in terms of data distribution, size, density, dimensionality etc. A large number of clustering algorithms have been developed for different contexts or purposes. The diversity, on one hand, provides us with many choices. On the other hand, the profusion of options causes confusion [2, 3]. How to choose an effective clustering algorithm and how to assess the quality of the clusters returned are critical for the success of data clustering. However, these fundamental questions are not well addressed in the research of data clustering.

As the new developments in one discipline usually spread slowly in the relevant disciplines, the development and use of data clustering as a formal analysis procedure by anthropologists, psychologies, political scientists, and sociologists is not only promising but challenging as well. Therefore, it is desirable to have a clustering tool that assists the researchers in different disciplines in clustering analysis. A few platforms for cluster analysis have been developed for some special contexts or purposes; see [4, 5]. However, they are either limited in their scope or are far from being user-friendly. One such platform, named Cluster 3.0 [4], allows a user to cluster gene expression data for use with bioinformatics. However, cluster 3.0 only offers K-means algorithm and hierarchical clustering method, and the results generated by the platform are not easy to interpret. Another such platform, RapidMiner [5], offers a variety of clustering. RapidMiner, however, requires extensive knowledge and skills in programming, which are lacked for the general users. There is no platform so far that offers a simple and user-friendly environment for cluster analysis.

In this paper, we will describe a generalized cluster analysis platform with easy-to-use interface. The platform provides the classic clustering algorithms with both textual and visual display of clustering results for any given data set. The platform serves the researchers in social sciences easy-to-use interface and the ability for future extension to include clustering evaluation criteria. The platform will assist the users in comparing clustering results of different clustering algorithms and making a wise decision of which cluster algorithm best serves their purpose.

The rest of the paper is organized as follows. Section 2 surveys the primary data clustering algorithms and clustering metrics. The general purpose clustering analysis platform is described and demonstrated in Section 3. Our conclusions and directions for future work are presented in Section 4.

2 Methodology

To cluster a data set, one question we need to answer is which similarity measure is appropriate in a given situation. The most common similarity measurements within a set of data are distance measurements, such as Euclidean distance, a special case ($p = 2$) of Minkowski metric

$$d_p(X_i, X_j) = (\sum_{k=1}^{n} |x_{i,k} - x_{j,k}|^p)^{1/p} = \|X_i - X_j\|_p,$$

where n is data space dimension. Instead of direct use of Minkowski distance, normalization of the continuous features is usually adopted in order to avoid the tendency of the largest-scaled feature to dominate the others [3]. There are other measures used for the clustering strings, such as Cosine similarity, which is the most commonly used method for document clustering [2]. Another important question for clustering analysis is clustering metrics. Data clustering algorithms are classified into different categories based on different clustering metrics. A few categories of commonly used clustering algorithms are introduced below.

2.1 Clustering analysis

(a) K-means algorithm and its variants

K-means algorithm is the most well-known centroid algorithm which assigns every data point to whichever cluster's center is nearest. Its basic procedure is to first select k points as initial group centroids, and to repeat the following steps until there is no change in the position of any cluster's centroid.

(1) Assign each point in the data set to the cluster with the nearest centroid.

(2) Re-calculate the position of each cluster's centroid.

The clustering metric for K-means is to minimize total intra-cluster variance:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2,$$

where S_i, $i = 1, 2, \cdots k$ denote k different clusters and μ_i is the centroid or mean point of cluster S_i.

A number of variations of this algorithm have been developed [6, 7, 8], such as fuzzy c-means [6], in which a data point, instead of belonging exclusively to a single cluster, belongs to all the clusters with a degree of membership and the sum of coefficients of being in clusters is one. Compared with other clustering algorithms, the class of K-means algorithm and its variants has fast convergence rate and is relatively easy to implement. However, to apply the algorithm, the user must provide pre-specified value of k, i.e., the number of clusters, which is usually difficult to determine. In addition, this class of algorithms is susceptible to noise in the data set as each data point must belong to a cluster, an outlier can distort the shapes of clusters [2, 3].

(b) Hierarchical methods

The hierarchical methods group data points into a tree of clusters. The data points are clustered with either agglomerative (bottom up) or divisive (top-down) method and the clustering procedure is presented with the dendrogram [2, 3, 9].

The clustering metric in hierarchical methods is inter-group distance, which is defined as single-linkage, complete linkage, or average linkage. Suppose $D(r, s)$ denotes the distance between two clusters r and s. The single linkage is defined as the minimum distance of any pairwise points between two clusters, i.e., $D(r, s) = \min\{d(i, j)$: where point i is in cluster r and point j is in cluster s $\}$. The complete linkage is defined as the maximum distance of any pairwise points between two clusters, i.e., $D(r, s) = \max\{d(i, j)$: where point i is in cluster r and point j is in cluster s $\}$. The average linkage is defined as the average distance between all pairwise points between two clusters, i.e., $D(r, s) = Trs/(Nr * Ns)$, where Trs is the sum of all pairwise distances between cluster r and cluster s and Nr and Ns are respectively respectively the sizes of the clusters r and s. At each stage of agglomerative methods, the clusters r and s with minimum clustering metric are merged as a single cluster.

The advantages of hierarchical methods are that a complete hierarchy of clusters is computed and visually illustrated. In addition, they do not need to specify the number of clusters in advance.

However, they cannot perform flexible adjustments once the splitting (divisive method) or merging (agglomative method) decisions are made during the clustering process.

(c) Density-based clustering algorithms

In density-based clustering algorithms [10], clusters are interpreted as dense regions in the data space and are separated by regions of low object density (noise). These regions may have an arbitrary shape and the points inside a region may be arbitrarily distributed.

For these algorithms, the clustering metric is density of data points in a region. The key idea is that for each data of a cluster the neighborhood of a given radius (Eps) has to contain at least a minimum number of data points (MinPts). The performance of density-based clustering thus relies on the parameters of Eps and MinPts.

The density-based clustering algorithms have some advantages over k-means and hierarchical clustering methods. They are more robust in identifying clusters of arbitrary shapes and sizes and can separate from surrounding noise [2,3]. Their disadvantage is that they might be sensitive to input parameters Eps and MinPts, which are difficult to determine in advance.

(d) Expectation-Maximization (EM) algorithm and its variants

The EM algorithm [11] is used in statistics to classify each point into the most likely probabilistic model and estimate the parameters of each model. EM regards the data set as incomplete and divides each data point into two parts - the observable features and the missing data. Its basic procedure is first to initialize the distribution parameters and then repeat the following steps until the estimations of the distribution parameters are convergent.

(1) Expectation (E): computes an expectation of the likelihood by including the latent variables as if they were observed.

(2) Maximization (M): re-estimates the parameters by maximizing the expected likelihood found on the E step.

The major disadvantages of EM algorithms are the sensitivity to the selection of initial parameters, the possibility of convergence to a local optimum, and the slow convergence rate [11, 2, 3]. The variants of EM have addressed these problems.

3 Implementation of Clustering Platform

To our knowledge, there is no platform so far that offers a simple and user-friendly environment for cluster analysis. We fill that gap by developing a general purpose cluster analysis platform, which implements some commonly used clustering algorithms and has an easy-to-use interface. The platform can be used for cluster analysis and validation of general data sets.

3.1 Development of the clustering platform

To implement the general purpose platform, we choose Java programming language due to its good support of graphical user interfaces (GUI) and platform independence. In the developing process, NetBeans is used as the integrated development environment because of its visual tools that generate skeleton code. In addition, the GUI Builder of NetBeans is used to support a sophisticated yet simplified Swing Application Framework and Beans Binding.

In the platform, the computation is conducted by Matlab. Matlab is designed for convenient numerical computations, especially matrix manipulation and includes many special functions developed for specific fields such as optimization, statistics etc. We choose Matlab because it is a high performance language for computation, simulation, and visualization. In addition, there are a variety of Matlab open source codes for clustering analysis. It saves us time not to write these algorithms from scratch. The platform brings together a variety of resources for performing cluster analysis using Matlab [12]. The only problem to develop clustering algorithms in Matlab is that Matlab is not a free software. To make our platform accessible to users with no extra financial cost, we installed one license of Matlab on our server [14] so users do not need to install Matlab on their local machines. To configure Matlab server and call Matlab code from Java interface, we first download the package from [13] and then do the following:

1. Copy the exitform.fig, exitform.m, RemoteMatlab.jar and StartMatlabServer.m to our working directory.
2. Copy xmlrpc libraries from apache and some commons libraries to our working directory.
3. To launch Matlab server, type “StartMatlabServer” in Matlab command window.

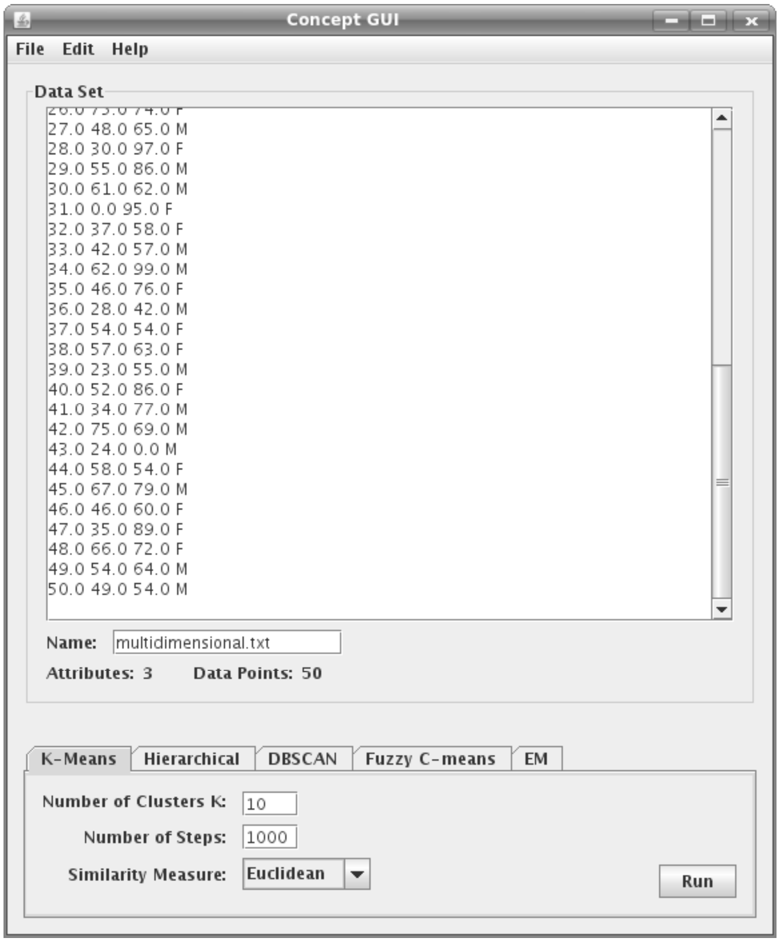
To be able to call Matlab procedures located on the server from Java program during the development of platform, we follow the steps below:

1. Install NetBeans.
2. Import lib directory library into NetBeans.
3. Import RemoteMatlab.jar into NetBeans.
4. Create instance of Matlabcilent in Java application.

3.2 Running of the clustering platform

To run the platform, a user does not need to install any software and only needs to download clusteringPlatform.jar file from our website [14] and run it on his machine with Internet access. After launching the file downloaded, the user will see a window as shown in Figure 1. The top half of the window shows the data set to be clustered, with the listing of number of data points and the number of attributes in the data set. The lower half of the window allows the user to choose which clustering algorithm to run on the current dataset and to decide the values of the parameters for the algorithm. For example, when the tab "k-means" is selected as shown in Figure 1, the user needs to input the number of clusters k and number of iterations, and select similarity measure from the pull-down list. When the user clicks on the "run" button, K-means algorithm is called and runs on Matlab server in the backend, and the results including output data file and visual display are sent back to the user and displayed in the platform on the user's machine.

To load the data set for clustering, the user can copy and paste the data set to the top window, or click on the menu "file" and choose "open" to open the right data file. The format of data set that the platform takes is similar to the relational database, i.e., each data entry is represented in one row, and each data entry consists of multiple attributes represented in columns separated by space or Tab key; see the data set in Figure 1.



```
26.0 75.0 74.0 F
27.0 48.0 65.0 M
28.0 30.0 97.0 F
29.0 55.0 86.0 M
30.0 61.0 62.0 M
31.0 0.0 95.0 F
32.0 37.0 58.0 F
33.0 42.0 57.0 M
34.0 62.0 99.0 M
35.0 46.0 76.0 F
36.0 28.0 42.0 M
37.0 54.0 54.0 F
38.0 57.0 63.0 F
39.0 23.0 55.0 M
40.0 52.0 86.0 F
41.0 34.0 77.0 M
42.0 75.0 69.0 M
43.0 24.0 0.0 M
44.0 58.0 54.0 F
45.0 67.0 79.0 M
46.0 46.0 60.0 F
47.0 35.0 89.0 F
48.0 66.0 72.0 F
49.0 54.0 64.0 M
50.0 49.0 54.0 M
```

Name: multidimensional.txt

Attributes: 3 Data Points: 50

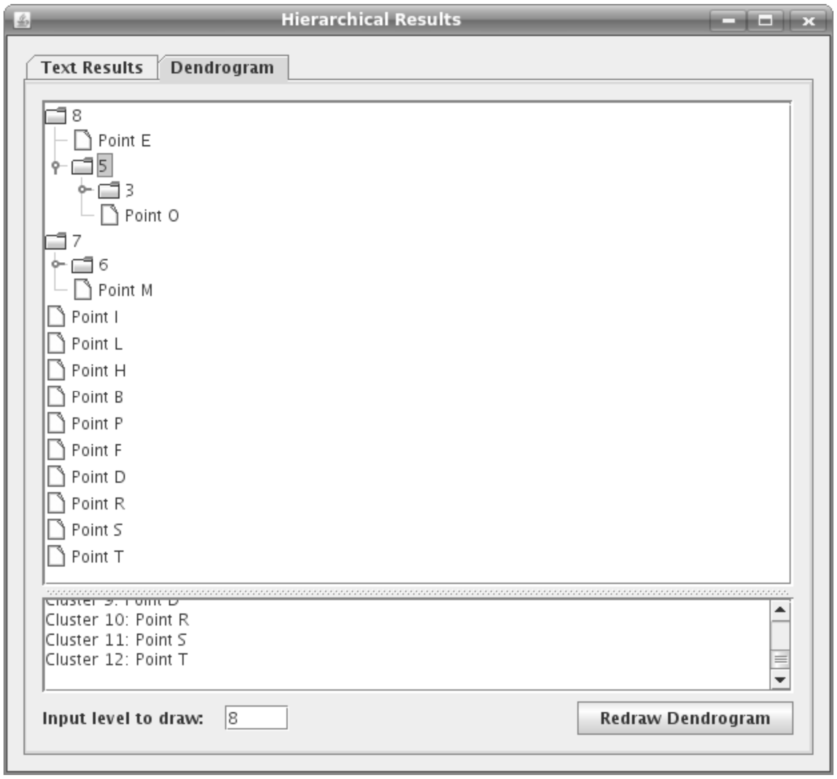
Number of Clusters K: 10

Number of Steps: 1000

Similarity Measure: Euclidean

Figure 1: Clustering platform GUI

To illustrate the visual display of the clustering results, a few snapshots are taken. In Figure 2, the hierarchical clustering result is displayed for a randomly picked data set. This dendrogram display gives the user the merge distance and the points contained within each cluster at the current level that the user is viewing. The user may pick at which level he would the dendrogram to be drawn. The default display for the dendrogram is the top level. In Figure 3, EM visual result is displayed. The chart displays the likelihood that each particular point belongs to a cluster. The darker the shade, the more likely it belongs to that cluster. Figure 4 shows K-means visual results where each cluster center is depicted as a circle. Figure 5 shows DBSCAN (Density-Based Spatial Clustering of Applications with Noise) visual results where the results display each cluster as a separate color.



```
8
  Point E
  5
    3
    Point O
7
  6
  Point M
Point I
Point L
Point H
Point B
Point P
Point F
Point D
Point R
Point S
Point T
```

```
Cluster 9: Point D
Cluster 10: Point R
Cluster 11: Point S
Cluster 12: Point T
```

Input level to draw: 8

Figure 2: Hierarchical results

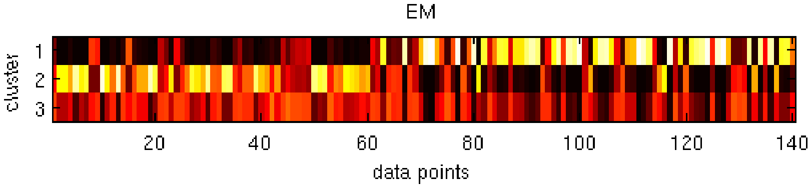


Figure 3: EM visual results

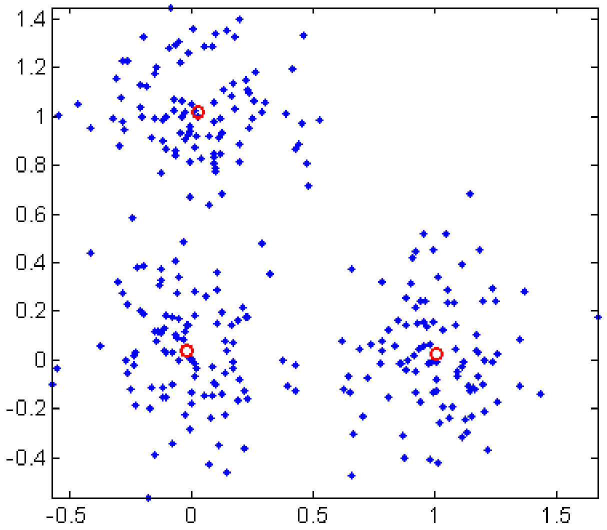


Figure 4: K-means visual results

4 Conclusion and Future Work

In this paper, a general purpose clustering analysis platform is introduced. The platform integrates a number of commonly-used clustering algorithms with friendly user interface and both textual and visual display of results. An advantage of the platform is that it is extendable, so it is easy to integrate more clustering algorithms in the future. Another advantage is that it is easy to use, so we expect that it can satisfy the need of researchers in social sciences to cluster data without writing a single line of code. A drawback of the visual display of results in our platform is that it is not capable of visualizing clustering results effectively in high-dimensional data space.

After a number of clustering algorithms have been implemented in this platform, it will be important to validate clustering results of different algorithms with different parameters. When assessing the output of a clustering algorithm, both the intraconnectivity and interconnectivity

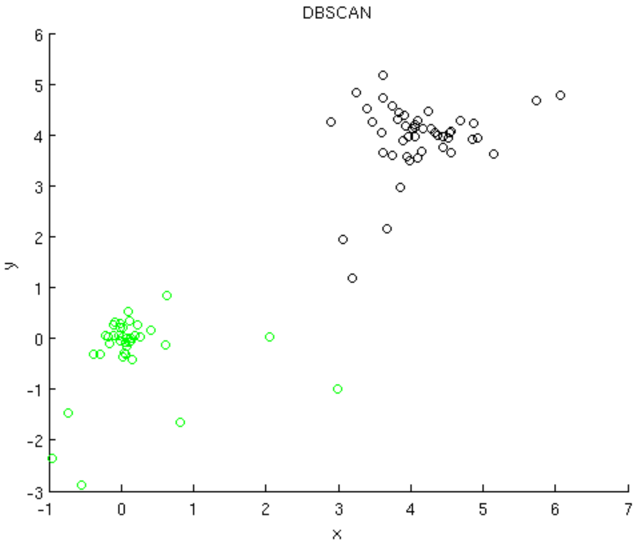


Figure 5: DBSCAN visual results

of clusters have to be taken into account to ensure that the clusters are compact and isolated. It is desirable to have a high intraconnectivity and a low degree of interconnectivity. However, there are no general standards for cluster validation in existing literature except in well-prescribed subdomains. Our future work will address the problem of clustering evaluation and integrate the evaluation criteria into the platform to assist users in making better choice of clustering algorithms and improving clustering quality.

Received: June 4, 2008. Revised: July 8, 2008.

References

[1] Wikipedia, the Free Encyclopedia, “Social Network”, 25 Jun. 2008. Wikimedia Foundation, Inc., http://en.wikipedia.org/wiki/Social_network

[2] R. Xu, and D. Wunsch II, *Survey of clustering algorithms*, IEEE Transactions on Neural Networks, 16(2005), No. 3, 645–678.

[3] A. K. Jain, M. N. Murty and P. J. Flyn, *Data clustering: a review*, ACM Computing Surveys, 31(1999), No. 3, 264–323.

[4] http://bonsai.ims.u-tokyo.ac.jp/∼mdehoon/software/cluster/software.htm

[5] http://www.rapid-i.com

[6] M. Sato, Y. Sato and L. C. Jain, *Fuzzy Clustering Models and Applications*, Physica-Verlag, 1997.

[7] Z. Huang, *Extensions to the k-means algorithm for clustering large datasets with categorical values*, Data Mining and Knowledge Discovery, 2(1998), 283–304.

[8] G. Hamerly and C. Elkan, *Learning the k in k-means*, Technical Report CS2002-0716, University of California, San Diego, 2002.

[9] N. Jardine and R. Sibson, *The construction of hierarchic and non-hierarchic classifications*, Computer Journal, 11(1968), 177–184.

[10] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, *A density-based algorithm for discovering clusters in large spatial data sets with noise*, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, 1996, 226–231.

[11] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, New York: Wiley, 1997.

[12] http://www.dcorney.com/ClusteringMatlab.html

[13] http://plasmapowered.com/wiki/index.php/Calling_MatLab_from_Java

[14] http://caplatform.ias.csusb.edu