Comparisons Between Data Clustering Algorithms

Osama Abu Abbas
Computer Science Department, Yarmouk University, Jordan

Abstract: *Clustering is a division of data into groups of similar objects. Each group, called a cluster, consists of objects that are similar between themselves and dissimilar compared to objects of other groups. This paper is intended to study and compare different data clustering algorithms. The algorithms under investigation are: k-means algorithm, hierarchical clustering algorithm, self-organizing maps algorithm, and expectation maximization clustering algorithm. All these algorithms are compared according to the following factors: size of dataset, number of clusters, type of dataset and type of software used. Some conclusions that are extracted belong to the performance, quality, and accuracy of the clustering algorithms.*

Keywords: *Clustering, k-means algorithm, hierarchical clustering algorithm, self-organizing maps algorithm, expectation maximization clustering algorithm.*

Received January 18, 200 ; accepted May 2, 2007

1. Introduction

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar amongst themselves and dissimilar compared to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification. It represents many data objects by few clusters, and hence, it models data by its clusters [3].

Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster. It is important to understand the difference between clustering (unsupervised classification) and discriminate analysis (supervised classification). In supervised classification, we are provided with a collection of labeled (preclassified) patterns; the problem is to label a newly encountered, yet unlabeled, pattern. Typically, the given labeled (training) patterns are used to learn the descriptions of classes which in turn are used to label a new pattern. In the case of clustering, the problem is to group a given collection of unlabeled patterns into meaningful clusters. In a sense, labels are associated with clusters also, but these category labels are data driven; that is, they are obtained solely from the data [5, 7, 8, 13].

Some researchers improved some data clustering algorithms, others implemented new ones, and some others studied and compared different data clustering algorithms. Following are some of the previous studies that considered the effect of different factors on the performance of some data clustering algorithms and compared the results. However, these studies differ from my analysis in the algorithms and the factors:

- [1] applied several indices to evaluate the performance of clustering algorithms, including hierarchical clustering, *k*-means, PAM and SOM. The indices were homogeneity and separation scores, silhouette width, redundant score (based on redundant genes), and WADP (testing the robustness of clustering results after small perturbation).
- [9] described the implementation of an out-of-core technique for the data analysis of very large data sets with the sequential and parallel version of the clustering algorithm AutoClass. They discussed the out-of-core technique and showed performance results in terms of execution time and speed up.
- [8] employed an agglomerative algorithm to construct a dendrogram and used a simple distinctness heuristic to extract a partition of the data. They studied the performance of Similarity-Based Agglomerative Clustering (SBAC) algorithm on real and artificially generated data sets. They demonstrated the effectiveness of this algorithm in unsupervised discovery tasks. They illustrated the superior performance of this approach by making comparisons with other clustering schemes.
- [10] showed how to perform some typical mining tasks using conceptual graphs as formal but meaningful representations of texts. Their methods involved qualitative and quantitative comparisons of conceptual graphs, conceptual clustering, building a conceptual hierarchy, and application of data mining techniques to this hierarchy in order to detect interesting associations and deviations. Their experiments showed that, despite widespread

disbelief, detailed meaningful mining with conceptual graphs is computationally affordable.

- [12] compared two graph-coloring programs: one exact and another based on heuristics which can give, however, provably exact results on some types of graphs. They proved that the exact graph coloring is not necessary for high-quality functional decomposers. Comparison of their experimental results with competing decomposers shows that for nearly all benchmarks their solutions are the best and time is usually not too high.
- Jain and Dubes [1988] and Dubes [1993] used a relative test to compare two structures and to measure their relative merit. They also discussed in detail the indices that are used for this comparison.

In this paper different data clustering algorithms that have not been considered before are compared according to different factors that haven't been studied yet.

2. How Algorithms are Implemented?

An extensive web search is done to find some data clustering algorithms implementation to test on. After selection, I ended up with two of them:

- LNKnet Software: It is public domain software made available from MIT Lincoln Laboratory [6]. It is located at the following site: (http://www.ll.mit.edu/IST/lknet).
- Cluster and TreeView Software**:** Cluster and TreeView are programs that provide a computational and graphical environment for analyzing data from different datasets [2]. It is located at the following site: (http://www.rana.lbl.gov/EisenSoftware.htm).

The reasons behind choosing these two software are:

- They are the most popular software for implementing different data clustering algorithms.
- They are very powerful in implementing data clustering algorithms.
- They implement the four data clustering algorithms that are chosen in this paper.
- They have ideal dataset as a part of them which can be used for testing and implementing the algorithms.

2.1. Data Sample

The dataset that is used to test the clustering algorithms and compare among them is obtained from the site: (http://kdd.ics.uci.edu/) or from another site, which is, (http://www.kdnuggets.com/datasets). This is a good dataset to test time series clustering algorithms because euclidean distance will not be able to achieve perfect accuracy. In particular the following pairs of classes will often be confused (normal/ cyclic) (decreasing trend/ downward shift) and (increasing trend/ upward shift). This dataset is stored in an ASCII file, 600 rows, 60 columns, with a single chart per line. The classes are organized as follows:

- 1-100 Normal
- 101-200 Cyclic
- 201-300 Increasing trend
- 301-400 Decreasing trend
- 401-500 Upward shift
- 501-600 Downward shift

However, this format is not totally suitable for the two packages used to compare the clustering algorithms (LNKnet Software and Cluster and TreeView Software). So some reasonable changes are done in the dataset format to be acceptable for the packages. The dataset formats of the two software are reviewed by referring to the manuals of the software and the changes that are done to the dataset formats do not affect the dataset itself at all.

Also, a part of this data set (200 rows and 20 columns) is taken as an input file for the clustering algorithms to study the size of the datasets (huge and small datasets) on these algorithms.

Finally, the clustering algorithms are tested using the dataset stored in the two packages themselves to study the effect of different datasets on the algorithms.

2.2. Which Algorithms are Compared?

Four different clustering algorithms are chosen to investigate, study, and compare them. The algorithms that are chosen are: *k*-means algorithm, hierarchical clustering algorithm, Self-Organization Map (SOM) algorithm and Expectation Maximization (EM) clustering algorithm. The general reasons for selecting these four algorithms are:

- Popularity.
- Flexibility.
- Applicability.
- Handling high dimensionality.

However, detailed reasons behind selecting every algorithm are listed in the context. In this section, for every algorithm some idea is given about it; how it works and the reasons for choosing it.

2.2.1. K-means Algorithm

K-means is a well-known partitioning method. Objects are classified as belonging to one of k groups, *k* chosen a priori. Cluster membership is determined by calculating the centroid for each group (the multidimensional version of the mean) and assigning each object to the group with the closest centroid. This approach minimizes the overall within-cluster dispersion by iterative reallocation of cluster members.

In a general sense, a *k*-partitioning algorithm takes as input a set *S* of objects and an integer *k*, and outputs a partition of *S* into subsets $S_1,...S_2$, S_k. It uses the sum of squares as the optimization criterion. Let x^i_r be the r^{th} element of S_i, $|S_i|$ be the number of elements in S_i, and $d(x^i_r , x^i_s)$ be the distance between x^i_r and x^i_s. The sum of squares criterion is defined by the cost function:

$$c(S_i) = \sum_{r=1}^{|S_i|} \sum_{x=1}^{|S_i|} (d(x^i_r, x^i_s))^2 \quad (1)$$

In particular, *k*-means works by calculating the centroid of each cluster S_i, denoted x^{-i}, and optimizing the cost function:

$$c(S_i) = \sum_{r=1}^{|S_i|} (d(x^{-i}, x^i_r))^2 \quad (2)$$

The goal of the algorithm is to minimize the total cost:

$$c(S_i) + ... + c(S_k) \quad (3)$$

Here, the pseudo code of the *k*-means algorithm is to explain how it works:

A. Choose K as the number of clusters.
B. Initialize the codebook vectors of the K clusters (randomly, for instance)
C. For every new sample vector:

C.1. Compute the distance between the new vector and every cluster's codebook vector.
C.2. Recompute the closest codebook vector with the new vector, using a learning rate that decreases in time.

The reason behind choosing the *k*-means algorithm to study is its popularity for the following reasons:

- Its time complexity is $O(nkl)$, where *n* is the number of patterns, *k* is the number of clusters, and *l* is the number of iterations taken by the algorithm to converge.
- Its space complexity is $O(k+n)$. It requires additional space to store the data matrix.
- It is order-independent; for a given initial seed set of cluster centers, it generates the same partition of the data irrespective of the order in which the patterns are presented to the algorithm.

2.2.2. Hierarchical Clustering Algorithm

Partitioning algorithms are based on specifying an initial number of groups, and iteratively reallocating objects among groups to convergence. In contrast, hierarchical algorithms combine or divide existing groups, creating a hierarchical structure that reflects the order in which groups are merged or divided. In an agglomerative method, which builds the hierarchy by merging, the objects initially belong to a list of singleton sets $S_1,..., S_2, S_n$. Then a cost function is used to find the pair of sets $\{S_i, S_j\}$ from the list that is the "cheapest" to merge. Once merged, S_i and S_j are removed from the list of sets and replaced with $S_i \cup S_j$. This process iterates until all objects are in a single group. Different variants of agglomerative hierarchical clustering algorithms may use different cost functions. Complete linkage, average linkage, and single linkage methods use maximum, average, and minimum distances between the members of two clusters, respectively.

Following is the pseudo code of the hierarchical clustering algorithm to explain how it works:

- Compute the proximity matrix containing the distance between each pair of patterns. Treat each pattern as a cluster.
- Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.
- If all patterns are in one cluster, stop. Otherwise, go to step 2.

The advantages of the hierarchical clustering algorithms are the reason this algorithm was chosen for discussion. These advantages include:

- Embedded flexibility regarding a level of granularity.
- Ease of handling of any forms of similarity or distance.
- Consequently applicability to any attributes types.
- Hierarchical clustering algorithms are more versatile.

2.2.3. Self-Organization Map Algorithm

Inspired by neural networks in the brain, Self-Organization Map (SOM) uses a competition and cooperation mechanism to achieve unsupervised learning. In the classical SOM, a set of nodes is arranged in a geometric pattern, typically 2-dimensional lattice. Each node is associated with a weight vector with the same dimension as the input space. The purpose of SOM is to find a good mapping from the high dimensional input space to the 2-D representation of the nodes. One way to use SOM for clustering is to regard the objects in the input space represented by the same node as grouped into a cluster. During the training, each object in the input is presented to the map and the best matching node is identified. Formally, when input and weight vectors are normalized, for input sample $x(t)$ the winner index *c* (best match) is identified by the condition:

$$for\ all\ i,\ \ \|x(t) - m_c(t)\| \le \|x(t) - m_i(t)\| \quad (4)$$

where *t* is the time step in the sequential training, m_i is the weight vector of the i^{th} node. After that, weight vectors of nodes around the best-matching node $c = c(x)$ are updated as:

$$m_i(t+1) = m_i(t) + ah_{c(x),i}(x(t) - m_i(t)) \quad (5)$$

where α is the learning rate and $h_{c(x),i}$ is the "neighborhood function", a decreasing function of the distance between the i^{th} and c^{th} nodes on the map grid. To make the map converge quickly, the learning rate and neighborhood radius are often decreasing functions of t. After the learning process finishes, each object is assigned to its closest node. There are variants of SOM to the above classical scheme.

Following is the pseudo code of the SOM algorithm to explain how it works:

A. Choose the dimension and size of the map.
B. For every new sample vector:

B.1. Compute the distance between the new vector and every cluster's codebook vector.
B.2. Recompute all codebook vectors with the new vector, using both a distance radius on the map and learning rate that decrease in time.

The following advantages of SOM are behind choosing this algorithm for studying:

- While the voronoi regions of the map units are convex, the combination of several map units allows the construction of non-convex clusters.
- Different kinds of distance measures and joining criteria can be utilized to form the big clusters.
- It has been successfully used for vector quantization and speech recognition.
- The SOM generates a sub-optimal partition if the initial weights are not chosen properly.

2.2.4. The Expectation Maximization Clustering Algorithm

Expectation Maximization (EM) is a well-established clustering algorithm in the statistics community. EM is a distance-based algorithm that assumes the data set can be modeled as a linear combination of multivariate normal distributions and the algorithm finds the distribution parameters that maximize a model quality measure, called *log likelihood*.

EM is chosen to cluster data for the following reasons among others:

- It has a strong statistical basis.
- It is linear in database size.
- It is robust to noisy data.
- It can accept the desired number of clusters as input.
- It can handle high dimensionality.
- It converges fast given a good initialization.

3. How Algorithms are Compared?

The four clustering algorithms are compared according to the following factors:

- The size of the dataset.
- Number of the clusters.
- Type of dataset.
- Type of software.

For each factor, four tests are made, one for each algorithm. For example, according to the size of data, each of the four algorithms: *k*-means, Hierarchical Clustering, SOM, and EM is executed twice; first by trying a huge dataset and then by trying a small dataset. Table 1 explains how the four algorithms are compared. The total number of times the algorithms have been executed is 32. For each 8-runs group, the results of the executions are studied and compared. The conclusions are written down. This step is repeated for all the factors.

Table 1. The factors according to which the algorithms are compared.

	Size of Dataset	**Number of Clusters**	**Type of Dataset**	**Type of Software**
k-means Alg.	Huge Dataset & Small Dataset	Large number of clusters & Small number of clusters	Ideal Dataset & Random Dataset	LNKnet Package & Cluster and TreeView Package
HC Alg.	Huge Dataset & Small Dataset	Large number of clusters & Small number of clusters	Ideal Dataset & Random Dataset	LNKnet Package & Cluster and TreeView Package
SOM Alg.	Huge Dataset & Small Dataset	Large number of clusters & Small number of clusters	Ideal Dataset & Random Dataset	LNKnet Package & Cluster and TreeView Package
EM Alg.	Huge Dataset & Small Dataset	Large number of clusters & Small number of clusters	Ideal Dataset & Random Dataset	LNKnet Package & Cluster and TreeView Package

According to the number of clusters, *k* (see Table 2), except for hierarchical clustering, all clustering algorithms compared here require setting *k* in advance (for SOM, *k* is the number of nodes in the lattice). Here, the performance of different algorithms for different *k*'s is compared in order to test the performances that are related to *k*. To simplify the situation and to make the comparisons easier, *k* is chosen equal to 8, 16, 32, and 64, and the lattices for SOM are the square of them.

To compare hierarchical clustering with other algorithms, the hierarchical tree is cut at two different levels to obtain corresponding numbers of clusters (8, 16, 32 and 64). As a result, as the value of *k* becomes greater the performance of SOM algorithm becomes lower. However, the performance of *k*-means and EM

algorithms become better than hierarchical clustering algorithm.

Table 2. The relationship between number of clusters and the performance of the algorithms.

Number Of Clusters (K)	Performance			
	SOM	*k*-Means	EM	HCA
8	59	63	62	65
16	67	71	69	74
32	78	84	84	87
64	85	89	89	92

According to the accuracy (see Table 3), SOM shows more accuracy in classifying most the objects to their clusters than other algorithms. But as the number of *k* becomes greater the accuracy of hierarchical clustering becomes better until it reaches the accuracy of SOM algorithm. *k*-means and EM algorithms have less quality (accuracy) than the others. However, all the algorithms have some ambiguity in some noisy data to be clustered.

Table 3. The relationship between number of clusters and the quality of the algorithms.

Number Of Clusters (K)	Quality			
	SOM	K-Means	EM	HCA
8	1001	1112	1101	1090
16	920	1089	1076	960
32	830	910	898	850
64	750	840	820	760

According to the size of dataset (see Table 4), a huge dataset is used consisting of 600 rows and 60 columns and a small dataset using 200 rows and 20 columns. The small dataset is extracted as a subset of the huge dataset. The quality of EM and *k*-means algorithms becomes very good when using a huge dataset. The other two algorithms hierarchical clustering and SOM algorithms show good results when using a small dataset. As a conclusion, partitioning algorithms (like *k*-means and EM) are used for huge dataset while hierarchical clustering algorithms are used for small dataset.

Table 4. The affect of the data size on the algorithms.

K=32				
Data Size	SOM	K-Means	EM	HCA
36000	830	910	898	850
4000	89	95	93	91

According to the type of dataset (see Table 5), a random dataset is used which is extracted from the internet and used for different jobs. On the other hand, an ideal dataset is used which is part of the software (LNKnet and Cluster and TreeView). It is ideal because it is designed to be suitable for testing and training the software itself and having less noisy data which leads to ambiguity. As a result, hierarchical clustering and SOM algorithms give better results than *k*-means and EM algorithms when using random dataset and the vice versa. This indicates that *k*-means and EM algorithms are very sensitive for noise in the dataset. This noise makes it difficult for the algorithm to include an object in a certain cluster. This will affect the results of the algorithm. However, hierarchical clustering algorithm is more sensitive for noisy dataset than SOM algorithm.

Table 5. The affect of the data type on the algorithms.

K=32				
Data Type	SOM	K-Means	EM	HCA
Random	830	910	898	850
Ideal	798	810	808	829

According to the type of software, two packages are used to compare between the algorithms: LNKnet (UNIX environment) and cluster and TreeView (WINDOWS environment). However, running the clustering algorithms using any one of them gives almost the same results even when changing any of the other three factors (dataset size, clusters number and dataset type). This, I believe, is because most software use the same procedures and ideas in any algorithm implemented by them.

4. Conclusions

After analyzing the results of testing the clustering algorithms and running them under different factors and situation, the following conclusions are obtained:

- As the number of clusters, *k* becomes greater; the performance of SOM algorithm becomes lower.
- The performance of *k*-means and EM algorithms is better than hierarchical clustering algorithm.
- SOM algorithm shows more accuracy in classifying most the objects into their suitable clusters than other algorithms.
- As the value of *k* becomes greater, the accuracy of hierarchical clustering becomes better until it reaches the accuracy of SOM algorithm.
- *k*-means and EM algorithms have less quality (accuracy) than the others.
- All the algorithms have some ambiguity in some (noisy) data when clustered.
- The quality of EM and *k*-means algorithms become very good when using huge dataset.
- Hierarchical clustering and SOM algorithms show good results when using small dataset.
- As a general conclusion, partitioning algorithms (like *k*-means and EM) are recommended for huge dataset while hierarchical clustering algorithms are recommended for small dataset.
- Hierarchical clustering and SOM algorithms give better results compared to *k*-means and EM algorithms when using random dataset and the vice versa.
- *k*-means and EM algorithms are very sensitive for noise in dataset. This noise makes it difficult for the algorithm to cluster an object into its suitable cluster. This will affect the results of the algorithm.

- Hierarchical clustering algorithm is more sensitive for noisy dataset than SOM algorithm.
- Running the clustering algorithms using any software gives almost the same results even when changing any of the factors because most software use the same procedures and ideas in any algorithm implemented by them.

5. Future Work

This paper was intended to compare between some data clustering algorithms. Through my extensive search I was unable to find any study that attempts to compare between the four clustering algorithms under investigation.

As a future work, comparisons between these four algorithms (or may other algorithms) can be attempted according to different factors other than those considered in this paper. One important factor is normalization. Comparing between the results of algorithms using normalized data or non-normalized data will give different results. Of course normalization will affect the performance of the algorithm and the quality of the results. Another approach may consider using data clustering algorithms in applications such as object and character recognition or information retrieval which is concerned with automatic storage and retrieval of documents.

References

[1] Chen G., Jaradat S., Banerjee N., Tanaka T., Ko M., and Zhang M., "Evaluation and Comparison of Clustering Algorithms in Analyzing ES Cell Gene Expression Data," *Statistica Sinica*, vol. 12, pp. 241-262, 2002.

[2] Eisen M., *Cluster and Tree View Manual*, Stanford University, 1998.

[3] Han J. and Kamber M., *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publishers, 2001.

[4] Jain A., Murty M., and Flynn P., "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, 1999.

[5] Keogh E., Chakrabarti K., Pazzani M., and Mehrotra S., "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Knowledge and Information Systems,* vol. 3, pp. 263-286, 2001.

[6] Kukolich L. and Lippmann R., *LNKnet User's Guide*, MIT Lincoln Laboratory, 1999.

[7] Lepere R. and Trystram D., "A New Clustering Algorithm for Large Communication Delays," *in Proceedings of 16th IEEE-ACM Annual International Parallel and Distributed Processing Symposium (IPDPS'02*), Fort Lauderdale, USA, 2002.

[8] Li C. and Biswas G., "Unsupervised Learning with Mixed Numeric and Nominal Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 4, pp. 673-690, 2002.

[9] Masciari E., Pizzuti C., and Raimondo G., "Using an Out-of-Core Technique for Clustering Large Data Sets," *in Proceedings of 12th International Workshop on Database and Expert Systems Applications (DEXA)*, Munich, Germany, pp. 133-137, 2001.

[10] MontesY-G Ḿez M., Gelbukh A., and LṔez-LṔez A., "Text Mining at Detail Level Using Conceptual Graphs," *Lecture Notes in Computer Science* vol. 2393, pp. 122-136, 2002.

[11] Ordonez C. and Cereghini P., "SQLEM: Fast Clustering in SQL using the EM Algorithm," *in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data,* Dallas, United States, pp. 559-570, 2000.

[12] Perkowski M., Malvi R., Grygiel S., Burns M., and Mishchenko A., "Graph Coloring Algorithms for Fast Evaluation Of Curtis Decompositions," *in Proceedings of the 36th Design Automation Conference(DAC)*, ACM, Louisiana, pp. 225-230, 1999.

[13] Riabov A., Liu Z., Wolf L., Yu S. and Zhang L., "Clustering Algorithms for Content-Based Publication-Subscription Systems," *in Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02),* USA, pp. 133, 2002.

[14] Zha H., He X., Ding C., Simon H., and Gu M., "Bipartite Graph Partitioning and Data Clustering," *in Proceedings of the 10th International Conference on Information and Knowledge Management*, ACM Press, pp. 25-32, 2001.

Osama Abu Abbas is an instructor in the department of Computer Science at Yarmouk University. He got the BSc in Computer Science from Yarmouk University, Jordan in 1992. He then got the Master degree in Computer Science and Information from Yarmouk University, Jordan in 2003. His teaching interest focus on algorithms design and analysis, data structure, compiler construction, and artificial intelligent programming.