CLUSTERING WITH PROTOTYPE ENTITY SELECTION COMPARED WITH K-MEANS

Eva Kovacs, Iosif Ignat

Technical University of Cluj-Napoca
Faculty of Automation & Computer Science
Computer Science Department
Cluj-Napoca,
evakovacs@hotmail.com, Iosif.Ignat@cs.utcluj.ro

Abstract: *Clustering is an important area of application for a variety of fields including data mining. This paper presents a new clustering method namely, the Clustering with Prototype Entity Selection (abbreviated CPES), proposed as a method of clustering for data mining. The CPES method is original to the authors. The paper describes its mathematical essence, presents the algorithm and the experimental results obtained as compared to the K-Means method. The K-Means algorithm is by far the most widely used method for discovering clusters in data. [2][9][13][14]*

Keywords: *Data Mining, Knowledge Discovery, Segmentation, Clustering, Cluster Analysis*

1. INTRODUCTION

Data mining is a new discipline in development that takes and uses resources and ideas from several different fields. Knowledge Discovery aims at extracting new and useful information from databases. [8] *Data mining* techniques are used to discover models, structures, and regularities in large databases. [1] *Data mining* algorithms can be categorized according to the representation of models, the incoming data and the field of application. Algorithms can belong to four categories: *predictive modeling, database segmentation, link analysis* and *deviation detection.* [5]

The aim of the paper is the description of the *CPES* algorithm and its evaluation on test data as compared to the *K-Means* method. The *CPES* algorithm belongs to the database segmentation category. This is a method that distributes the objects from a set of incoming data into different clusters. The developed method can be used only with numerical incoming data. In case the data in the database belong to other types as well, they need to be preprocessed.

Clustering is an important area of application for a variety of fields including data mining. [1][6][8] *K-Means* clustering is a popular clustering method (also known as MacQueen's algorithm [12]). *K-Means* method is based on unsupervised learning and it partitions a set of *n* objects in *k* clusters, so that the similarity level in a cluster is the highest possible, and the similarity level among different clusters is the lowest possible.

2. RELATED WORK

Existing clustering algorithms can be broadly classified into *hierarchical* and *partitioning clustering* algorithms. [5] Hierarchical algorithms decompose a data set *D* of *n* objects into several levels of nested partitioning, represented by a *dendrogram*, i.e. a tree that iteratively splits *D* into smaller subsets until each subset consists of only one object. In such a hierarchy, each node of the tree represents a cluster of *D*. The *Single-Linkage* method is a commonly used hierarchical clustering method. [5] Starting with the clustering obtained by placing every object in a unique cluster, in every step the two closest clusters in the current clustering are merged until all points are in one cluster. Other algorithms which in principle produce the same hierarchical structure have also been suggested. [5]

Partitioning algorithms construct a single level partition of a data set *D* of *n* objects into a set of *k* clusters such that the objects in a cluster are more similar to each other than to objects in different clusters.

Partitioning algorithms typically represent clusters by a *prototype*. Objects are assigned to the cluster represented by the most similar prototype. These clustering algorithms are effective in determining a good clustering if the clusters are of convex shape, similar size and density, and if their number *k* can be reasonably estimated. Depending on the kind of prototypes, one can distinguish *K-Means*, *K-Modes* and *K-Medoid* algorithms. For *K-Means* algorithms [12], the prototype is the mean value of all objects belonging to a cluster.

K-Means clustering is a popular clustering method (also known as MacQueen's algorithm [12]). Note that *K-Means* is defined over numerical (continuous-valued) data since it requires the ability to compute the mean.

K-Means method is based on unsupervised learning and it partitions a set of *n* objects in *k* clusters, so that the similarity level in a cluster is the highest possible, and the similarity level among different clusters is the lowest possible.

K-Means algorithm functions as follows: *k* objects are selected at random, at the beginning each represents the mean, the centre of a cluster. Then the rest of the objects are attributed to the most alike cluster, i.e. to the centre of which it is the closest. The mean of each cluster is recalculated, and then each object is reattributed to the clusters, taking into consideration the newly calculated mean. This process continues until the criterion function converges.

The *K-Means* algorithm has the following characteristics [7]:

- it is efficient in processing large data sets;
- it often terminates at a local optimum;
- it works only on numerical data;
- the clusters have convex shapes.

Due to these properties the algorithm is successfully used in data mining. However, *K-Means* method has an inconvenience, the user must specify the number of clusters *k*, and for a better result of the algorithm, the first representative objects must be selected in an optimal manner.

The *K-Modes* [7] algorithm extends the *K-Means* paradigm to categorical domains. For *K-Medoid* algorithms [1], the prototype, called the *medoid*, is one of the objects located near the centre of a cluster. The algorithm CLARANS [5] is an improved *K-Medoid* type algorithm restricting the huge search space by using two additional user-supplied parameters. It is significantly more efficient than the well-known *K-Medoid* algorithms PAM and CLARA, nonetheless producing a result of nearly the same quality.

3. CLUSTERING WITH PROTOTYPE ENTITY SELECTION

This section presents the CPES method. [10][11] As incoming data we have the data set $X = \{x_1, x_2, ..., .x_n\}$ of *n* objects that will be the object of clustering. Each object is an entry in the database. It is compulsory that the attributes of the objects are numeric data.

If $x_k \in X$, the attributes according to which clustering is performed are $x_k = \{x_{k1}, x_{k2}, ..., .x_{kp}\}$, where *p* is the number of the attributes. Incoming data must be of numeric type because the distance between objects is used to determine the existing clusters. [16] *Euclidean* measure is chosen where the distance is calculated according to the following formula:

$$d(x_i, x_j) = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + ... + \left|x_{ip} - x_{jp}\right|^2} \qquad (1)$$

where $x_i = \left(x_{i1}, x_{i2}, ..., x_{ip}\right)$ and $x_j = \left(x_{j1}, x_{j2}, ..., x_{jp}\right)$ are two *p* dimensional objects.

In the developed method a fitness function is used, defined as follows:

$$f(x_i) = \sum_{k=1}^{n} \frac{1}{d(x_i, x_k) + A} \qquad (2)$$

where *A* is a constant.

The method is based on finding the local maximums of the fitness function. Hill climbing is an optimization technique which belongs to the family of local search. Hill climbing attempts to maximize (or minimize) a function *f(x)* until a local maximum (or minimum) is reached. Hill climbing is used widely in artificial intelligence fields. The CPES clustering method uses this technique to search the local maximum of the *f* function (2). These local maximums will be considered as the representatives of the clusters, named prototype entities. We consider that the number of local maximums will be identical with the optimal number of clusters for the data set proposed for clustering. All points from the data set will belong to a representative object. A cluster is formed of the prototype entity and the objects that belong to this representative object.

3.1 Initialization

The proposed method begins with an initialization phase in which the constant values used in the algorithm are calculated. Thus the algorithm is executed in optimal runtime. Consequently, the average distance between objects is calculated, constant *A* and radius *r*

$$d_{av} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} d(x_i, x_j)}{n(n-1)} \qquad (3)$$

$$A = \frac{d_{av}}{n} \qquad (4)$$

$$r = \frac{d_{av}}{2} \qquad (5)$$

Having these three pieces of information, the value of the fitness function is calculated for each object using (2). The values of the fitness function will be used within the clustering algorithm. Since these are calculated only once, at the beginning of execution of the algorithm, runtime is shorter then if it were calculated after each step of the algorithm.

3.2 Clustering

The actual clustering follows the initial phase. It is considered that at the beginning each object belongs to a cluster. After each step of the algorithm the number of clusters diminishes and several objects will belong to one cluster. Clustering is a cycle that is executed until the stop condition is achieved. Further on a step of this cycle is described.

For each object x_i another one is chosen from among the objects that are in its radius *r*, with which it forms a pair. This constraint:

- limits the massive migration between clusters;
- prevents the destruction of useful clusters;
- ensures a higher probability for finding each solution.

The partner of object x_i will be chosen from the vicinity $V(x_i, r)$ of x_i, in compliance with the value of the fitness function *f*. If x_j is an object from the vicinity $V(x_i, r)$ of x_i, the probability that this x_j is selected as the partner of x_i is noted with $p(x_j)$ and defined as:

$$p(x_j) = \frac{f(x_j)}{\sum_{a \in V(x_i, r)} f(a)} \qquad (6)$$

In order to choose the partner of an object we use proportional selection. We compare the values of function *f* for the two objects x_i and x_j, and choose the object that has a higher value for the fitness function. Namely, out of x_i and x_j that object will be chosen whose function value *f*, $f(x_i)$ or $f(x_j)$ is higher.

3.3 Stop condition

The CPES algorithm ends when the stop condition is fulfilled, meaning that the position of objects within the clusters remains the same after a step is completed.

$$cluster(x_i) = cluster(x'_i) \qquad (7)$$

for each *i=1,..,n*, where $cluster(x_i)$ is the cluster to which x_i belongs after *k* steps and $cluster(x'_i)$ is the cluster to which x_i belongs after *k+1* steps. The stop condition is always achieved, in worst case scenario all the objects will be classified in only one cluster.

4. THE CPES ALGORITHM

The use of the CPES method is proposed as a method of clustering for data mining. By using this method, the user does not need to specify the number of clusters, it is the algorithm that will obtain this number. The method also ensures optimal clustering.

In brief the algorithm looks as follows:

```
1. Initialization of constants $d_{av}$, A, r and fitness function f
2. Generation of clusters $cluster(x_i) = x_i$, for i=1,..,n
3. repeat
   3.1. For each $cluster(x_i)$ a pair, $x_j$ is chosen
   3.2. if $f(cluster(x_i)) < f(x_j)$
            set $cluster(x_i) = x_j$
   until there are no changes in the clusters.
```

The steps of the algorithm are describes as follows:

1. Constants d_{av}, *A*, *r* are initialized and the values of the fitness function *f* for each object of the data set is calculated using (3), (4), (5) and (2).

2. After initialization *n* clusters are defined, each object will have its own cluster, namely $cluster(x_i) = x_i$ for each i=1,..,n. $cluster(x_i)$ is the cluster to which x_i belongs. At this step the value of the cluster $cluster(x_i)$ is exactly x_i and there are *n* clusters with different values.

3. A cycle is repeated until there are no modifications in the values of $cluster(x_i)$ for each *i=1,..,n* from one step to the next. The cycle is repeated until the stop condition is fulfilled, namely until there are no changes in the values of $cluster(x_i)$ for each *i=1,..,n* from one step to the next.

3.1. For each $cluster(x_i)$ a pair is chosen. This pair is chosen from among the objects of $cluster(x_j)$ so that $i \neq j$ and on the condition that $cluster(x_j)$ is in the radius *r* of object $cluster(x_i)$, namely

$$cluster(x_j) \in V(cluster(x_i), r)$$

For the actual choosing the above-described proportional selection is used, and the chosen pair is noted with $cluster(x_p)$.

3.2. The values of function *f* are compared for objects $cluster(x_i)$ and $cluster(x_p)$, and that object is chosen which has a higher value for fitness function *f*. If the condition $f(cluster(x_i)) < f(cluster(x_p))$ is true, then the value of the cluster x_i is set with the new value of $cluster(x_p)$. If the condition is not true, then the value of cluster x_i remains unchanged.

In each step of the algorithm the number of different clusters will diminish. Finally there will be only *m* distinct values in $cluster(x_i)$. They are marked with c_j, *j=1,..,m*. These distinct values can be considered as prototype entities for their clusters, and the x_i objects for which $cluster(x_i) = c_j$ belong to cluster c_j. The number of clusters thus obtained will be equal with *m* and for each object a cluster is determined to which it belongs.

By CPES method a clean clustering is obtained with an optimal number of clusters for incoming data.

5. EXPERIMENTAL STUDY

The efficiency of the *CPES* method results from the tests on the same data sets of the algorithm presented in comparison with the *K-Means* method and with other methods presented in specialized literature. [3][4][9][15]

5.1 Data set no. 1

The first data set has 800 records and two attributes. The set is formed of two close clusters. The data set was used by A. Ultsch to test the algorithm for clustering proposed in [15].

CPES algorithm has found two clusters in the data set, and clustered the objects in these clusters. Since the data set has only two attributes, data can be presented as points in a two-dimensional system of coordinates. The data set is visualized in Fig. no. 1.

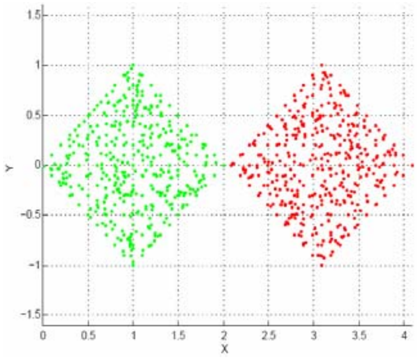

Fig. 1. Data set no. 1

Table 1: Confusion matrix for set no. 1

	Cluster 1	Cluster 2
Cluster 1	**397**	**3**
Cluster 2	**12**	**388**

There are certain objects that had been inaccurately clustered by the CPES algorithm, more precisely 15 objects out of 800. The confusion matrix for the test set is presented in Table 1.

The error rate is (3+12)/800 = 1.875 %. This error rate is very low; the accuracy of the algorithm was almost 100%.

5.2 Data set no. 2

The data set has 300 records and ten attributes. This set is formed of two clusters and was used by J.K. Vermont and J. Magidson in [13] and [14].

CPES algorithm has found two clusters in the data set, and clustered the objects in these clusters. The data set has ten attributes, thus data cannot be presented as points in a two-dimensional system of coordinates. The result of clustering cannot be visually exemplified if all the ten attributes are considered.

For this data set the error rate is 0. Each object is correctly clustered. For this data set the accuracy of the algorithm is 100%. The confusion matrix for the test set is presented in Table 2.

Table 2: Confusion matrix for set no. 2

	Cluster 1	Cluster 2
Cluster 1	**200**	**0**
Cluster 2	**0**	**100**

5.3 Data set no. 3

The third data set has 212 records and three attributes. The set is formed of seven clusters of different density within the clusters. A. Ultsch used the data set to test the clustering algorithm proposed in [15]. The data set is visually exemplified in Fig. 2.

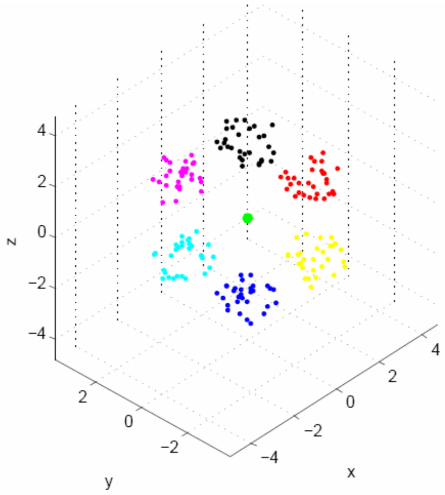

Fig. 2. Data set no. 3

The CPES algorithm has found seven clusters in the data set, and clustered the objects in these clusters. Since the data set has three attributes, data can be presented as points in a three-dimensional system of coordinates. For this data set the error rate is equal with 0. Each object is correctly clustered. For this data set the accuracy of the algorithm is 100%. The confusion matrix is presented in Table 3.

Table 3: Confusion matrix for set no. 3

	1	2	3	4	5	6	7
Clust. 1	**32**	**0**	**0**	**0**	**0**	**0**	**0**
Clust. 2	**0**	**30**	**0**	**0**	**0**	**0**	**0**
Clust. 3	**0**	**0**	**30**	**0**	**0**	**0**	**0**
Clust. 4	**0**	**0**	**0**	**30**	**0**	**0**	**0**
Clust. 5	**0**	**0**	**0**	**0**	**30**	**0**	**0**
Clust. 6	**0**	**0**	**0**	**0**	**0**	**30**	**0**
Clust. 7	**0**	**0**	**0**	**0**	**0**	**0**	**30**

6. RESULTS

The CPES method for above-mentioned data sets is compared with other clustering algorithms used in data mining and also with the *K-Means* method.

For the first data set *Single-Linkage* and *K-Means* methods are tested. Table 4 presents the number of objects that had been badly clustered due to the method used. It is noticeable that the CPES method had an error rate of 1.87%, *K-Means* had an error rate that equals 3.12% and the *Single-Linkage* method had an error rate of 50%.

Table 4: Incorrectly classified objects for set no. 1

	Cluster 1	Cluster 2
CPES	**3**	**12**
Single-Linkage	**0**	**400**
K-Means	**0**	**25**

For the second data set supervised learning methods: *Discriminant analysis* and *Logistic regression*, as well as non-supervised learning methods like *LC Cluster*, *Single-Linkage* and *K-Means* are tested. Table 5 presents the number of objects that had been badly clustered due to the method used.

It is noticeable that only the CPES method had an error rate equal with 0. All the other methods had higher error rates. *Discriminant analysis* had an error rate of 1.33%, *Logistic regression* an error rate of 1.66%, *LC Cluster* an error rate of 1.33% %, *Single-Linkage* an error rate of 2.66% and *K-Means* had the highest error rate of all the analyzed methods, 8%.

Table 5: Incorrectly classified objects for set no. 2

	Cluster 1	Cluster 2
CPES	**0**	**0**
Discriminant analysis	**1**	**3**
Logistic regression	**2**	**3**
LC Cluster	**1**	**3**
K-Means	**18**	**6**
Single-Linkage	**0**	**8**

For the third data set *Single-Linkage* and *K-Means* methods are tested. Both methods, as well as the CPES algorithm have 100% efficiency. For this data set each object is correctly classified.

The proposed *CPES* method is more efficient than *K-Means* and other clustering algorithms used in data mining. The results obtained with different data sets demonstrate the efficiency of this new algorithm. The sets used differ in the number of attributes and the number of the found clusters. The data sets were not specially created for this algorithm, in order not to demonstrate a false correctness of the method; they had been created and used by other researchers.

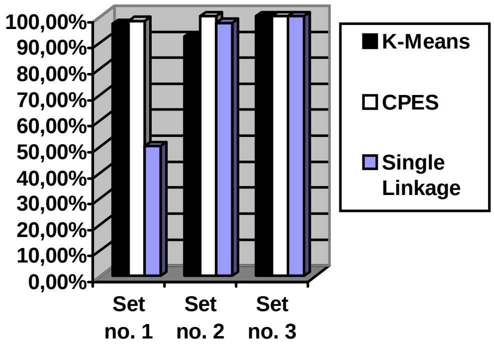

Fig. 3. Results of the K-Means method vs. CPES and Single-Linkage methods

For the third case presented, the results of the CPES algorithm are identical with the results of the *K-Means* algorithm. The CPES method however has a huge advantage over the *K-Means* and *Single-Linkage* methods: the user does not have to specify the number of clusters to be classified.

The CPES algorithm finds itself the optimal number of clusters, and then it classifies all the objects in these clusters. The *K-Means* and *Single-Linkage* algorithms need this number as incoming data. [6][9][13] If the data mining analyst fails to give the optimal number of clusters, clustering by *K-Means* and *Single-Linkage* will be inconclusive.

7. CONCLUSIONS

This paper proposed a new method of clustering called *Clustering with Prototype Entity Selection*, *CPES*. This method has been developed to be used in data mining.

The main advantage of our method, when compared to the clustering algorithms proposed in the literature, is that it does not require as incoming data the number of clusters. Instead, the CPES algorithm finds itself the optimal number of clusters, and then classifies all the objects in these clusters. For other methods the data mining analyst must configure many parameters so that the algorithm has optimal results. If these parameters are incorrectly configured the algorithm will provide inconclusive results. The method proposed by this paper is easily used by the analyst and has better or as good results as the methods that need complicated configuration of the used parameters.

The efficiency of the new algorithm has been demonstrated in comparison with the *K-Means* algorithm by presenting the results of performed experimental studies.

REFERENCES

[1] Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. Zanasi, A., Discovering Data Mining From Concept to Implementation; Prentice Hall PTR, 1998.

[2] Elkan, C., Using the Triangle Inequality to Accelerate k-Means, Proceedings of the Twentieth International Conference on Machine Learning, pp. 147-153, 2003.

[3] Faber, V., Clustering and the Continuous k-Means Algorithm, Los Alamos Science number 22 138-144, 1994.

[4] Goulbourne, G., Coenen, F., Leng, P., Algorithms for computing association rules using a partial support tree, Journal of Knowledge Based Systems, 13:141--149, 2000.

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

[6] Hegland, M., Data Mining - Challenges, Models, Methods and Algorithms, May 2003, http://datamining.anu.edu.au/

[7] Huang, Z., Extensions to the k-means algorithm for clustering large data sets with categorical values. — Data Mining Knowl. Discov., Vol. 2, No. 2, pp. 283–304, 1998.

[8] John, G. H., Enhancements to the Data Mining Process, Stanford University, Ph.D. Thesis, 1997.

[9] Kanungo, T., Mount, D. M., Netanyahu, N., Piatko, C., Silverman, R., Wu, A. Y., An efficient k-means clustering algorithm: Analysis and implementation, IEEE Trans. Pattern Analysis and Machine Intelligence, 24, 881-892, 2002.

[10] Kovacs, E., Ignat, I., Clustering with Prototype Entity Selection in Data Mining, Proceedings of the International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, pages 415-419, 25-28 May 2006.

[11] Kovacs, E., Ignat, I., Clustering with Prototype Entity Selection with variable radius, Proceedings of the 2nd International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, pages 17-21, 1-2 September 2006.

[12] MacQueen, J., Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281-297, Berkeley, California.. 1967.

[13] Magidson, J., Vermunt, J.K., Latent class modeling as a probabilistic extension of K-means clustering, Quirk's Marketing Research Review, 77-80, March 2002.

[14] Magidson, J., Vermunt, J.K., Latent class models for clustering: A comparison with K-means, Canadian Journal of Marketing Research, 36-43, 2002.

[15] Ultsch, A., Clustering with SOM: U*C, In Proc. Workshop on Self-Organizing Maps, Paris, France, pp. 75-82, 2005.

[16] Vermeulen-Jourdan, L. , Dhaenens, C., Talbi, E., Clustering Nominal and Numerical Data: A New Distance Concept for a Hybrid Genetic Algorithm, Proceedings of the Fourth European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP), Coimbra, Portugal, LNCS, vol. 3004, p. 220–229, April 2004.