A NEW INCREMENTAL CORE-BASED CLUSTERING METHOD

Alina Câmpan and Gabriela Şerban
Department of Computer Science, "Babes-Bolyai" University, 1 M. Kogalniceanu Street, Cluj-Napoca, Romania

Abstract: Clustering is a data mining activity that aims to differentiate groups inside a given set of objects, with respect to a set of relevant attributes of the analyzed objects. Generally, existing clustering methods, such as *k-means* algorithm, start with a known set of objects, measured against a known set of attributes. But there are numerous applications where the attribute set characterizing the objects evolves. We propose an incremental, *k-means* based clustering method, *Core Based Incremental Clustering* (*CBIC*), that is capable to re-partition the objects set, when the attribute set increases. The method starts from the partitioning into clusters that was established by applying *k-means* or *CBIC* before the attribute set changed. The result is reached more efficiently than running *k-means* again from the scratch on the feature-extended object set. Experiments proving the method's efficiency are also reported.

Key words: data mining, clustering, k-means

1. INTRODUCTION

Unsupervised classification, or clustering, as it is more often referred as, is a data mining activity that aims to differentiate groups (classes or clusters) inside a given set of objects. The inferring process is carried out with respect to a set of relevant characteristics or attributes of the analyzed objects. The resulting groups are to be built so that objects within a cluster to have high similarity with each other and low similarity with objects in other groups. Similarity and dissimilarity between objects are calculated using metric or semi-metric functions, applied to the attribute values characterizing the objects. A large collection of clustering algorithms is available in the literature. [6] and [7] contain comprehensive overviews of existing techniques.

A well-known class of clustering methods is the one of the partitioning methods, with representatives such as the *k-means* algorithm or the *k-medoids* algorithm. Essentially, given a set of *n* objects and a number *k*, *k≤n*, such a method divides the object set into *k* distinct and non-empty partitions. The partitioning process is iterative and heuristic; it stops when a "good" partitioning is achieved. A partitioning is "good", as we said, when the intra-cluster similarities are high and inter-cluster similarities are low.

Generally, these methods start with a known set of objects, measured against a known set of attributes. But there are numerous applications where the object set is dynamic, or the attribute set characterizing the objects evolves. Obviously, for obtaining in these conditions a partitioning of the object set, the clustering algorithm can be applied over and over again, beginning from the scratch, every time the objects or attributes change. But this can be inefficient. What we want is to propose an incremental, *k-means* based clustering method, named *Core Based Incremental Clustering* (*CBIC*), that is capable to efficiently re-partition the object set, when the attribute set increases. The method starts from the partitioning into clusters that was established by applying *k-means* or *CBIC* before the attribute set changed. The result is reached more efficiently than running *k-means* again from the scratch on the feature-extended object set.

2. THEORETICAL MODEL

Let $\{O_1, O_2, \ldots, O_n\}$ be the set of objects to be classified. Each object is measured with respect to a set of *m* initial attributes and is therefore described by an *m*-dimensional vector $O_i = (O_{i1}, \ldots, O_{im}), O_{ik} \in \Re$, $1 \le i \le n,\ \ 1 \le k \le m$. Usually, the attributes associated to the objects are standardized, in order to ensure an equal weight to all of them ([7]).

The measure used for discriminating objects can be any *metric* function, *d*. We used the *Euclidian distance*:

$$d(O_i, O_j) = d_E(O_i, O_j) = \sqrt{\sum_{l=1}^{m} (O_{il} - O_{jl})^2} \ .$$

Let $\{K_1, K_2, \ldots, K_p\}$ be the set of clusters discovered in data by applying the *k-means* algorithm. Each cluster is a set of objects, $K_j = \{O_1^j, O_2^j, \ldots, O_{n_j}^j\},\ \ 1 \le j \le p$.

The centroid (cluster mean) of the cluster K_j is denoted by f_j, where

$$f_j = \left(\frac{\sum_{k=1}^{n_j} O_{k1}}{n_j}, \ldots, \frac{\sum_{k=1}^{n_j} O_{km}}{n_j} \right).$$

The measured set of attributes is afterwards extended with s ($s \geq 1$) new attributes, numbered as *(m+1), (m+2),..., (m+s)*. After extension, the objects' vectors become $O_i^{'} = (O_{i1}, \ldots, O_{im}, O_{i,m+1}, \ldots, O_{i,m+s})$, $1 \leq i \leq n$.

We want to analyze the problem of recalculating the objects' grouping into clusters, after attribute set extension and starting from the current partitioning. We aim to obtain a performance gain with respect to the partitioning from scratch process.

We start from the fact that, at the end of the initial clustering process, all objects are closer to the centroid of their cluster than to any other centroid. So, for any cluster j and any object $O_i^j \in K_j$, inequality (1) below holds.

$$d_E(O_i^j, f_j) \leq d_E(O_i^r, f_r), \forall j, r, 1 \leq j, r \leq p, r \neq j \qquad (1)$$

We denote by $K_j^{'}, 1 \leq j \leq p$ the set containing the same objects as K_j, after the extension. By $f_j^{'}, 1 \leq j \leq p$ we denote the mean (center) of the set $K_j^{'}$. These sets K_j, $1 \leq j \leq p$, will not necessarily represent clusters after the attribute set extension. The newly arrived attributes can change the objects arrangement into clusters, formed so that the intra-cluster similarity to be high and inter-cluster similarity to be low. But there is a considerable chance, when adding one or few attributes to objects, and the attributes have equal weights and normal data distribution, that the old arrangement in clusters to be close to the actual one. The actual clusters could be obtained by applying the *k-means* classification algorithm on the set of extended objects. But we try to avoid this process and replace it with one less expensive but not less accurate. With these being said, we agree, however, to continue to refer the sets $K_j^{'}$ as clusters.

We therefore take as starting point the previous partitioning into clusters and study in which conditions an extended object $O_i^{j\,'}$ is still correctly placed in its cluster $K_j^{'}$. For that, we express the distance of $O_i^{j\,'}$ to the center of its cluster, $f_j^{'}$, compared to the distance to the center $f_r^{'}$ of any other cluster $K_r^{'}$.

Theorem 1. When inequality (2) holds for an extended object $O_i^{j\,'}$ and its cluster $K_j^{'}$

$$O_{il} \geq \frac{\sum_{k=1}^{n_j} O_{kl}}{n_j}, \forall l \in \{m+1, m+2, \ldots, m+s\} \qquad (2)$$

then the object $O_i^{j\,'}$ is closer to the center $f_j^{'}$ than to any other center $f_r^{'}$, 1≤ j, r ≤ p, r ≠j.

Proof. We prove below this statement.

$$d^2(O_i^{j\,'}, f_j^{'}) - d^2(O_i^{j\,'}, f_r^{'}) = d^2(O_i^j, f_j) + \sum_{l=m+1}^{m+s} \left(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - O_{il} \right)^2 -$$

$$- d^2(O_i^j, f_r) - \sum_{l=m+1}^{m+s} \left(\frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} - O_{il} \right)^2$$

Using the inequality (1), we have:

$$d^2(O_i^{j\,'}, f_j^{'}) - d^2(O_i^{j\,'}, f_r^{'}) \leq \sum_{l=m+1}^{m+s} \left(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - O_{il} \right)^2 - \sum_{l=m+1}^{m+s} \left(\frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} - O_{il} \right)^2 \Leftrightarrow$$

$$d^2(O_i^{j\,'}, f_j^{'}) - d^2(O_i^{j\,'}, f_r^{'}) \leq \sum_{l=m+1}^{m+s} \left(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} \right) \cdot \left(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} + \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} - 2 \cdot O_{il} \right)$$

If the inequality (2) holds for every new attribute of $O_i^{j\,'}$, then the inequality above becomes:

$$d^2(O_i^{j'}, f_j^{'}) - d^2(O_i^{j'}, f_r^{'}) \le -\sum_{l=m+1}^{m+s} \left(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} \right)^2 \Leftrightarrow$$

$$d^2(O_i^{j'}, f_j^{'}) - d^2(O_i^{j'}, f_r^{'}) \le 0.$$

Because all distances are non-negative numbers, it follows that:

$$d(O_i^{j'}, f_j^{'}) \le d(O_i^{j'}, f_r^{'}) \, \forall r, 1 \le r \le p, r \ne j.$$

We have to notice that the inequality (2) imposes only intra-cluster conditions. An object is compared against its own cluster in order to decide its new affiliation to that cluster.

3. CORE BASED INCREMENTAL CLUSTERING

We will use the property enounced in the previous paragraph in order to identify inside each cluster $K_j^{'}, 1 \le j \le p$, those objects that have a considerable chance to remain stable in their cluster, and not to move in other cluster as a result of the attribute set extension. We will say that these objects form the *core* of their cluster.

Definition 1.

(a) We denote by $StrongCore_j = \{O_i^{j'} \mid O_i^{j'} \in K_j^{'}, O_i^{j'}$ satisfies inequalities set (2)$\}$ the set of all objects in $K_j^{'}$ satisfying inequality (2) for each new attribute l, $m+1 \le l \le m+s$;

(b) Let $sat(O_i^{j'})$ be the set of all new attributes l, $m+1 \le l \le m+s$ for which object $O_i^{j'}$ satisfies inequality (2). We denote by $WeakCore_j = \left\{ O_i^{j'} \mid O_i^{j'} \in K_j^{'}, \mid sat(O_i^{j'}) \mid \ge \frac{\sum_{k=1}^{n_j} \mid sat(O_k^{j'}) \mid}{n_j} \right\}$ the set of all objects in $K_j^{'}$ satisfying inequality (2) for at least so many new attributes that all objects in $K_j^{'}$ are satisfying (2) for, in the average.

(c) $Core_j$ = $StrongCore_j$ iif $StrongCore_j \neq \emptyset$; otherwise, $Core_j$ = $WeakCore_j$. $OCore_j = K_j^{'} - Core_j$ is the set of out-of-core objects in cluster $K_j^{'}$;

(d) We denote by *CORE* the set $\{Core_j,\ 1 \leq j \leq p\}$ of all clusters cores and by *OCORE* the set $\{OCore_j,\ 1\leq j \leq p\}$.

For each new attribute l, $m+1 \leq l \leq m+s$, and each cluster $K_j^{'}$ there is at least one object that satisfies the inequality (2) in respect to the attribute l. Namely, the object that has the greatest value for attribute l between all objects in $K_j^{'}$ certainly satisfies the relation (the maximum value in a set is greater or equal than the mean of the values in the set). But it is not sure that there is in cluster $K_j^{'}$ any object that satisfies relation (2) for all new attributes $m+1,\ldots,m+s$. If there are such objects ($StrongCore_j \neq \emptyset$), we know that, according to **Theorem 1**, they are closer to the cluster center $f_j^{'}$ than to any other cluster center $f_r^{'}$, $1\leq r \leq p,\ r\neq j$. Then, $Core_j$ will be initialized with $StrongCore_j$ and will be the seed for cluster j in the incremental algorithm. But if $StrongCore_j = \emptyset$, than we will choose as seed for cluster j other objects, the most stable ones between all objects in $K_j^{'}$. These objects ($WeakCore_j$) can be less stable than would be the objects in $StrongCore_j$. This is not, however, a certain fact: the objects in the "weaker" set $WeakCore_j$ can be as good as those in $StrongCore_j$. This comes from the fact that **Theorem 1** enounces a *sufficient* condition for the objects in $K_j^{'}$ to be closer to $f_j^{'}$ than to any other $f_r^{'}$, but not a *necessary* condition too.

The *cluster cores*, chosen as we described, will serve as seed in the incremental clustering process. All objects in $Core_j$ will surely remain together in the same group if clusters do not change. This will not be the case for all core objects, but for most of them, as we will see in the results section.

We give next the *Core Based Incremental Clustering* algorithm. We mention that the algorithm stops when the clusters from two consecutive iterations remain unchanged or the number of steps performed exceeds the maximum allowed number of iterations.

```
Algorithm Core Based Incremental Clustering is
Input: - the set $X = \{O_1,\ldots,O_n\}$ of m-dimensional objects,
       - the set $X^{'} = \{O_1^{'},\ldots,O_n^{'}\}$ of (m+s)-dimensional extended objects
         to be clustered, $O_i^{'}$ has the same first m components as $O_i$,
       - the metric $d_E$ between objects in a multi-dimensional space,
       - $p$, the number of desired clusters,
       - $K = \{K_1,\ldots,K_p\}$ the previous partitioning of objects in $X$,
       - noMaxIter the maximum number of iterations allowed.
```

```
Output:- the re-partitioning $K' = \{K'_1, \ldots, K'_p\}$ for the objects in $X'$.
Begin
    For all clusters $K_j \in K$ do
        Calculate $Core_j = (StrongCore_j \neq \emptyset)\ ?\ StrongCore_j : WeakCore_j$
        $K'_j = Core_j$
        Calculate $f'_j$ as the mean of objects in $Core_j$
    EndFor
    While ($K'_j$ changes between two consecutive steps) and
            (there were not performed noMaxIter iterations) do
        For all clusters $K'_j$ do
            $K'_j = \{O'_i \mid d(O'_i, f'_j) \leq d(O'_i, f'_r), \forall r, 1 \leq r \leq p, 1 \leq i \leq n\}$
        EndFor
        For all clusters $K'_j$ do
            $f'_j =$ the mean of objects in $K'_j$
        EndFor
    EndWhile
End.
```

The algorithm starts by calculating the old clusters' cores. The cores will be the initial clusters for the iterative process. Next, the algorithm proceeds in the same manner as the classical *k-means* method does. We mention that the computation of the core of a cluster *C* depends only on the current cluster (does not depend on other clusters).

4. EXPERIMENTAL EVALUATION

In this section we present some experimental results obtained by applying the *CBIC* algorithm described in section 3.

For this purpose, we used a programming interface for non-hierarchical clustering described in ([1]). We have to mention that using this interface we can simply develop non-hierarchical clustering applications for different kind of data (objects to be clustered). As it is shown in our experiments, the objects to be clustered are very different (patients, wine instances).

As a case study, for experimenting our theoretical results described in section 2 and for evaluating the performance of the *CBIC* algorithm, we consider some experiments that are briefly described in the following subsections. We have to mention that all data were taken from the website at "http://www.cormactech.com/neunet".

4.1 Quality Measures

As a quality measure we take the movement degree of the core objects and of the extra-core objects. In other words, we measure how the objects in either $Core_j \in CORE$, or $OCore_j \in OCORE$, remain together in clusters after the algorithm ends.

As expected, more stable the core objects are and more they remain together with respect to the initial sets $Core_j$, better was the decision to choose them as seed for the incremental clustering process. Also, as the experiments will show, the movement degree was always smaller for the core objects than for the extra-core objects.

We express the *core stability factor* as:

$$CSF(CORE) = \frac{\sum_{j=1}^{p} \frac{|Core_j|}{no\ of\ clusters\ where\ the\ objects\ in\ Core_j\ ended}}{\sum_{j=1}^{p} |Core_j|} \tag{4}$$

The worst case is when each object in $Core_j$ ends in a different final cluster, and this happens for every core in $CORE$. The best case is when every $Core_j$ remains compact and it is found in a single final cluster. So, the limits between which CSF varies are given below, where the higher the value of CSF is, better was the cores choice:

$$\frac{p}{\sum_{j=1}^{p} |Core_j|} \leq CSF(CORE) \leq 1 \tag{5}$$

Accordingly, the *out-of-core stability factor*, *OCSF(OCORE)*, is defined similar to *CSF(CORE)*, replacing the sets $Core_j$ with $OCore_j$.

For comparing the informational relevance of the attributes we used the *information gain* (*IG*) measure ([9]).

4.2 Experiment 1. Cancer

The objects to be clustered in this experiment are patients: each patient is identified by 9 attributes [3]. The attributes have been used to represent instances. Each instance has one of 2 possible classes: benign or malignant. In this experiment there are 457 patients (objects).

4.3 Experiment 2. Dermatology

The objects to be clustered in this experiment are also patients: each patient is identified by 34 attributes, 33 of which are linear valued and one of them is nominal. There are 366 objects (patients).

The aim of the clustering process is to determine the type of Eryhemato-Squamous Disease [4].

4.4 Experiment 3. Wine

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines [5].

The objects to be clustered in this experiment are wine instances: each is identified by 13 attributes. There are 178 objects (wine instances).

4.5 Results

In this section we present comparatively the results obtained after applying the *CBIC* algorithm for the experiments described in the above subsections. We mention that the results are calculated in average, for six executions.

Table 1. Comparative results

Experiment	Cancer	Dermatology	Wine
No of objects	457	366	178
No of attributes (m+s)	9	34	13
No of new attributes (s)	4	3	4
No of k-means iterations for (m+s) attributes	6.2857	11.57	7
No of k-means iterations for m attributes	6	11.85	9.37
No of CBIC iterations for (m+s) attributes	7	6.14	6.8
CSF(CORE)	0.9	0.8406	0.5244
OCSF(OCORE)	0.5	0.7008	0.4194

From Table 1 we observe that using the *CBIC* algorithm the number of iterations for finding the solution is, in the average, smaller, and also the cores' stability factor, *CSF(CORE)*, is high. We mention that for every running of each experiment, *CSF(CORE)≥ OCSF(OCORE)*. So, every time, the stability of the objects chosen to be part or cores was greater than the stability of out-of-core objects.

In Table 2 we present, for each experiment, the attributes in decreasing order of their information gain (*IG*).

Table 2. The decreasing order of attributes in respect to the information gain measure

Experiment	Order of attributes	*IG* of new attributes / *IG* of old attributes (%)
Cancer	2 3 **6 7** 5 4 **8** 1 **9**	64,7%
Dermatology	22 21 23 1 **34** 30 28 13 26 7 17 9 29 10 16 11 25 15 6 27 4 20 **32** 8 5 24 3 31 12 2 19 18 14 **33**	7,6%
Wine	7 **10 12 13** 6 1 2 **11** 9 4 5 3 8	57%

From Table 2 it results that the importance of the added attributes influences the number of iterations performed by the *CBIC* algorithm for finding the solution. For example, in the "cancer" experiment where the information brought by the added attributes was close to that of the initial ones, the number of iterations performed by *CBIC* is also close to the number of iterations performed for all the attributes.

5. CONCLUSIONS AND FUTURE WORK

Further works can be done in the following directions: to apply the incremental algorithm on precise problems, from where the need for such an incremental algorithm originated; to study how the theoretical results described for non-hierarchical clustering could be applied/generalized for other clustering techniques.

REFERENCES

[1] Serban, G.: A Programming Interface for Non-Hierarchical Clustering, Studia Universitatis "Babes-Bolyai", Informatica, XLX(1), 2005, to appear.

[2] Serban, G., Campan, A.: Core Based Incremental Clustering, Studia Universitatis "Babes Bolyai", Informatica, XLXI(2), 2005, to appear.

[3] Wolberg, W., Mangasarian, O.L.: "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990: 9193-9196.

[4] Demiroz, G., Govenir, H. A., Ilter, N.: "Learning Differential Diagnosis of Eryhemato Squamous Diseases using Voting Feature Intervals", Artificial Intelligence in Medicine.

[5] Aeberhard, S., Coomans, D., de Vel, O.: "THE CLASSIFICATION PERFORMANCE OF RDA" Tech. Rep. no. 92--01, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.

[6] Jain, A., Dubes, R, "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs, New Jersey, 1998.

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

[8] http://www.cormactech.com/neunet, "Discover the Patterns in Your Data", CorMac Technologies Inc, Canada.

[9] Quinlan, J. R., "C4.5: Programs for Machine Learning, Morgan Kaufmann", 1993.