Unsupervised Case Memory Organization: Analysing Computational Time and Soft Computing Capabilities

A. Fornells, E. Golobardes, D. Vernet, and G. Corral

Research Group in Intelligent Systems
Enginyeria i Arquitectura La Salle, Ramon Llull University
Quatre Camins 2, 08022 Barcelona, Spain
{afornells, elisabet, dave, guiomar}@salle.url.edu
http://www.salle.url.edu/GRSI

Abstract. There are problems that present a huge volume of information or/and complex data as imprecision and approximated knowledge. Consequently, a Case-Based Reasoning system requires two main characteristics. The first one consists of offering a good computational time without reducing the accuracy rate of the system, specially when the response time is critical. On the other hand, the system needs soft computing capabilities in order to construct CBR systems more tractable, robust and tolerant to noise. The goal of this paper is centred on achieving a compromise between computational time and complex data management by focusing on the case memory organization (or clustering) through unsupervised techniques. In this sense, we have adapted two approaches: 1) neural networks (Kohonen Maps); and 2) inductive learning (X-means). The results presented in this work are based on datasets acquired from medical and telematics domains, and also from UCI repository.

Keywords: Data Intensive, Maintenance and management for CBR, Case Memory, Soft Case-Based Reasoning, Clustering, Kohonen Maps.

1 Introduction

There are different problems that present a huge volume of information or very complex data. Therefore, they may present imprecision, uncertainty, partial truth, and approximated knowledge. Case-Based Reasoning (CBR) [1] tries to solve new problems using others previously solved. Nevertheless, CBR systems often have to face two main problems when they have to manage a huge dataset. The first problem is a reduction of system accuracy when the cases are composed by a large set of features. In this case, the system may not be able to detect the most relevant features. The second problem is an increase in CPU time because the retrieval phase depends of the number of features and cases. In this sense, the organization of the case memory may be crucial in order to reduce the computational cost of the retrieval phase (i.e. minimize the CPU time), and, if it is possible, improve system accuracy. On the other hand, soft computing techniques (e.g. neural networks) can be used for building CBR systems that

T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 241–255, 2006.

can exploit a tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieving more tractable, robustness, low solution cost and closer to the human making process [7].

Nowadays, there are lots of real domains with these characteristics. Our work in some of these areas has been the motivation of this paper. The first domain is related to applications on medical field. In fact, we mainly work on breast cancer diagnosis using mammographic images. A mammographic image is processed in order to identify the microcalcifications (μCa) that appear. After characterizing the μCa through a set of features, we diagnose each image using machine learning techniques. Previous studies applying machine learning techniques have found that these techniques improve the accuracy rate (in terms of correct classifications) but decrease the reliability rate (in terms of robustness and stability) compared to human experts [17]. The second domain, in which we are working, is related to security applications on computer networks. Comprehensive network security analysis must coordinate diverse sources of information to support large scale visualization and intelligent response [10]. Security applications require of some intelligence to recognize malicious data, unauthorized traffic, identify intrusion data patterns, learn from previous decisions and also provide a proactive security policy implementation [8,32].

We propose a data intensive approach based on a soft computing technique such as neural networks [4], Kohonen Maps [28], in order to organise the CBR case memory. The main goals of this approach are to manage complex data such as the explained domains, and improve the computational time spent on retrieving the information. Furthermore, these goals have to be defined to avoid decreasing the accuracy rate. We previously organized the CBR case memory using an inductive approach based on the adaptation of the X-means algorithm [38] in order to reduce the computational time [45]. For this reason, we compare both approaches to measure the benefit of our new proposal. The experiments presented in this work are based on datasets acquired from medical and telematics domains, and also from UCI repository [5].

The paper is organized as follows. Section 2 surveys related work using clustering techniques to organize the CBR case memory. Section 3 resumes the main ideas of Kohonen Maps and the adaptation of the X-means algorithm in order to explain later their roles in the case memory. Section 4 explains the approaches proposed to organize the case memory based on inductive learning and neural networks. Section 5 summarizes the experiments and a comparative study of the two approaches. Finally, we present the conclusions and further work.

2 Related Work

This section summarises related work found in the literature on the subject of clustering methods and regarding different approaches used to organise the case memory in Case-Based Reasoning systems.

First of all, most of the clustering methods are described in Hartigan's book [22]. There exist a large number of clustering algorithms. Thus, the choice of a clustering algorithm depends on the type of available data and on the particular purpose and application [19]. In general, clustering methods can be classified in the following approaches.

The first approach is the *partitioning method.* It consists of clustering training data into K clusters where $K < M$ and M is the number of objects in the data set. One of the most representative examples of this approach is the K-means algorithm [21]. There are special variations to improve some aspects of the algorithm. One variation is the K-medoids algorithm or PAM (Partition Around Medoids) [26], whose objective is to reduce the sensibility of the K-means algorithm when some extremely large values that distort data distribution are found. A variation of the K-medoids algorithm is the CLARA algorithm (Clustering LARge Applications) [27]. In this case, the algorithm extends the capabilities of the last algorithm in order to perform results when large data sets are explored. The automatic definition of the number of clusters was proposed in the X-means [38] algorithm. Finally, another widely used algorithm is the Self Organizing Maps (SOM) or Kohonen Maps [28], which is based on neural network theory [4].

The second approach is called *hierarchical method*, which works by grouping data objects into a tree of clusters. The hierarchical decomposition can be formed as a bottom-up or top-down procedure.

Another considered approach is based on the *density-based method.* The main objective of this method is to discover clusters with an arbitrary shape. This typically regards clusters as dense regions of objects in the data space that are separated by regions of low density (representing noise). The most popular algorithms in this category are the following: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [14], OPTICS (Ordering Points to Identify Clustering Structure) [3] and DENCLUE (DENsity-based CLUstEring) [24].

Grid-based method uses a multiresolution grid data structure that divides the space into a finite number of cells that form a grid structure on which all clustering operations are performed. This method has a constant processing time as an advantage, independently of the number of data objects. In this group we can identify algorithms such as CLIQUE (Clustering High-Dimensional Space) [2], STING (STatistical INformation Grid) [47], and WaveCluster [42] (an algorithm that clusters using the wavelet transformation).

Finally, *model-based method* uses mathematical and probability models. This method can be focused on two ways: firstly, as a statistical approach, and secondly, as a neural network approach. Some examples of these methods are AUTOCLASS [6] and COBWEB [15].

Hanson and Bauer stated that clustering of objects or events without a context, goal or information concerning the function of the derived clusters (as in [33]) is not likely to be useful for real-world problems [20]. Therefore, they proposed a different point of view and approach real-world problems by means of the WITT algorithm [20].

Regarding to the case memory organization in CBR systems, the most important approaches are the following: RISE [13] treats each instance as a rule that can be generalised. EACH [40] introduced the Nested Generalized Exemplars (NGE) theory, in which hyperrectangles are used to replace one or more instances, thus reducing the original training set. And finally, a method that avoids building sophisticated structures around a case memory or complex operations is presented by Yang and Wu [49]. Their method partitions cases into clusters where the cases in the same cluster are more similar than cases in other clusters. Clusters can be converted to new smaller case-bases. However, not all the approaches are focused on the organisation of the case memory in order to improve the case memory and, at the same time, the computational time.

3 Clustering Methods

Case-Based Reasoning (CBR) systems solve problems by reusing the solutions to similar problems stored as cases in a case memory [39] (also known as case-base). However, these systems are sensitive to the cases present in the case memory and often their good accuracy rate depends on the stored significant cases. Also, CBR systems have problems when a huge number of cases exist in the case memory, specially when the response time is critical (e.g. real time systems). Therefore, a compromise between computational time and soft computing capabilities will be pursued. Clustering the case memory tries to obtain different clusters of cases. Each cluster represents a generic case which corresponds to a region of the domain. Thus, the retrieval phase [1] only has to find a similar cluster to the new case. Consequently, the system improves its computational time. The key is: *Which is the better way to cluster the case memory?*

Previously to explain the integration of our new approach based on Kohonen [28], and the other approach based on the adaptation of X-Means [38] used to make the evaluation, we will make a short review of both algorithms. Although CBR [1,29,31] is used in a wide variety of fields and applications (e.g. diagnosis, planning, language understanding), we focus on CBR as an automatic classifier.

3.1 Kohonen Maps Algorithm

Kohonen Maps or Self-Organizing Maps (SOM) [28] are one of the major unsupervised learning paradigms in the family of artificial neural networks. The most important features of a SOM neural network are the following: (1) It preserves the original topology; (2) It works well even though the original space has a high number of dimensions; (3) It incorporates the selection feature approach; (4) Although one class has few examples they are not lost; (5) It provides an easy way to show data; (6) It is organized in an autonomous way to be adjusted better to data. On the other hand, the drawbacks of this technique are that it is influenced by the order of the training samples, and it is not trivial to define how many clusters are needed. They have successfully been used in a variety of clustering applications such as systems for Content-Based Image Retrieval (CBIR) [30] or documents retrieval [25]. Also, they have been used in a large variety of domains such as medical [46], chemical [44] or financial [11] data.

The SOM network is composed by two layers. First, there is the input layer, which is represented by a set of n-dimensional inputs that define the example to evaluate. The other is the output layer, which is a m-dimensional (although it is usually bidimensional) grid where neurons are placed. Each one of these neurons represents a cluster or model with certain properties. Also, each neuron is connected with all the n-inputs.

Figure 1 details the SOM training process algorithm. The models, which are represented by a set of properties using a n-dimensional vector, are iteratively fitted in order to create clusters with different properties. This process is achieved by means of updating the models using the training samples. For each training sample, a model is selected using a similarity measure shown in the Equation 1. Then, the model vector selected and the neighbours models are updated to better fit to this example by means of the Equation 2. This updating process is performed in two steps: (1) First, it affects the great majority of the models with a high influence value; (2) Second, it only affects the selected model and its immediately neighbours with a low influence. The training ends when the lowest error value is achieved, or the configured iteration ends.

```
input  : $CM$ is the case memory; $I_s$ is the new example; $Total$ is the number
         of iterations; $T_1$ is the number of iterations of the first phase; $E_{min}$ is
         the lower error accepted; $Map$ is the Kohonen map of size $K \times K$; $\alpha(0)$
         - $\alpha(F)$ and $\nu(0)$ - $\nu(F)$ are the initial and final values of the learning
         and neighbour factors respectively
output : $Map$ is the built Kohonen Map
Function trainingSOM is
    The $N_{i,j}$ models of $Map$ are randomly initialized between [0..1]
    for $(t=0;\ ((t < Total)\&(E_{min} < error));\ t{+}{+})$ do
        error=0
        forall $I_s \in CM$ do
            Let $N_{best}$ be the most similar model to $I_s$ using the Eq. 1
            All the neighbour models of $N_{best}$ are updated using the Eq. 2
            $error = error + \|\overline{I_s} - \overline{N_{best}}\|$
        $error = error /\ K \times K$
        $\alpha(t)$ and $\nu(t)$ are updated by the Eq. 3, if $t < T_1$
    return $Map$
```

Fig. 1. Cluster creation through the SOM algorithm

$$\forall i,j : 1 \leq i,j \leq K : \|\overline{I_s} - \overline{N_{best}}\| \leq \|\overline{I_s} - \overline{N_{i,j}}\| \tag{1}$$

$$\overline{N_{i,j}}(t+1) = \overline{N_{i,j}}(t) + \alpha(t) \cdot (\overline{I_s} - \overline{N_{i,j}}(t)) \tag{2}$$

$$X(t+1) = X(0) + (X(F) - X(0)) \cdot \frac{t}{T_1} \tag{3}$$

3.2 *SX*-Means Algorithm

The adaptation of the *X*-means algorithm [38] in order to cluster the CBR case memory was proposed in [45]. This variation finds spherical data groups through moving the location of the centre of these spheres, called centroids. The centroid is the mean value for all the objects in the cluster. It also uses splitting to determine the right number of centroids and, consequently, the number of clusters. It restricts the search of the best cluster distribution by setting a lower and an upper threshold of the number of clusters. The algorithm starts allocating the centroids with *K*-means [21] using the lower value of *K*. It continues adding centroids until the upper threshold of *K* is reached. At each step only one centroid is inserted by splitting the original in two; then, a sub-cluster from the original cluster is detected. Thus, centroids relocation is achieved regarding to the same elements of the original cluster. The centroid set that achieves the best score is selected, based on a BIC (Bayesian Information Criterion) function. This is a recursive process that finishes when *K* reaches the upper bound and the local sub-*K*-means has run for all centroids. Figure 2 resumes the main steps of the *X*-means algorithm. We will call this adaptation using spheres as *SX*-means (Sphere *X*-means) algorithm.

K-means and *X*-means algorithms have been applied in a variety of clustering applications including systems for 3D objects modeling [12], computer architecture research [18], network security [8] or text summarization [35].

```
input  : CM is the case memory; lowerbound and upperbound are the minimal
         and maximum value of K;
output : The K clusters defined
Function X-means is
    Let k[i] be the actual number of clusters by class
    Let kbest[i] be the best number of clusters by class
    Let accuracy be the rate of examples correctly classified
    maxaccuracy=0
    for (i=0; (i < NumberOfClasses); i++) do
        k[i]=kbest[i]=lowerbound class i
        initialize k[i] clusters ramdomly in class i
    for (i=0; (i < NumberOfClasses); i++) do
        for (j=k[i]; (j < upperbound class i); j++) do
            cluster class i in j partitions
            verify system accuracy
            if (accuracy >maxaccuracy) then
                maxaccuracy=accuracy
                save configuration in kbest
    return kbest
```

Fig. 2. Cluster creation through the *SX*-means algorithm (*X*-means adaptation)

4 Organizing the Case Memory

This section presents our new approach based on Kohonen Maps, and it also describes the previous approach based on SX-Means.

4.1 Kohonen Maps into CBR: The Neural Network Approach

Kohonen Maps [28] are a soft computing technique that allows the management of uncertain, approximate, partial truth and complex knowledge.

We propose a case memory organized such as a map of size $K \times K$ as we can see in the left part of the Figure 3, where each neuron is represented by a vector that models the behaviour and the properties of the samples that it represents. We propose a Kohonen Map training based on the X-means strategy to automatically define the number of clusters: execute several map configurations using different sizes of K, and select the one which has the lowest error. This is a critical decision because we want to improve the retrieval time through the separation of data in several clusters, and the lowest error value will be achieved with few clusters. Thus, a minimal value of clusters needs to be forced. This way of organizing the case memory affects the retrieval and retain phase as CBR function (see Figure 4) describes. The difference is that this approach only compares the cases of the most similar cluster instead of comparing all the elements. Thus, CPU time is reduced. On the other hand, clusters are built at the beginning of the process. The SOM network can not be readjusted and it needs to be rebuilt. Therefore, the optimal environment is the one where the memory is not modified.

This strategy has been implemented over a framework called SOM-CBR (*Self-Organizing Maps inside a Case-Based Reasoning*). Other authors have adapted SOM approach to work as the CBR [34], but they do not integrate SOM inside the CBR in order to manage complex data and to improve the retrieval time, that are our main goals. Also, we propose an automatic definition of the map size in this work.

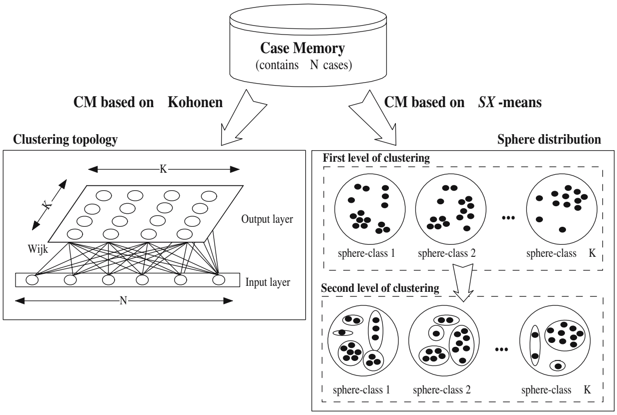

Fig. 3. Case memory representation through the Kohonen and SX-Means approaches

4.2 *SX*-Means into CBR: The Inductive Approach

This approach proposes a case memory organization based on two levels of clustering as we can see in the right part of the Figure 3 by means of the SX-means algorithm [45]. Firstly, a construction of the spheres is done based on the class distribution of the cases present in the case memory. The concept of sphere was introduced in the CaB-CS and exploited with success in preliminary work such as [17]. The success of this type of Case Memory representation is based on two aspects: first of all this representation greatly improves the speed of the CBR system, and secondly the spheres offer high reliability in the selection of the candidate cases. Each case from the original case memory is distributed to one sphere depending on the class associated with the case. All the cases that belong to the same sphere represent the same class. The union of all spheres is the whole set of cases in the original case memory. Later, a second level of clustering is applied using the results of the previous one. Consequently, each sphere contains a set of clusters obtained using the SX-means algorithm. This strategy is implemented in ULIC (*Unsupervised Learning in CBR*)[45].

As in the SOM-CBR approach, the organization of the case memory affects the retrieval and retain phase as Figure 4 describes. The retrieval phase is only applied over the cases of the selected cluster, and it allows CBR to reduce the CPU time. Adding a new example into the case memory implies updating the centroids of clusters. If a lot of examples are added the case memory performance can be drastically reduced. For this reason, rebuilding the clusters is the only way to assure a good performance. Anyway, the update process could be done in background mode.

```
input  : $CM$ is the case memory; $I_s$ is the example to classify; $K - NN$ is the
         number (odd) of cases to retrieve
output : $C$ is the classification predicted
Function CBR is
    //Retrieve phase
    if method configured is Kohonen then
        Let $S$ be the most similar cluster of the $I_s$ example
        Select the most $K - NN$ similar samples from $S$ in comparison with $I_s$
    if method configured is $SX$-means then
        Select the most $K - NN$ similar centroids
    //Reuse phase
    Propose a classification $C$ for $I_s$ using the retrieved cases
    //Revise phase
    Evaluate if class $C$ is correct
    //Retain phase
    Add $I_s$ in case memory if it is 'useful' by means of an updating ($SX$-means)
    or rebuilding (Kohonen) task
    return $C$
```

Fig. 4. CBR cycle [1] adapted to apply the Kohonen and the SX-means clustering strategies

5 Experiments and Results

In this section we shall describe the data sets for testing the proposed techniques and the obtained results.

5.1 Testbed

The performance rate is evaluated using the datasets described in Table 1. *Breast Cancer Wisconsin, Glass, Ionosphere, Iris, Sonar and Vehicle* come from UCI repository [5]. The rest of them are from medical and telematics domains. The medical datasets deal with *breast cancer diagnosis*. These are mammographic images digitalized by the Computer Vision and Robotics Group from the University of Girona. The *Biopsy* [16] and *Mammogram* [17] datasets contain samples of mammographies previously diagnosed by surgical biopsy in Trueta Hospital (in Girona), which can be benign or malign. DDSM [23] and MIAS [43] are public mammographic image datasets, which have been studied and preprocessed in [37,36] respectively. DDSM and MIAS classify mammography densities, which was found relevant for the automatic diagnosis of breast cancer. Experts classify them either in four classes (according to BIRADS [41] classifications) or three classes (classification used in Trueta Hospital).

Regarding to telematics domain, datasets are focused on network security. There are no standard datasets that contain all the information obtained after a thorough security test is performed, so there are no class labels for the data and so no obvious criteria to guide the search. On the other hand, security experts have noticed that collecting logs, capturing network traffic and identifying potential threats is becoming difficult to handle when managing large data sets. A corporate network can handle many devices, thus a thorough test can result in a great amount of data [8]. Therefore, trying to manually find a behaviour pattern or certain vulnerabilities becomes a difficult task.

In order to perform our evaluation of Kohonen Maps and SX-means in a completely unsupervised environment such as data from security tests, we have applied these clustering algorithms to three datasets obtained from Consensus system [9]. These datasets differ in the number and detail of the attributes that describe a case (see Table 1). As explained before, this domain is completely unsupervised; therefore the number of classes is unknown. This is why techniques such as Kohonen Maps and SX-means can help discovering 'natural' grouping in a set of patterns without knowledge of any class labels.

All the proposed datasets aim to be a representative benchmark of the different characteristics of the type of problems to solve. These datasets have been tested using CBR, Kohonen and SX-means . All the approaches have been tuned with 1-Nearest Neighbour algorithm and Euclidean distance without weighting methods as retrieval strategy. We have chosen this configuration because our goal is focused on the evaluation of the retrieval time.

5.2 Results and Discussion

This section presents a discussion over the clustering methods explained before. First, we analyse the accuracy rate and the computational time needed to retrieve

Table 1. Description of the datasets used in this work

Code	Dataset	Cases	Features	Classes	Uncertainty
BC	*Breast-cancer (Wisconsin)*	699	9	2	Yes
GL	*Glass*	214	9	6	No
IO	*Ionosphere*	351	34	2	No
IR	*Iris*	150	4	3	No
SO	*Sonar*	208	60	2	No
VE	*Vehicle*	846	18	4	No
BI	*Biopsy*	1027	24	2	Yes
MA	*Mammogram*	216	23	2	Yes
DD	*DDSM*	501	143	4	Yes
M3	*Mias-3C*	320	153	3	Yes
MB	*Mias-Birads*	320	153	4	Yes
NS1	*Network Security (Consensus) 1*	45	60	-	Yes
NS2	*Network Security (Consensus) 2*	45	57	-	Yes
NS3	*Network Security (Consensus) 3*	45	165	-	Yes

a case using both approaches over the UCI Repository and medical datasets. Second, we perform a qualitative study of the case memory organization obtained using the evaluated clustering methods in telematics domain.

Table 2 summarizes the results of SOM-CBR (Kohonen) and ULIC (SX-means) approaches. In SX-means approach we have clustered cases in several spheres in order to detect different behaviours of the data contained in them. On the other hand, in Kohonen approach we have mapped data patterns onto a n-dimensional grid of neurons or units. For each technique, we present the average percentage of accuracy resulting of a 10-fold stratified cross-validation, their corresponding standard deviations, and the average computational time (i.e. CPU time) in milliseconds of one case resolution. In addition, the results shown in Table 2 are the mean of ten executions using several random seeds in a P4-3Ghz computer with 1 GRAM. All the experiments have been done without retaining any case in the case memory because this paper does not focus on Retain phase.

As we can observe, results in general improve both the mean accuracy and the CPU time of classifying one case. Clustering the case memory is the result of grouping similar data, which possibly have the same classification. When the Retrieve phase is applied, CBR only compares with potentially 'good' examples and not with redundant data. We consider 'good' examples these examples which are similar in comparison with the new example to classify.

The accuracy rate has been analysed by means of the t-test student (at 95% confidence level). In SX-means CM approach the accuracy rate is usually maintained or improved (not significantly) in comparison with Linear CM in UCI problems. However, the accuracy rate is significantly reduced in some problems (SO, VE, BI, MB and M3) which present more uncertainty. On the other hand, SOM CM approach is more stable and it provides results like the Linear CM. Also, it improves the results in MA dataset in comparison with Linear CM, and

Table 2. Summary of the mean percentage of accuracy rate (%AR), the standard deviation (std) and the mean retrieval time of one case (in milliseconds) of a CBR with three case memory organization approaches: linear, SOM and SX-means. The best accuracy rates are marked in **Bold**. The ↑ and ↓ indicate if the cluster method significantly improves or decreases the accuracy rate in comparison with Linear CM when a t-test student (at 95% of confidence level) is applied . The √ indicates that SOM CM significatively improves SX-means CM.

Code	**Linear CM**		**SOM CM**			***SX*-Means CM**		
	%AR (std.)	Time	%AR (std.)		Time	%AR (std.)		Time
BC	96.14 (2.1)	1.8000	96.42 (2.6)		0.7000	**96.71 (1.9)**		1.0200
GL	69.16 (7.3)	0.6000	70.66 (7.8)		0.2100	**70.79 (8.7)**		0.5500
IO	**90.32 (4.2)**	0.3600	89.12 (4.8)		0.0800	90.31 (5.3)		0.0060
IR	96.32 (3.1)	0.3000	96.00 (3.2)		0.0150	**97.33 (3.2)**		0.0015
SO	**87.02 (6.9)**	0.3600	85.58 (7.2)	√	0.1400	82.93 (7.7)	↓	0.1600
VE	69.05 (6.1)	0.4800	**69.15 (5.7)**	√	0.2200	65.60 (3.7)	↓	0.0080
BI	**83.15 (3.5)**	0.7200	82.08 (3.7)		0.4300	81.40 (3.7)	↓	0.3100
MA	62.50 (13.7)	0.1200	**68.06 (8.3)**	√ ↑	0.0400	63.89 (9.8)		0.0900
DD	46.51 (5.4)	1.9800	**46.41 (4.1)**		1.2000	46.17 (5.2)		1.1000
M3	**70.81 (6.9)**	1.5000	69.57 (6.09)	√	0.7000	65.34 (6.2)	↓	0.5400
MB	**70.31 (5.5)**	1.5000	**70.31 (5.4)**	√	0.7000	60.16 (9.2)	↓	0.5400

it significatively improves the results in SO, VE, MA, M3 and MB datasets in relation with SX-means CM.

Concerning to the the CPU time, the two approaches always drastically reduce computational time requirements. This is directly related to the number of clusters defined by each approach. Table 3 summarizes the clusters defined for each configuration explained in Table 2. In both approaches, the ideal number of clusters has been tuned in order to minimize the minimal square error. SX-means tends to build more clusters than SOM because SX-means defines several 'patterns' for each class, whereas SOM defines patterns that work as 'index' to compare only with the most potentially similar cases. Thus, SX-means only compares with the 'patterns', and SOM compares with the patterns and its cases. This situation produces that the computational time in SOM is higher than in SX-means approach because it has to use more information. Eq. 4, 5 and 6 model the cost (time) needed to retrieve one case by Linear, SOM and SX-means approaches respectively, where Tr represents the number of cases in the case memory and K the number of clusters used. Depending on the number of clusters (K), the size of case memory (Tr), and the cases distribution in the clusters the difference of performance between SOM CM and SX-means CM could vary.

$$time(Linear) = O(Tr) \tag{4}$$

$$time(SOM) = O(K + \frac{Tr}{K}) \tag{5}$$

$$time(SX - means) = O(K) \tag{6}$$

Table 3. Summary of the number of the case memory clusters for each dataset and method. Also, SOM approach includes the map size ($K \times K$), and SX-means includes the number of clusters by class.

Code	Classes	Clusters in SOM CM	Clusters in SX-means CM
BC	2	30 (K=8)	42 (27-15)
GL	7	7 (K=6)	78 (20-15-10-0-20-3-10)
IO	2	44 (K=8)	30 (24-6)
IR	3	10 (K=6)	34 (20-4-10)
SO	2	37 (K=8)	52 (25-27)
VE	4	62 (K=10)	115 (25-20-35-35)
BI	2	4 (K=4)	44 (28-16)
MA	2	8 (K=16)	90 (50-40)
DD	4	3 (K=8)	10 (1-4-2-3)
M3	3	6 (K=10)	8 (2-3-3)
MB	4	6 (K=10)	8 (2-3-3)
NS1	-	3 (K=8)	3 (3)
NS2	-	8 (K=8)	8 (8)
NS3	-	8 (K=8)	8 (8)

Therefore, we can conclude that CPU time is improved and the accuracy rate is maintained for all the problems when the SOM approach is applied, because it seems to be more suitable to tackle general or uncertain problems due to its soft computing capabilities. On the other hand, SX-means improves the CPU time but the accuracy rate decreases in problems with uncertainty.

Regarding to network security and clustering, not only SX-means [8] but also Kohonen Maps have revealed very good results when using port scanning and operative system fingerprinting information as main features. We must highlight that this domain was completely unsupervised; thus, the number of classes was unknown. However, both techniques have found 8 different clusters for the used datasets. They have identified groups of similar computers, but have also found devices that unexpectedly appear separated from what it seamed like similar devices. Therefore, these techniques can help analysts handling information obtained from security tests in order to detect abnormal groups of devices or atypical system behaviours.

6 Conclusions and Further Research

This paper has proposed a case memory organization based on Kohonen Maps in order to manage complex and uncertain problems, and also reduce the retrieval time. Furthermore, we have analysed this approach in comparison with a Linear CM organization and a SX-means CM organization previously proposed in [45] over datasets from UCI Repository and from medical and telematics domains.

The results have shown that the soft computing capabilities of Kohonen Maps allow CBR to better retrieve the information in comparison with a SX-means CM organization when the problems present uncertainty, and faster in comparison with the Linear CM organization. However, the SX-means CM needs less operations to retrieve one case because only needs to compare with 'pattern' (centroids) and not with the cases of the 'patterns'. Therefore, the solution with best accuracy is the Linear CM, the faster is the SX-means CM, and the more balanced is SOM CM. Anyway, SOM case memory organization is more suitable for managing uncertain domains.

One weak point of both approaches, and more concretely in SOM-CBR, is the Retain phase. The case memory is clustered at the beginning of the process and the clusters are built to promote the groups between similar data. If we add knowledge in the case memory in form of new cases, these relations can be altered and the performance is reduced. One issue of further work would be focused on the Retain phase in order to add new cases without reducing the system performance (accuracy rate and computational time).

All the studied datasets are composed by numeric attributes because the metric used in SX-means and Kohonen Maps do not support discrete data with reliability. Thus, it would be interesting to study the application of other metrics such as the Heterogeneous distance [48].

Acknowledgements

We would like to thank the Spanish Government for their support under grants TIC 2002-04160-C/02-02, TIN 2005-08386-C05-04 and CIT-390000-2005-27, and *Generalitat de Catalunya* (DURSI) for its support under grants 2005SGR-302 and 2006FIC-0043. Also, we would like to thank *Enginyeria i Arquitectura La Salle* of Ramon Llull University for their support to our research group.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: Foundations issues, methodological variations, and system approaches. *IA Communications*, 7:39–59, 1994.
2. R. Agrawal, J. Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.
3. M. Ankerst, M.M. Breunig, H. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 49–60. ACM Press, 1999.
4. Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
5. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
6. P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, pages 153–180, 1996.
7. W. Cheetham, Simon Shiu, and R. Weber. Soft case-based reasoning. *The Knowledge Engineering*, 0:1–4, 2005.
8. G. Corral, E. Golobardes, O. Andreu, I. Serra, E. Maluquer, and A. Martínez. Application of clustering techniques in a network security testing system. *Artificial Intelligence Research and Devolopment*, 131:157–164, 2005.

9. G. Corral, A. Zaballos, X. Cadenas, and A. Grane. A distributed vulnerability detection system for an intranet. In *Proceedings of the 39th IEEE International Carnahan Conference on Security Technology*, pages 291–295, 2005.
10. J. Dawkins and J. Hale. A systematic approach to multi-stage network attack analysis. *Second IEEE Int. Inf. Assurance Workshop*, 2004.
11. G. Deboeck and T. Kohonen. *Visual Explorations in Finance using self-organizing maps.* Springer-Verlag, 1998.
12. A. Domingo and M.A. Garcia. Hierachical clustering of 3d objects and its application to minimum distance computation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 5287–5292, 2004.
13. F. Domingos. Control-sensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11(1-5):227–253, 1997.
14. M. Ester, H.P. Kriegel, and X. Xu. A database interface for clustering in large spatial databases. In *Knowledge Discovery and Data Mining*, pages 94–99, 1995.
15. D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. In *Machine Learning*, pages 2:139–172, 1987.
16. J.M. Garrell, E. Golobardes, E. Bernadó, and X. Llorà. Automatic diagnosis with genetic algorithms and case-based reasoning. *AI in Engineering*, 13(4):362–367, 1999.
17. E. Golobardes, X. Llorà, M. Salamó, and J. Martí. Computer aided diagnosis with case-based reasoning and genetic algorithms. *Journal of Knowledge Based Systems*, 15:45–52, 2002.
18. G. Hamerly, E. Perelman, , and B. Calder. Comparing multinomial and k-means clustering for simpoint. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software*, 2006.
19. J. Han and M. Kamber. Data mining: Concepts and techniques, 2000.
20. S. J. Hanson. *Conceptual Clustering and Categorization: Bridging the Gap Between Induction and Causal Models*, volume 3. Kaufmann, 1990.
21. J. Hartigan and M. Wong. A k-means clustering algorithm. In *Applied Statistics*, pages 28:100–108, 1979.
22. J.A. Hartigan. *Clustering Algorithms.* John Wiley and Sons, New York, 1975.
23. M. Heath, K. Bowyer, D. Kopans, R. Moore, and P.J. Kegelmeyer. The digital database for screening mammography. *International Workshop on Dig. Mammography*, 2000.
24. A. Hinneburg and D.A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Knowledge Discovery and Data Mining*, pages 58–65, 1998.
25. S. Kaski, T. Honkela, K. Lagus, and T. Kohonen. Websom—self-organizing maps of document collections. *Neurocomputing*, 21(1):101–117, 1998.
26. L. Kaufman and P.J. Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416. Y. Dodge, 1987.
27. L. Kaufman and P.J. Rousseeuw. Finding groups in data: An introduction to cluster analysis. *John Wiley & Sons*, 1990.
28. T. Kohonen. The self-organizing map. *In Proc. of the IEEE*, 78:1464–1480, 1990.
29. J. Kolodner. Reconstructive memory, a computer model. *Cognitive Science*, 7:281–328, 1983.
30. J. Laaksonen, M. Koskela, and E. Oja. Picsom: Self-organization maps for content-based image retrieval. *Proceedings of International Joint Conference on NN*, 1999.
31. R. López de Mántaras and E. Plaza. Case-based reasoning : An overview. *AI Communications, IOS Press*, 10(1):21–29, 1997.

32. F. Martin. *Case-Based Sequence Analysis in Dynamic, Imprecise, and Adversarial Domains.* PhD thesis, Universitat Politècnica de Catalunya, 2004.
33. R.S. Michalski. Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. Technical Report 1026, LIIA, Ensais/Univ. Louis-Pasteur, Urbana, Illinois, 1980.
34. L. E. Mujica, J. Vehí, and J. Rodellar. A hybrid system combining self organizing maps with case based reasoning in structural assessment. In *Artificial Intelligence Research and Development*, volume 131, pages 173–180. IOS Press, 2005.
35. T. Nomoto and Y. Matsumoto. An experimental comparison of supervised and unsupervised approaches to text summarization. In *First IEEE International Conference on Data Mining*, page 630, 2001.
36. A. Oliver, J. Freixenet, A. Bosch, D. Raba, and R. Zwiggelaar. Automatic classification of breast tissue. *Iberian Conference on Pattern Recognition and Image Analysis*, pages 431–438, 2005.
37. A. Oliver, J. Freixenet, and R. Zwiggelaar. Automatic classification of breast density. In *International Conference on Image Processing*, 2005.
38. D. Pelleg and A. Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference of Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.
39. C. K. Riesbeck and R. C. Schank. *Inside Case-Based Reasoning.* Lawrence Erlbaum Associates, Cambridge, MA, 1989.
40. S. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:277–309, 1991.
41. T. H. Samuels. *Illustrated Breast Imaging Reporting and Data System BIRADS.* American College of Radiology Publications, 3rd edition, 1998.
42. G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of 24th International Conference Very Large Data Bases, VLDB*, pages 428–439, 24–27 1998.
43. J. Suckling, J. Parker, and D.R. Dance. The mammographic image analysis society digital mammogram database. In A.G. Gale, editor, *Proceedings of 2nd Internat. Workshop on Digital Mammography*, pages 211–221, 1994.
44. A. Ultsch. Self organized feature maps for monitoring and knowledge acquisition of a chemical process. *International Conference on Artificial Neural Networks*, pages 864–867, 1993.
45. D. Vernet and E. Golobardes. An unsupervised learning approach for case-based classifier systems. *Expert Update. The Specialist Group on Artificial Intelligence*, 6(2):37–42, 2003.
46. T. Villmann. Neural networks approaches in medicine - a review of actual developments. *Proceedings of European Symposium on Artificial Neural Networks*, pages 165–176, 2000.
47. W. Wang, J. Yang, and R.R. Muntz. STING: A statistical information grid approach to spatial data mining. In *The VLDB Journal*, pages 186–195, 1997.
48. D.R. Wilson and T.R. Martinez. Improved heterogeneus distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.
49. Q. Yang and J. Wu. Keep it simple: A case-base maintenance policy based on clustering and information theory. In *Proceedings of the Canadian AI Conference*, pages 102–114, 2000.