STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **L**, Number 2, 2005

ADAPTIVE CLUSTERING USING A CORE-BASED APPROACH

GABRIELA ŞERBAN AND ALINA CÂMPAN

Abstract. This paper studies an adaptive clustering problem. We focus on re-clustering an object set, previously clustered, when the feature set characterizing the objects increases. We propose an adaptive, *k-means* based clustering method, *Core Based Adaptive k-means (CBAk)*, that adjusts the partitioning into clusters that was established by applying *k-means* or *CBAk* before the feature set changed. We aim to reach the result more efficiently than running *k-means* starting from the current clustering. Experiments testing the method's efficiency are also reported.

Keywords: Data Mining, clustering, k-means.

1. Introduction

A large collection of clustering algorithms is available in the literature. The papers [5], [6] and [7] contain comprehensive overviews of the existing clustering techniques.

A well-known class of clustering methods is the one of the partitioning by relocation methods, with representatives such as the *k-means* algorithm or the *k-medoids* algorithm. Essentially, given a set of n objects and a number $k, k \leq n$, such a method divides the object set into k distinct clusters. The partitioning process is iterative and stops when a "good" partitioning is achieved. Finding a "good" partitioning coincides with optimizing a criterion function. The criterion function used in *k-means* is the squared error criterion, which tends to work well with isolated and compact clusters [7].

Generally, these methods apply on a set of objects measured against a known set of features (attributes). But there are applications where the attribute set characterizing the objects evolves. For obtaining in these conditions a partitioning of the object set, the clustering algorithm can be, obviously, applied over and over again, beginning from scratch or from the current partitioning, each time

Received by the editors: October 15, 2005.

2000 *Mathematics Subject Classification.* 62H30, 68U35.

1998 *CR Categories and Descriptors.* 62H30 [**Statistics**]: Multivariate analysis – *Classification and discrimination; cluster analysis*; 68U35 [**Computer science**]: Computing methodologies and applications – *Information systems (hypertext navigation, interfaces, decision support, etc.)*;

when the attributes change. But this can be inefficient. What we want is to propose an adaptive, *k-means* like clustering method, named *Core Based Adaptive k-means (CBAk)*, 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 *CBAk* before the attribute set changed. We aim to reach the result more efficiently than running *k-means* starting from the current clustering.

Related Work

There are few approaches reported in the literature that address the problem of adapting the result of a clustering when the object feature set is extended. Early works treat the sequential use of features in the clustering process, one by one. An example of such a monothetic approach is mentioned in [7]. A more recent paper [10] analyzes the same problem of adapting a clustering produced by a *DBSCAN* like algorithm, using some additional structures and distance approximations in an Euclidian space. However, adapting a clustering resulted from a partitioning algorithm, using partitioning-based methods hasn't been reported by none of these works.

2. Theoretical Model

Let $X = \{O_1, O_2, \ldots, O_n\}$ 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 $O_i = (O_{i1}, \ldots, O_{im}), O_{ik} \in \Re^+, 1 \leq i \leq n, 1 \leq k \leq m$. Usually, the attributes associated to objects are standardized, in order to ensure an equal weight to all of them [5].

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, $K_j = \{O_1^j, O_2^j, \ldots, O_{n_j}^j\}, 1 \leq j \leq p$. The centroid (cluster mean) of the cluster K_j is denoted by f_j, where

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

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

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

The measured set of attributes is afterwards extended with s ($s \geq 1$) new attributes, numbered as $(m+1), (m+2), \ldots, (m+s)$. 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 denote by $extO_i' = (O_{i,m+1}, \ldots, O_{i,m+s})$ the s-attribute extension of the vector associated to O_i.

We want to analyze the problem of recalculating the objects' grouping into clusters, after object extension and starting from the current partitioning. We start from the fact that, at the end of the initial *k-means* 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.

$$d_E(O_i^j, f_j) \leq d_E(O_i^j, f_r), \forall j, r,\ 1 \leq j, r \leq p,\ r \neq j. \tag{1}$$

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 of K'_j. We denote by $extf'_j = \left(\frac{\sum_{k=1}^{n_j} O^j_{k,m+1}}{n_j}, \ldots, \frac{\sum_{k=1}^{n_j} O^j_{k,m+s}}{n_j} \right)$ the s-attribute extension of the K'_j center (mean). 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. But there is a considerable chance, when adding one or few attributes to objects, that the old arrangement in clusters to be close to the actual one. The actual clusters can be obtained by applying the *k-means* algorithm on the set of extended objects starting from the current clustering. 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 $O_i^{j\prime}$ is still "correctly" placed into its cluster K'_j. For that, we express the distance of $O_i^{j\prime}$ to the center of its cluster, f'_j, compared to the distance to the center f'_r of any other cluster K'_r.

Lemma 1. *When inequality (2) holds for an extended object $O_i^{j\prime} \in K'_j$*

$$d^2(extO_i^{j\prime}, extf'_j) \leq d^2(extO_i^{j\prime}, extf'_r) \tag{2}$$

for all $r = \overline{1,p},\ r \neq j$ then the object $O_i^{j\prime}$ is closer to the center f'_j than to any other center $f'_r, 1 \leq j,\ r \leq p,\ r \neq j$.

Proof

We prove this statement. For $O_i^{j\prime}$ and $1 \leq r \leq p$

$d^2(O_i^{j\prime}, f'_j) - d^2(O_i^{j\prime}, f'_r) = d^2(O_i^j, f_j) + d^2(extO_i^{j\prime}, extf'_j) - d^2(O_i^j, f_r) - d^2(extO_i^{j\prime}, extf'_r)$.

Using the inequality (1), we have:

$d^2(O_i^{j\prime}, f'_j) - d^2(O_i^{j\prime}, f'_r) \leq d^2(extO_i^{j\prime}, extf'_j) - d^2(extO_i^{j\prime}, extf'_r)$.

If the inequality (2) holds, then the inequality above becomes:

$d^2(O_i^{j\prime}, f'_j) - d^2(O_i^{j\prime}, f'_r) \leq 0$.

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

$d(O_i^{j\prime}, f_j') \leq (O_i^{j\prime}, f_r'), \forall r,\ 1 \leq r \leq p,\ r \neq j.$

Remark The global complexity of the *CBAk* algorithm is not increased by the cluster cores calculation.

3. The *Core Based Adaptive k-means* Algorithm

We will use the property enounced in the previous paragraph in order to identify inside each cluster $K_j', 1 \leq j \leq p$, the objects that have a considerable chance to remain stable in their cluster, and not to move into another cluster as a result of the attribute set extension. These objects form the *core* of their cluster.

Definition 1.

a) We denote by $StrongCore_j = \{O_i^{j\prime} | O_i^{j\prime} \in K_j', O_i^{j\prime}$ satisfies the inequality (2)$\}$, $\forall r,\ 1 \leq r \leq p,\ r \neq j$.

b) Let $sat(O_i^{j\prime})$ be the set of all clusters $K_r', \forall r,\ 1 \leq r \leq p,\ r \neq j$ not containing $O_i^{j\prime}$ and for which object $O_i^{j\prime}$ satisfies inequality (2).
We denote by $WeakCore_j = \{O_i^{j\prime} | O_i^{j\prime} \in K_j', |sat(O_i^{j\prime})| \geq \frac{\sum_{k=1}^{n_j} |sat(O_k^{j\prime})|}{n_j}\}$ the set of all objects in K_j' satisfying inequality (2) for at least so many clusters that all objects in K_j' are satisfying (2), in the average.

c) $Core_j = StrongCore_j$ iif $StrongCore_j \neq \emptyset$; otherwise,
$Core_j = WeakCore_j$. $OCore_j = K_j' \setminus Core_j$ 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 cluster cores and by $OCORE$ the set $\{OCore_j, 1 \leq j \leq p\}$.

We have chosen the above cluster cores definition because of the following reasons. It is not sure that there is in cluster K_j' any object that satisfies inequality (2) for all clusters $K_r', 1 \leq r \leq p,\ r \neq j$. If there are such objects ($StrongCore_j \neq \emptyset$), we know that, according to Lemma 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 taken to be equal to $StrongCore_j$ and will be the seed for cluster j in the adaptive algorithm. But if $StrongCore_j = \emptyset$, for the core not to be empty, we will choose as seed for cluster j other objects, the most stable ones between all objects in K_j'.

The cluster cores, chosen as we described, will serve as seed in the adaptive 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 Adaptive k-means* 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 Adaptive k-means is
Input: - the set X = {O_1, ..., O_n} of m-dimensional previously clustered
         objects,
       - the set X' = {O'_1, ..., 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, ..., K_p} the previous partition of objects in X,
       - noMaxIter the maximum number of iterations allowed.
Output: - the new partition K' = {K'_1, ..., K'_p} for the objects in X'.
Begin
    For all clusters K_j ∈ K
       Calculate Core_j = (StrongCore_j ≠ ∅)?StrongCore_j : WeakCore_j
       K'_j = Core_j
       Calculate f'_j as the mean of objects in K'_j
    EndFor
    While (K' changes between two consecutive steps) and
          (there were not performed noMaxIter iterations) do
       For all clusters K'_j do
          K'_j = {O'_i | d(O'_i, f'_j) ≤ d(O'_i, f'_r), ∀r, 1 ≤ r ≤ p, 1 ≤ i ≤ 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 new initial clusters from which the iterative processing begins. Next, the algorithm proceeds in the same manner as the classical *k-means* method does.

4. Experimental Evaluation

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

As case studies, for experimenting our theoretical study described in section 2 and for evaluating the performance of the *CBAk* algorithm, we considered the data

sets described in [1]. The data were taken from the website "http://www.cormactech.com/neunet" and have also been used in [2, 4, 9].

4.1. **Quality Measures.** As a quality measure for our algorithm 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 in respect to the initial sets $Core_j$, better was the decision to choose them as seed for the adaptive clustering process.

We denote by $S = \{S_1, S_2, \ldots, S_p\}, S_i \subseteq K_i$, a set of clusters' subsets (as $CORE$ and $OCORE$ are). We express the *stability factor* of S as:

$$SF(S) = \frac{\sum_{j=1}^{p} \frac{|S_j|}{\text{no of clusters where the objects in } S_j \text{ ended}}}{\sum_{j=1}^{p} |S_j|} \tag{3}$$

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

$$\frac{p}{\sum_{j=1}^{p} |Core_j|} \leq SF(CORE) \leq 1 \tag{4}$$

For comparing the quality of the partitions produced by our algorithm and by *k-means*, we consider the *squared sum error (SSE)* of a clustering K, defined as:

$$SSE(K) = \sum_{K_j \in K} \sum_{O_i \in K_j} d^2(O_i, f_j) \tag{5}$$

When comparing two partitions K_1 and K_2 for the same data set, we will say that K_1 is better than K_2 iff $SSE(K_1) < SSE(K_2)$.

For measuring the clustering tendency of a data set, we use the Hopkins statistics, H [11], an approach that uses statistical tests for spatial randomness. H takes values between 0 and 1, and a value near 1 indicates that data is highly clustered. Usually, for a data set with clustering tendency, we expect for H values greater than 0.5.

4.2. **Results.** In this section we comparatively present the results obtained by applying the *CBAk* algorithm and *k-means*, for the experimental data. We mention that the results are calculated in average, for several executions.

TABLE 1. The 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 clusters	2	6	3
No of k-means iterations for m attributes	5.66	11.2	9.28
No of k-means iterations for +s attributes	4	1.33	3.85
No of CBAk iterations for +s attributes	4	5.66	2.42
k-means SSE for +s attributes	13808.784	12683.82	49.016
CBAk SSE for +s attributes	13808.784	12522.95	49.019
SF(CORE)	1.0	0.8119	0.97
SF(OCORE)	0.5	0.646	0.475
H for s attributes	0.666	0.68122	0.7018
H for m+s attributes	0.7148	0.6865	0.7094

From Table 1 we observe that using the *CBAk* algorithm the number of iterations for finding the solution is not always smaller that in case of using *k-means*; but the cores' stability factor, $SF(CORE)$, is high. We mention that for every run of each experiment, SSE(CBAk) has been roughly equal to SSE(k-means). Also, every time, the stability of the objects chosen to be part or cores was greater than the stability of out-of-core objects.

5. Conclusions and Future Work

In this paper we proposed a new method for adapting the result of a clustering when the attribute set describing the objects increases. The experiments on different data sets prove that, in most cases, the result is reached more efficiently using the proposed method than running *k-means* starting from the current partition, on the feature-extended object set. But there are some situations when it is better to resort to a *k-means* clustering of the feature-extended object set, starting from the existing clustering, than using the *CBAk* algorithm. For example, such situations can be: the addition of a large number of features or the addition of new features with large information gain and contradictory information with respect to the old feature set.

Further work may be done in the following directions:

- to isolate conditions to decide when it is more effective to adapt (using *CBAk*) the result of a clustering of the feature-extended object set than to resume its clustering using *k-means*;
- to study how the information brought into the system by the newly added attributes, their correlation with the initial ones, influences the number of iterations performed by the *CBAk* algorithm for finding the solution;
- to apply the adaptive algorithm on precise problems, from where the need of such an adaptive algorithm originated;
- to study how the theoretical results described for non-hierarchical clustering could be applied/generalized for other clustering techniques.

References

[1] Şerban, G., Câmpan, A.: "Core Based Incremental Clustering", Studia Universitatis "Babeş-Bolyai", Informatica, L(1), 2005, pp 89–96
[2] Aeberhard, S., Coomans, D., de Vel, O.: "THE CLASSIFICATION PERFORMANCE OF RDA", Tech. Rep. 92–01, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992
[3] CorMac Technologies Inc, Canada: "Discover the Patterns in Your Data", http://www.cormactech.com/neunet
[4] Demiroz, G., Govenir, H. A., Ilter, N.: "Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals", Artificial Intelligence in Medicine
[5] Han, J., Kamber, M.: "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, 2001
[6] Jain, A., Dubes, R.: "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs, New Jersey, 1998
[7] Jain, A., Murty, M. N., Flynn, P.: "Data clustering: A review", ACM Computing Surveys, 31(3), 1999, pp 264-323
[8] Quinlan, J. R.: C4.5: "Programs for Machine Learning", Morgan Kaufmann. San Mateo, California, 1993
[9] 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, pp 9193–9196
[10] Wu, F., Gardarin, G.: "Gradual Clustering Algorithms", Proceedings of the 7th International Conference on Database Systems for Advanced Applications (DASFAA'01), 2001, pp 48–57
[11] Tan, P.-N., Steinbach, M., Kumar, V.: "Introduction to Data Mining", Addison Wesley, 2005, chapters 8,9

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania
E-mail address: gabis@cs.ubbcluj.ro

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania
E-mail address: alina@cs.ubbcluj.ro