INFORMATICA, 2008, Vol. 19, No. 1, 101–112
© 2008 *Institute of Mathematics and Informatics, Vilnius*

# Hierarchical Adaptive Clustering

Gabriela ŞERBAN, Alina CÂMPAN
*Department of Computer Science, "Babeş-Bolyai" University*
*1, M. Kogalniceanu Street, Cluj-Napoca, Romania*
*e-mail: {gabis, alina}@cs.ubbcluj.ro*

Received: January 2006

**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 clustering method based on a hierarchical agglomerative approach, Hierarchical Adaptive Clustering (*HAC*), that adjusts the partitioning into clusters that was established by applying the hierarchical agglomerative clustering algorithm (*HACA*) (Han and Kamber, 2001) before the feature set changed. We aim to reach the result more efficiently than running *HACA* again from scratch on the feature-extended object set. Experiments testing the method's efficiency and a practical distributed systems problem in which the *HAC* method can be efficiently used (the problem of adaptive horizontal fragmentation in object oriented databases) are also reported.

**Key words:** data mining, hierarchical agglomerative clustering, adaptive clustering.

## 1. Introduction

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. A large collection of clustering algorithms is available in the literature. The papers (Han and Kamber, 2001) and (Jain and Dubes, 1998) contain comprehensive overviews of existing techniques. All clustering techniques employ a dissimilarity (similarity) measure between objects, usually expressed on the basis of a *distance function*. Generally, the distance function is defined on the attribute (feature) set describing the objects.

Hierarchical clustering methods represent a major class of clustering techniques. There are two types of hierarchical clustering algorithms. Given a set of $n$ objects and a number $k$, $k \leqslant n$, the agglomerative (bottom-up) methods begin with $n$ singletons (sets with one element), merging them until the desired number of clusters ($k$) is achieved. At each step, the most similar two clusters are chosen for merging. The divisive (top-down) methods start from one cluster containing all $n$ objects and split it until the desired number of clusters ($k$) is achieved. In the following, we will name *HACA* the classical hierarchical agglomerative clustering algorithm.

Generally, the clustering 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, each time when the attributes change. But this can be inefficient.

We propose an adaptive clustering algorithm, named *Hierarchical Adaptive Clustering* (*HAC*), based on detecting stable structures (called cores) inside the existing clusters, when the attribute set increases. We aim to reach the result more efficiently than applying the hierarchical agglomerative clustering algorithm (*HACA*) (Han and Kamber, 2001) again from the scratch on the feature-extended object set.

**Related Work**. There are few approaches reported in the literature that take into account 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 (Jain *et al.*, 1999). A more recent paper (Wu and Gardarin, 2001) 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 hierarchical agglomerative algorithm has not been reported by none of these works.

The term *adaptive clustering* has also been used in (Bagherjeiran *et al.*, 2005), but with a different meaning. In this paper the authors address, under the name of *adaptive clustering*, the problem of adapting the distance function for discriminating the objects, using a reinforcement learning approach. Besides the terminology coincidence, this problem has nothing in common with the one we approach in this paper.

The remaining of the paper is organized as follows. Section 2 presents the theoretical modeling of the adaptive clustering problem. The *Hierarchical Adaptive Clustering algorithm* (*HAC*) is described in Section 3. Section 4 presents a comparative experimental evaluation of *HAC* and *HACA* algorithms, with respect to several cost measures we describe. Section 4 also indicates how a practical distributed systems problem can be efficiently approached, modeled and solved using the proposed adaptive clustering method. Conclusions and future work are given in Section 5.

## 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 a $n$-dimensional vector $O_i = (O_{i1}, \ldots, O_{im})$, $O_{ik} \in \Re^+$, $1 \leqslant i \leqslant n$, $1 \leqslant k \leqslant m$. Usually, the attributes associated to objects are standardized, in order to ensure an equal weight to all of them (Han and Kamber, 2001).

Let $\{K_1, K_2, \ldots, K_p\}$ be the set of clusters discovered in data by applying the *HACA* algorithm. Each cluster is a set of objects, $K_j = \{O_1^j, O_2^j, \ldots, O_{n_j}^j\}$, $1 \leqslant j \leqslant p$. Even if it is a concept appeared from and employed in non-hierarchical clustering (*k-means*), we will also make use in our method of the centroid notion. The centroid of the cluster $K_j$, denoted by $f_j$, represents the cluster's mean and is defined as

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

The measure used for discriminating objects can be any *metric* or *semi-metric* function, $d$. We use in our approach the *Euclidian distance* as distance function between the objects:

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

The measured set of attributes is afterwards extended with $s$ $(s \geqslant 1)$ new attributes, numbered as $(m+1), (m+2), \ldots, (m+s)$. After extension, the objects' feature vectors become $O'_i = (O_{i1}, \ldots, O_{im}, O_{i,m+1}, \ldots, O_{i,m+s})$, $1 \leqslant i \leqslant n$.

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

We denote by $K'_j$, $1 \leqslant j \leqslant p$ , the set containing the same objects as $K_j$, after the attribute set extension. By $f'_j$, $1 \leqslant j \leqslant p$, we denote the mean (center) of the set $K'_j$.

These sets $K'_j$, $1 \leqslant j \leqslant p$, 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, and the attributes have equal weights and normal data distribution, that the old arrangement into clusters to be close to the new actual one. The actual $K'_j$ clusters could be of course obtained by applying the *HACA* clustering 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 $O_i^{j\prime}$ is still "correctly" placed in its cluster $K'_j$. For that, we express the distances between $O_i^{j\prime}$ and the centers of its old and new clusters, $f_j$ and $f'_j$, compared to the distances to the centers $f_r$ and $f'_r$ of any other cluster $K'_r$, $1 \leqslant r \leqslant p,\ r \neq j$. The objects in cluster $j$ that are satisfying certain conditions are similar enough to be kept together. So we keep them in one cluster. The remaining objects in cluster $j$ (those that are not satisfying the imposed conditions) will be extracted and distributed each one in its singleton. Clearly, from this cluster adjustment process will result a number $k'$ of clusters, $k \leqslant k' \leqslant n$. In order to reach again the targeted number $k$ of clusters, we proceed next to merge clusters in the same manner as *HACA* does. But, as we do not generally start again from singletons, the number of steps will be significantly reduced. Also, as we will demonstrate by experiments, we do not lose significantly the quality of the clusters obtained by the hierarchical adaptive clustering method described above compared to the quality of the clusters provided by *HACA*. We mention that the group metric ("linkage metric") that we have used in our experiments is "average-link".

In Theorem 1 below we give conditions in which an extended object $O_i^{j\prime}$ is "correctly" placed in its cluster $K_j'$, assuming that $K_j$ is the cluster in which $O_i^j$ was disposed by *HACA*.

**Theorem 1.** *When inequality* (1) *holds for an object* $O_i^j$ *and its cluster* $K_j$

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

*and inequality* (2) *holds for the extended object* $O_i^{j\prime}$ *and its cluster* $K_j'$

$$\sum_{l=m+1}^{m+s} \Big(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r}\Big) \cdot \Big(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} + \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} - 2 \cdot O_{il}\Big) \leqslant 0 \tag{2}$$

*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* $f_r'$, $1 \leqslant j$, $r \leqslant p$, $r \neq j$.

*Proof.* We prove below this statement.

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

Using the inequality (1), we have:

$$\begin{aligned}
&d^2(O_i^{j\prime}, f_j') - d^2(O_i^{j\prime}, f_r') \\
&\quad \leqslant \sum_{l=m+1}^{m+s} \Big(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - O_{il}\Big)^2 - \sum_{l=m+1}^{m+s} \Big(\frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} - O_{il}\Big)^2 \Leftrightarrow \\
&d^2(O_i^{j\prime}, f_j') - d^2(O_i^{j\prime}, f_r') \\
&\quad \leqslant \sum_{l=m+1}^{m+s} \Big(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} - \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r}\Big) \cdot \Big(\frac{\sum_{k=1}^{n_j} O_{kl}}{n_j} + \frac{\sum_{k=1}^{n_r} O_{kl}}{n_r} - 2 \cdot O_{il}\Big).
\end{aligned}$$

If the inequality (2) holds for $O_i^{j\prime}$, then the inequality above becomes

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

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

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

The first condition in Theorem (1) requires that an object $O_i \in K_j$ to be, at the end of the initial clustering process, closer to the centroid of its cluster than to any other centroid. Inequality (1) will not hold for every object with respect to the clusters produced by *HACA*. But as we used as linkage-metric in *HACA* average-link, it is likely that a lot of objects will satisfy inequality (1). All objects $O_i \in K_j$ which satisfy inequality (1) and have extensions that meet the requirements expressed by condition (2), are sufficiently similar with each other and dissimilar to the objects in other clusters. So, it is justified to keep them together in one cluster, after the attribute set extension.

## 3. The *Hierarchical Adaptive Clustering* Algorithm

We will use the properties enounced in Section 2 in order to identify inside each cluster $K'_j$, $1 \leqslant j \leqslant p$, those objects that have a considerable chance to remain together in one cluster, and not to move in different clusters as a result of the attribute set extension. We say that these objects form the *core* of their cluster.

DEFINITION 1. We introduce the following terms that will be used in our adaptive algorithm:

a) We denote by $StrongCore_j = \{O_i^{j\prime} | O_i^{j\prime} \in K'_j,\ O_i^j$ satisfies inequality (1) and $O_i^{j\prime}$ satisfies inequalities set (2)$\}$ - the set of all objects in $K'_j$ closer, before extension, to the center of their cluster than to the center of any other cluster and satisfying, after extension, inequality (2) for each cluster $K'_r$ $1 \leqslant j,\ r \leqslant p,\ r \neq j$.

b) Let $sat(O_i^{j\prime})$ be the set of all clusters $K'_r, \forall r,\ 1 \leqslant r \leqslant p,\ r \neq j$ not containing $O_i^{j\prime}$ and for which object $O_i^{j\prime}$ satisfies inequality (2). We denote by $WeakCore_j = \{O_i^{j\prime} | O_i^{j\prime} \in K'_j, O_i^j$ satisfies inequality (1) and $|sat(O_i^{j\prime})| \geqslant \frac{\sum_{k=1}^{n_j} |sat(O_k^{j\prime})|}{n_j}\}$ the set of all objects in $K'_j$ satisfying: before extension, inequality (1); after extension, inequality (2) for at least so many clusters that all objects in $K'_j$ are satisfying (2), in 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 $K'_j$.

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 $K'_r$, $1 \leqslant r \leqslant p$, $r \neq j$. 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 $f'_r$, $1 \leqslant r \leqslant p$, $r \neq j$. 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 $StrongCore_j = \emptyset$, then, 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 $K'_j$.

We give next the *Hierarchical Adaptive Clustering* (*HAC*) algorithm.

The algorithm starts by calculating the old clusters' cores. The cores will be the new initial clusters from which the iterative process begins. Next, the algorithm proceeds in the same manner as the classical *HACA* method does.

We mention that the algorithm stops when the desired number of clusters is obtained.

```
Algorithm Hierarchical Adaptive Clustering is
Input:
    - the set $X = \{O_1, \ldots, O_n\}$ of $m$-dimensional previously clustered
      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;
    - the number p of desired clusters;
    - $K = \{K_1, \ldots, K_p\}$ the previous partition of objects in $X$.
Output:
    - the new partition $K' = \{K'_1, \ldots, K'_p\}$ for the objects in $X'$.
Begin
    For all clusters $K_j \in K$ do
        Calculate $Core_j \leftarrow (StrongCore_j \neq \emptyset)?StrongCore_j : WeakCore_j$
        Calculate $OCore_j \leftarrow K_j \setminus Core_j$
    EndFor
    $C \leftarrow \emptyset$ // the current cluster set
    For i = 1 to p do
        If $Core_i \neq \emptyset$ then
            $C \leftarrow C \cup \{Core_i\}$
        EndIf
        For all $O \in OCore_i$ do
            $C \leftarrow C \cup \{O\}$ //add a singleton to $C$
        EndFor
    EndFor
    While $\mid C \mid > p$ do
        $(C_{u^*}, C_{v^*}) \leftarrow argmin_{(C_u, C_v)} d_E(C_u, C_v)$
        $C_{new} \leftarrow C_{u^*} \cup C_{v^*}$
        $C \leftarrow C \setminus \{C_{u^*}, C_{v^*}\} \cup \{C_{new}\}$
    EndWhile
    $K' \leftarrow C$
End.
```

As we have already mentioned, we consider the distance $d_E(C_u, C_v)$ between two clusters $C_u$ and $C_v$ expressed by the average-link metric:

$$d_E(C_u, C_v) = \frac{\sum_{a \in C_u} \sum_{b \in C_v} d_E(a, b)}{\mid C_u \mid \times \mid C_v \mid}. \tag{3}$$

This linkage metric leads to higher probability of well formed cores than would lead the single-link metric, for example. We are working on experimentally validating and formally proving this statement.

REMARK 1. The global complexity of the *HAC* algorithm is not increased by the clusters' cores calculation. This happens because cores can be optimally computed in $\mathrm{O}(n \cdot p \cdot s)$, where $s$ is usually a small number, and $p < n$, generally $p << n$. Consequently, the cores computation is feasible in $\mathrm{O}(n^2)$, while *HACA* also runs in $\mathrm{O}(n^2)$.

## 4. Experimental Evaluation

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

As a case study, for experimenting our theoretical results described in Section 2 and for evaluating the performance of the *HAC* 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" (CorMac, 2005).

### 4.1. *Quality Measures*

We present in the following the measures that will be used for evaluating the results of the *HAC* and *HACA* clustering algorithms.

**Number of iterations.** It determines the global calculus complexity and it is used for evaluating the performances of both *HAC* and *HACA*.

**The degree of compactness of a partition.** The degree of compactness, or the *dispersion* (*DISP*) of a partition *K* is defined as follows:

$$DISP(K) = \frac{\sum_{l=1}^{p} \frac{\sum_{O_i, O_j \in K_l, i>j} d(O_i, O_j)}{C^2_{|K_l|}}}{p}, \tag{4}$$

where $K = \{K_1, \ldots, K_p\}$ is the set of clusters obtained after applying a clustering algorithm. *DISP* expresses the average distance between objects in a cluster, for all clusters, and $C^2_{|K_l|}$ represents the number of combinations of two elements from the set $K_l$.

As expected, the smaller the dispersion is, more compact clusters we have obtained and better was the cores choice at the beginning of the adaptive clustering process.

**Information gain.** For comparing the informational relevance of the attributes we used the *information gain (IG)* measure (Quinlan, 1993).

### 4.2. *Experiment* 1. *Cancer*

The breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

The objects to be clustered in this experiment are patients: each patient is identified by 9 attributes (Wolberg and Mangasarian, 1990). The attributes have been used to represent instances. Each instance has one of two 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 (CorMac, 2005).

In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values.

### 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 (Aeberhard *et al.*, 1992).

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

We have to mention that all attributes in this experiment are continuous.

### 4.5. *Results*

In this section we comparatively present the results obtained by applying *HAC* and *HACA* algorithms for the experiments described in the above subsections.

From Table 1 we observe that using *HAC* the number of iterations for finding the solution is much smaller than in the case of *HACA*. Also, the clusters obtained by *HAC* are roughly equally dispersed as those given by *HACA*. So, the clusters quality remains at about the same level, but the clustering process is more efficient.

To see how relevant the newly arrived attributes were, we present in Table 2 all attributes, from each experiment, in the decreasing order of their information gain (*IG*). The new attributes are the highlighted ones.

The computation of the information gain measure has been performed as follows. We cluster, using *HACA*, the set of extended objects, and a partition $K = \{K_1, K_2, \ldots, K_p\}$ is produced. Then, information gain has been computed for each attribute, considering that the class or label for each object is indicated by its cluster. So, an object $O_i \in K_j$ has as label $K_j$. The larger the information gain of an attribute is, the more discriminative that attribute is for the object set – it determines more strongly the objects separation into classes (clusters).

We think that there might be a connection between how discriminative a new added attribute is for the object set and how strong changes it causes to the old partitioning of

Table 1

The comparative results for *HAC* and *HACA*

| 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 HACA iterations for m attributes | 455 | 360 | 175 |
| No of HACA iterations for (m+s) attributes (N1) | 455 | 360 | 175 |
| No of HAC iterations for (m+s) attributes (N2) | 32 | 150 | 83 |
| Reduction of the no of iterations (N1-N2)/N1(%) | 92.9 % | 58.3 % | 52.5 % |
| DISP(HACA) for m attributes | 5.3507 | 8.0207 | 0.83 |
| DISP(HACA) for (m+s) attributes | 7.6505 | 7.9284 | 0.9871 |
| DISP(HAC) for (m+s) attributes | 7.78 | 8.21 | 1.06 |
| No of objects StrongCore/WeakCore<br>(% from no of objects) HAC | 92/0 | 18/41 | 53/0 |

the object set. However, intuitively, not only the discriminative power of a new attribute is important in this issue, but also the way this attribute is correlated with the old attributes.

Table 2 sustains these ideas. Namely, it can be seen that when the new attributes have large information gain values compared to the old attributes (Cancer and Wine experiments), the old partitioning is more easily adapted, because it does not require to be significantly changed. This fact reflects in the good quality of the partitions produced by *HAC*, i.e., the dispersion of the new partition, produced by *HAC*, is close to the dispersion of the old partition. By contrast, when new attributes are less significant compared to the old ones (Dermatology experiment), the old partition is less well-adapted and the new set of clusters quality decreases (dispersion increases). These observations consider equally weighted attributes.

Table 2

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

| Experiment | Order of attributes | IG of new attributes<br>/<br>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<br>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% |

We still have to inspect more carefully how the information gain of the new attributes and the correlations between the old and the new attributes have impact on partition changing when object feature set extension happens.

### 4.6. *Adaptive Horizontal Fragmentation in Object Oriented Databases*

A practical problem, where the proposed method can be efficiently used, is the adaptive horizontal fragmentation of object oriented databases.

A horizontal fragmentation approach that uses data mining clustering methods for partitioning object instances into fragments has been presented in (Darabant and Campan, 2004a), (Darabant and Campan, 2004b), (Darabant and Campan, 2004c), (Darabant *et al.*, 2004). Essentially, that approach takes full advantage of existing data, where statistics are already present, and develops fragmentation around user applications (queries) that are to be optimized by the obtained fragmentation. But real databases applications evolve in time, and consequently they require re-fragmentation in order to deal with new applications entering the system and others leaving. Obviously, for obtaining the fragmentation that fits the new user applications set, the original fragmentation scheme can be applied from scratch. However, this process can be inefficient.

We have applied the *HAC* method in the case when new user applications arrive in the system and the current fragments must be accordingly adapted (Campan *et al.*, 2005). The obtained results were good. The adaptive fragmentation keeps the fragmentation quality around the non-adaptive one and the processing time is improved, as the adaptive method performs, generally, in less time than the full fragmentation process.

## 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 the result is reached more efficiently using the proposed method than running the agglomerative hierarchical clustering algorithm *HACA* again from scratch, on the feature-extended object set.

But there are some situations when it is better to resort to a *HACA* clustering of the feature-extended object set, than to use the *HAC* algorithm. Intuitively, such situations can be: the addition of a large number of features or the addition of new features with small information gain and contradictory information with respect to the old feature set. A discussion about how new attributes information gain and correlation with the old attributes can influence the adaptive clustering results has been presented in Subsection 4.5. We intend to analyse this issue and to identify situations in which is better to run *HACA* from scratch on the feature-extended object set, than to run the adaptive clustering algorithm *HAC*. It would be interesting to obtain rigourous conditions about the effectiveness of applying our adaptive clustering approach.

Further work may be done in the following directions:

- to isolate conditions to decide when it is more effective to adapt (using *HAC*) the result of a clustering of the feature-extended object set than to restart the clustering using *HACA*;
- to study how the information brought into the system by the newly added attributes, their correlation with the initial ones influences the quality (dispersion) of clusters obtained by the *HAC* algorithm;
- to apply the adaptive algorithm on precise problems, from where the need of such an adaptive algorithm originated.

## References

Aeberhard, S., D. Coomans and O. de Vel (1992). The classification performance of RDA. *Technical Report 92–01*, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland.

Bagherjeiran, A., C.-F. Eick and R. Vilalta (2005). Adaptive Clustering: Better Representatives with Reinforcement Learning. Technical Report Number UH-CS-05-06, Department of Computer Science, University of Houston.

Campan, A., A.S. Darabant and G. Serban (2005). Clustering techniques for adaptive horizontal fragmentation in object oriented databases. In *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics ICTAMI 2005*. Alba-Iulia, Romania. pp. 263–274.

CorMac Technologies Inc, Canada (2005). *Discover the Patterns in Your Data*. `http://www.cormac-tech.com/neunet`.

Darabant, A.S., and A. Campan (2004a). Semi-supervised learning techniques: $k$-means clustering in OODB fragmentation. In *IEEE International Conference on Computational Cybernetics ICCC 2004*. Vienna University of Technology, Austria. pp. 333–338.

Darabant, A.S., and A. Campan (2004b). Hierarchical AI clustering for horizontal object fragmentation. In *Proceedings of Int. Conf. of Computers and Communications*. Oradea. pp. 117–122.

Darabant, A.S., and A. Campan (2004c). AI clustering techniques: a new approach to object oriented database fragmentation. In *Proceedings of the 8th IEEE International Conference on Intelligent Engineering Systems*. Cluj Napoca, Romania. pp. 73–78.

Darabant, A.S., A. Campan and O. Cret (2004). Hierarchical clustering in object oriented data models with complex class relationships. In *Proceedings of the 8th IEEE International Conference on Intelligent Engineering Systems*. Cluj Napoca, Romania. pp. 307–312.

Han, J., and M. Kamber (2001). *Data Mining*: *Concepts and Techniques*. Morgan Kaufmann Publishers.

Jain, A., and R. Dubes (1998). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey.

Jain, A., M.N. Murty and P. Flynn (1999). Data clustering. A review. *ACM Computing Surveys*, **31**(3), 264–323.

Manning, C., and H. Schutze (1999). *Foundation of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Quinlan, J.R. (1993). *C4.5*: *Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California.

Serban, G., and A. Campan (2005). Adaptive clustering using a core-based approach. Studia Universitatis "Babeş-Bolyai", *Informatica*, **L(2)**, 33–40.

Tan, P.-N., M. Steinbach and V. Kumar (2005). *Introduction to Data Mining*. Addison Wesley, chapters 8, 9.

Wolberg, W., and O.L. Mangasarian (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *Proceedings of the National Academy of Sciences*, U.S.A., vol. 87. pp. 9193–9196.

Wu, F., and G. Gardarin (2001). Gradual clustering algorithms. In *Proceedings of the 7th International Conference on Database Systems for Advanced Applications* (*DASFAA'01*). pp. 48–57.

**G. Şerban** has graduated from "Babeş-Bolyai" University of Cluj-Napoca, Faculty of Mathematics and Computer Science in 1987. She has received the PhD degree in computer science in 2003, with the "cum laude" distinction. She is an associate professor at the Department of Computer Science, Faculty of Mathematics and Computer Science, "Babeş-Bolyai" University of Cluj-Napoca, Romania. Her research interests include artificial intelligence, machine learning, multiagent systems, programming paradigms.

**A. Câmpan** has graduated from "Babeş-Bolyai" University of Cluj-Napoca, Faculty of Mathematics and Computer Science in 1999. She has received the PhD degree in computer science in 2006, with the "magna cum laude" distinction. She is an assistant professor at the Department of Computer Science, Faculty of Mathematics and Computer Science, "Babeş-Bolyai" University of Cluj-Napoca, Romania. Her research interests include databases, data mining, data privacy, Web programming.

## Adaptyvus hierarchinis klasterizavimas

Gabriela ŞERBAN, Alina CÂMPAN

Straipsnyje tiriamas adaptyvaus klasterizavimo uždavinys. Koncentruojamasi į perklasterizavimą, kai objektus charakterizuojanti savybių aibė yra papildoma. Siūlomas hierarchiniu kaupimu pagrįstas adaptyvus klasterizavimas. Siekiama gauti rezultatą efektyviau negu vykdant klasterizavimo algoritmą nuo pradžių. Aprašyti metodo efektyvumo tyrimo eksperimentai ir praktinis paskirstytų sistemų uždavinys (adaptyvus horizontalus objektiškai orientuotų duomenų bazių fragmentavimas), kuriame metodas gali būti efektyviai naudojamas.