An unsupervised learning approach for Case-Based classifier systems

David Vernet and Elisabet Golobardes

Enginyeria i Arquitectura La Salle, Universitat Ramon Llull,
Pg. Bonanova 8, 08022 Barcelona, Spain
{dave,elisabet}@salleurl.edu

Abstract. Case-Based Classifier Systems obtain low accuracies on generalisation and higher waste on CPU time when the class distribution space is not well defined. This paper presents the Mean Sphere and the Mean K-Means approach based on unsupervised learning to improve the CPU time and to improve or maintain the accuracy. We use clustering in an unsupervised way to decide which is the representational space of each class. The concept of clustering is introduced in two levels. **First level** cluster the training data into *spheres*, obtaining one *sphere* for each class. **Second level** consists of clustering the spheres in order to detect the behaviour of the elements present in the sphere. In this level two policies are applied, Mean Sphere and Mean K-Means approaches. Experiments using different domains, most of them from the UCI repository, show that the CPU time is considerably decremented while maintaining, and sometimes improving, the accuracy of the system.

1 Introduction

Case-Based Reasoning systems are often faced with two main problems when a great number of features and cases exist in the case memory. The first problem is the reduction of the system performance because the system can not detect different behaviours on the data. The second problem is an increase in CPU time because the retrieval phase has to use all the information available.

This paper describes the Mean Sphere and the Mean K-Means (MKM) approach to organising the case memory. Our aim is twofold: (1) to reduce the CPU time and (2) to distinguish between different behaviours of the data, avoiding noisy instances. The organisation of the case memory proposed consists of applying two levels of clustering. Firstly, a construction of the *spheres* is done based on the class distribution of the cases present in the case memory. Later, a second level of clustering is applied using the results of the previous one. In the second level, each sphere contains a set of clusters obtained using the Mean Sphere approach or K-Means algorithm. Both approaches have been introduced into our ULIC (Unsupervised Learning In CBR) platform.

This paper is organized as follows: first, section 2 presents the related work about clustering and unsupervised learning in general; the next section introduces the unsupervised organisation approaches. Then, section 4 explains the

testbed and experiments used and the results obtained. Finally, section 5 presents the conclusions and further work.

2 Related work

This section summarises related work present in the literature for clustering methods and for different approaches used in Case-Based Reasoning systems to organise the case memory.

First of all, most of the clustering methods are described in Hartigan's book [13].

There exist a large number of clustering algorithms. The choice of clustering algorithm depends on the type of data available and on the particular purpose and application [10].

In general, clustering methods can be classified in the following approaches:

The first approach is the *partitioning methods*. They consist of clustering training data into k clusters where $k < n$ and n is the number of objects in the data set. An example of this approach is *k-means algorithm* [12]. There are special variations to improve some aspects of the algorithm. The first variation is the *k-medoids algorithm* or PAM (Partition Around Medoids) [15]. In this algorithm, the objective is to reduce the sensibility of the k-means algorithm when some extremely large values that distort the distribution of data are found. A variation of the k-medoids algorithm is the *CLARA algorithm* (Clustering LARge Applications) [16]. In this case, the algorithm extends the capabilities of the last algorithm so as to perform the results when large data sets are explored.

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

Another approach considered are the *density-based methods*. The main objective of these methods is to discover clusters with an arbitrary shape. These typically regard 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: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [5], OPTICS (Ordering Points to Identify Clustering Structure) [2] and DENCLUE (DENsity-based CLUstEring) [14].

Grid-based methods use a multiresolution grid data structure that divides the space into a finite number of cells that form a grid structure on which all operations for clustering are performed. This method has as an advantage a constant processing time, independently of the number of data objects. We can identify in this group algorithms such as CLIQUE (Clustering High-Dimensional Space) [1], STING (STatistical INformation Grid) [23], and WaveCluster [22] (an algorithm that clusters using the wavelet transformation).

Finally, *model-based methods* use mathematical and probability models. These methods can be focused in two ways: firstly, as a statistical approach, and second as a neural network approach. Some examples of this approach are AUTOCLASS [3] and COBWEB [6].

One criticism directed at researchers that use conceptual clustering (similar to [20]) has been that the clustering of objects or events without a context, goal or some information concerning the function of the derived clusters is not likely to be useful for real-world problems [11]. Hanson and Bauer propose a different point of view and approach real-world problems with algorithms like WITT [11].

On the other hand, in the literature there exist different approaches of Case-Based reasoning to produce a new organisation of case memory. The most important approaches are the following: RISE [4] treats each instance as a rule that can be generalised. EACH [21] 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 [24]. 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   Unsupervised organisation approaches

The spheres construction process performs the first level of the organisation of the case memory. The concept of *sphere* had been introduced in the CaB-CS [8] and exploited with success in preliminary works such as [9, 17].

The success of this type of representation of the Case Memory 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 to the case (see figure 1). One sphere contains a subset n of cases from the original case memory (NC). 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, such as is represented in equation 1.

$$\sum_{i=1}^{k} n_i = NC \tag{1}$$

where n_i is the number of cases that belong to the sphere i, k is the number of different classes and NC is the number of cases in the training data.

3.1   Mean Sphere

The Mean Sphere uses the previously computed spheres to generate the centroid of each class (see algorithm in figure 2).

Each Mean Sphere (of any class C) contains the mean value for each feature computed using the cases present in the sphere of class C.
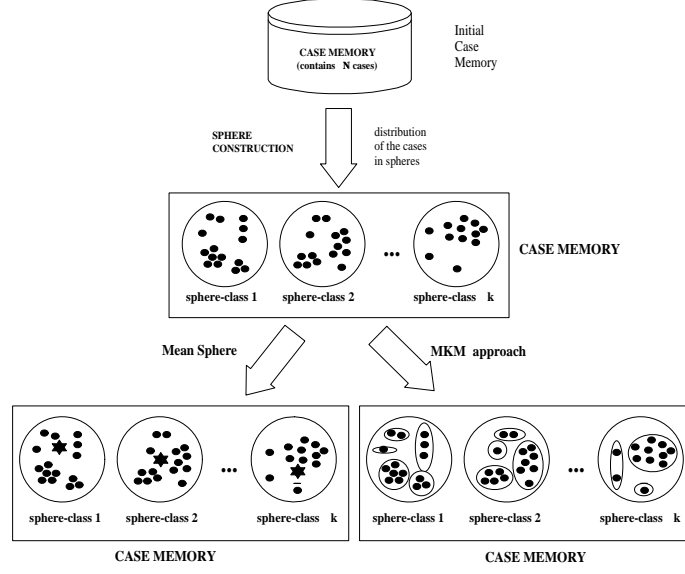
Fig. 1. Representation of the Case Memory for the two approaches. The symbol •
represents a case and the symbol ⋆ represents the centroid of the sphere.

```
1. for each sphere Sᵢ in k different classes
2.          Let be nᵢ represented as N the number of cases of sphere i
3.          for each attribute j in F different features
4.                 centroid Sᵢⱼ = Σ_{c=1}^{N} a_{ci}/N
5.          end for
6. end for
```

Fig. 2. Algorithm to compute the centroid ⋆ of each sphere. The variable a_{ci} represents
the value of the attribute j in the cth case of the ith sphere.

Later, in the retrieval phase, we search for the sphere that represents the
new case, we compute the *Euclidean Distance* between each centroid and the
new case. We classify the new case depending on the class represented by the
sphere selected.

3.2   Mean K-Means Approach

The *Mean K-Means (MKM)* approach takes the original spheres and obtains
a new categorisation of the case memory by applying the *k-means clustering
method* (figure 3) internally in each sphere . This new categorisation also pro-
duces a smaller case memory. Thus, the CPU time is reduced.

For each sphere we obtain C_i clusters and we compute the centroid of each
cluster in the sphere. Then, we select the nearest cluster to the new case in the

```
1. Choose an initial partition of the cases into k clusters. This
   is a random assignment to k clusters.
2. Compute the distance from every case to the mean of each cluster
   and assign the cases to their nearest clusters.
3. Recompute the cluster means following any change of cluster mem-
   bership at step 2.
4. Repeat steps 2 and 3 until no further changes of cluster membership
   occur in a complete iteration. The procedure has now converged
   to a stable k-partition.
```

Fig. 3. K-Means algorithm modified to cluster each sphere.

same way as the Mean Sphere approach. The number of clusters for each sphere can be different.

The clustering method is configured independently of the class treated. Thus, we define the optimum number of clusters for each sphere.

Moreover, we can configure a class with n_i clusters (one cluster for each case). In this case we can combine the application of traditional Case-Based Reasoning in one class with clustering in the other classes. Therefore, a sphere with n_i clusters is equivalent to applying CBR in this class.

4   Experiments and results

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

4.1   Testbed

In order to evaluate the performance rate, we use eight datasets. Datasets can be grouped in two ways: *public* and *private*. The datasets and their characteristics are listed in table 1.

Public datasets are obtained from the UCI repository [19]. They are: *Breast Cancer Wisconsin, Glass, Ionosphere, Iris, Sonar and Vehicle.* **Private datasets** come from our own repository. They deal with *diagnosis* of breast cancer. Datasets are *Biopsy* and *Mammogram*. *Biopsy* [7] is the result of digitally processed biopsy images, whereas *Mammogram* consists in detecting breast cancer using the microcalcifications (μCa) present in a mammogram [18, 9]. In *mammogram* each example contains the description of several μCa present in the image; in other words, the input information used is a set of real valued matrices.

These datasets were chosen in order to provide a wide variety of application areas, sizes, combinations of feature types, and difficulty as measured by the accuracy achieved on them by current algorithms. The choice was also made with the goal of having enough data points to extract conclusions.

Table 1. Datasets and their characteristics used in the empirical study.

	Dataset	Reference	Samples	Features	Classes	Inconsistent
1	*Biopsy*	*BI*	1027	24	2	Yes
2	*Breast-cancer (Wisconsin)*	*BC*	699	9	2	Yes
3	*Glass*	*GL*	214	9	6	No
4	*Ionosphere*	*IO*	351	34	2	No
5	*Iris*	*IR*	150	4	3	No
6	*Mammogram*	*MA*	216	23	2	Yes
7	*Sonar*	*SO*	208	60	2	No
8	*Vehicle*	*VE*	846	18	4	No

The configuration of the ULIC system for this paper is 1-Nearest Neighbour algorithm. Training cases are represented by *spheres*. We have not used weighting methods in order to test the reliability of our system. The retain phase is limited to the original training data. The learning process in the test is skipped in order to to check the behaviour of the spheres.

4.2 Results

In this section we describe the results obtained by the ULIC system using both approaches.

In this paper we have only performed one distance function in order to test the new organisation of the case memory. Other similarity functions will be studied in further research.

The mean accuracy is the result of 10 different executions of the stratified 10-fold cross-validation.

Firstly, in the *Mean Sphere approach*, we have a sphere for each class without internal clustering. We have computed the centroid values for each sphere. Therefore, we reduce all the cases that belong to a class to one case.

Secondly, in the *MKM approach*, we cluster each sphere in order to detect different behaviours of the data contained in each sphere.

In table 2 we present the results obtained by the traditional 1-NN CBR, the *Mean Sphere approach* and the *MKM approach*. As we can observe, the results in general improve both the mean accuracy and the CPU time of resolution of one case.

As we can observe *MKM approach* improves the prediction accuracy obtained by the *Mean approach* in some of cases. The explication of this is that MKM not reduce so high the case memory. In *Mean approach* the case memory is reduced to a unique case for each class. So, the CPU time is lower in the *Mean approach* in all of data sets.

Table 3 shows the optimum number of clusters for each data set and the average accuracy, the standard deviation and the average CPU time of resolution for one case.

Table 2. This table compares the mean percentage of correct classifications (%PA), standard deviation (std) and mean CPU time (CPUt) of the Retrieval phase using traditional CBR, a first level of clustering (Mean Sphere) and a second level of clustering (MKM approach). The results that improve the prediction accuracy or the CPU time of the traditional CBR are marked with a $\sqrt{}$.

Ref.	CBR			MeanSphere			MKM		
	%PA	CPUt	std	%PA	CPUt	std	%PA	CPUt	std
BC	96.14	56.34	1.45	96.28 $\sqrt{}$	0.13 $\sqrt{}$	1.87	96.71 $\sqrt{}$	31.96 $\sqrt{}$	1.53
BI	83.15	199.63	3.55	79.07	0.27 $\sqrt{}$	4.66	81.40	1.52 $\sqrt{}$	3.76
GL	69.16	33.60	7.32	53.74	0.51 $\sqrt{}$	7.01	70.79 $\sqrt{}$	30.52 $\sqrt{}$	8.70
IO	90.03	99.61	4.28	81.77	0.45 $\sqrt{}$	5.07	90.31 $\sqrt{}$	1.60 $\sqrt{}$	5.38
IR	95.33	6.00	3.06	92.67	0.20 $\sqrt{}$	2.00	97.33 $\sqrt{}$	1.50 $\sqrt{}$	3.27
MA	62.50	90.41	13.73	64.81 $\sqrt{}$	0.23 $\sqrt{}$	9.42	63.89 $\sqrt{}$	65.68 $\sqrt{}$	9.86
SO	82.21	198.71	6.99	70.67	1.83 $\sqrt{}$	7.19	82.93 $\sqrt{}$	89.20 $\sqrt{}$	7.73
VE	66.90	125.06	4.33	84.04 $\sqrt{}$	0.54 $\sqrt{}$	4.39	65.60	2.00 $\sqrt{}$	3.75

The number of clusters can be different for each sphere. Thus we can define different numbers of clusters in different classes. For each data set, we have determined with previous executions of the system which is the best combination of number of clusters in order to configure the system for further experiments.

Table 3. Best configuration of the clusters. We indicate for each class the number of clusters generated. A number n_i indicates that there is a number of clusters equal to the number of cases in the sphere

Dataset	Classes	Number of clusters
BI	2	28-16
BC	2	27-n_i
GL	7	20-n_i-10-n_i-20-n_i-10
IO	2	24-6
IR	3	20-4-10
MA	2	n_i-40
SO	2	25-n_i
VE	4	25-20-35-35

It is important to note that the CPU time is quite long when any sphere with n_i clusters exists. It means that we apply traditional Case-based Reasoning in this sphere because each case represents a cluster.

5 Conclusions and further research

We have introduced two *unsupervised learning approaches* in a traditional Case-Based Reasoning System achieving our initial objectives.

It is important to keep in mind that the main goal was to maintain the prediction accuracy obtained by the traditional Case-based Reasoning improving the speed on CPU time of the Retrieval phase. As we have seen, we have improved the speed and we have also improved the results.

The results show that the clustering methods notably reduce the CPU time when resolving a new case. Moreover, it maintains or even improves the prediction accuracy obtained by the traditional Case-based Reasoning.

MKM approach obtains better prediction accuracies because this approach organises the memory case with a higher number of cases than the Mean approach. However, the CPU time is lower in the Mean approach.

On the other hand we can introduce in further work the idea of not predict when the clustering method had not a reliable result. In this way, we want to increase the reliability of the system, in particular when we are working in medical environments (like mammograms, biopsies and so on).

Another further objective is to add to the system other clustering algorithms that automatically detect the optimum number of clusters. In the current configuration we have found the best configuration for each problem.

Another aspect to take into consideration is the possibility of applying clustering methods with non-real data (e.g. discrete values). Algorithms such as WITT and variations of k-means algorithms should be adapted in order to solve this question.

Finally, we want to improve the accuracy results adding weighting methods in the clustering algorithms.

Acknowledgements

References

1. 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 Conference on Management of Data*, pages 94–105, 1998.

2. M. Ankerst, M.M. Breunig, H. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. pages 49–60, 1999.

3. P. Cheeseman and J. Stutz. Bayesian classification (autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, pages 153–180. 1996.

4. P. Domingos. Context-sensitive feature selection for lazy learners. In *AI Review*, volume 11, pages 227–253, 1997.

5. 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.

6. D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. In *Machine Learning*, pages 2:139–172, 1987.

7. J.M. Garrell, E. Golobardes, E. Bernadó, and X. Llorà. Automatic diagnosis with Genetic Algorithms and Case-Based Reasoning. *Elsevier Science Ltd. ISSN 0954-1810*, 13:367–362, 1999.

8. E. Golobardes. *Aportacions al raonament basat en casos per resoldre problemes de classificació*. PhD thesis, Enginyeria La Salle, Universitat Ramon Llull, Juny 1998.

9. 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*, pages 45–52, 2002.

10. J. Han and M. Kamber. Data mining: Concepts and techniques, 2000.

11. S. J. Hanson. Conceptual clustering and categorization: Bridging the gap between induction and causal models. In Y. Kodratoff and R. S. Michalski, editors, *Machine Learning: An Artificial Intelligence Approach (Volume III)*, pages 235–268. Kaufmann, San Mateo, CA, 1990.

12. J. Hartigan and M. Wong. A k-means clustering algorithm. In *Applied Statistics*, pages 28:100–108. 1979.

13. J.A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, New York, 1975.

14. 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.

15. 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, North-Holland. Y. Dodge.

16. L. Kaufman and P.J. Rousseeuw. Finding groups in data: An introduction to cluster analysis. *John Wiley & Sons*, 1990.

17. X. Llorà, E. Golobardes, M. Salamó, and J. Martí. Diagnosis of microcalcifications using Case-Based and Genetic Algorithms. In *Proceedings of Engineering of Intelligent Systems (EIS2000)*, volume 1, pages 254– 263, 2000.

18. J. Martí, J. Español, E. Golobardes, J. Freixenet, R. García, and M. Salamó. Classification of microcalcifications in digital mammograms using case-based reasonig. In *International Workshop on digital Mammography*, 2000.

19. C.J. Merz and P.M. Murphy. UCI Repository for Machine Learning Data-Bases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. *Irvine, CA: University of California, Department of Information and Computer Science*, 1998.

20. R.S. Michalski. Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. Technical Report 1026, Urbana, Illinois, 1980.

21. S. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:277–309, 1991.

22. G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 428–439, 24–27 1998.

23. 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.

24. Q. Yang and J. Wu. Keep it Simple: A Case-Base Maintenance Policy Based on Clustering and Information Theory. In *Proc. of the Canadian AI Conference*, pages 102–114, 2000.