A Methodology for Analyzing Case Retrieval from a Clustered Case Memory

Albert Fornells, Elisabet Golobardes, Josep Maria Martorell,
Josep Maria Garrell, Núria Macià, and Ester Bernadó

Grup de Recerca en Sistemes Intel·ligents
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull
Quatre Camins 2, 08022 Barcelona, Spain
{afornells,elisabet,jmmarto,josepmg,nmacia,esterb}@salle.url.edu
http://www.salle.url.edu/GRSI

Abstract. Case retrieval from a clustered case memory consists in finding out the clusters most similar to the new input case, and then retrieving the cases from them. Although the computational time is improved, the accuracy rate may be degraded if the clusters are not representative enough due to data geometry. This paper proposes a methodology for allowing the expert to analyze the case retrieval strategies from a clustered case memory according to the required computational time improvement and the maximum accuracy reduction accepted. The mechanisms used to assess the data geometry are the complexity measures. This methodology is successfully tested on a case memory organized by a Self-Organization Map.

Keywords: Case Retrieval, Case Memory Organization, Soft Case-Based Reasoning, Complexity Measures, Self-Organization Maps.

1 Motivation

The computational time of Case-Based Reasoning (CBR) [1] systems is mainly related to the case memory: the greater the size, the greater the time. This fact can be a problem for real time environments, where the user needs a fast response from the system. For this reason, a reduction of the number of cases is sometimes the only way for achieving this goal.

The case memory organization plays an important role because it helps CBR to concentrate on the potentially useful cases instead of the whole case memory. We focus on a case memory organization based on the definition of groups of similar cases by means of clustering techniques. The new retrieve phase selects the set of clusters most similar to the input case, and then it retrieves a set of cases from them. Although the reduction of cases improves the computational time, it may also imply a degradation of the accuracy rate if the clusters are not representative enough. This last issue depends on the data complexity[1].

[1] The data complexity refers to the class separability and the discriminant power of features, and not about its representation in the case memory.

R.O. Weber and M.M. Richter (Eds.): ICCBR 2007, LNAI 4626, pp. 122–136, 2007.

We present a methodology for analyzing the behavior of the different ways in which the case retrieval can be performed from a clustered case memory according to the performance desired. The performance is defined as the relation between the required computational time improvement and the maximum accuracy reduction accepted with respect to using all the cases.

The first step is to know the performance of each one of the different case retrieval strategies. For this reason, we propose a taxonomy of them represented as a decomposition based on the number of clusters selected and the percentage of cases used from them in the retrieve phase. Thus, the strategies defined in the taxonomy are run over a wide set of datasets with the aim of evaluating its performance. The next step is to analyze the results. However, these executions generate a large volume of results which are very complex and difficult to study. That is why we have developed a scatter plot to understand in a more intuitive way these results instead of using huge results tables. This plot is a 2-D graphical representation in which the relations between the computational time improvement and the maximum reduction of the accuracy rate accepted are drawn for all the configurations from the last taxonomy. It allow us to compare the performance between the strategies and with respect to a CBR system based on a linear search of the case memory. Nevertheless, the behavior of the strategies depends on the definition of clusters, which are more closely related to data complexity. By taking into account the analysis of dataset complexity, we are able to identify separate behaviors that otherwise would remain hidden. The analysis of the scatter plot is done according to *a priori* classification of the dataset based on three levels of defined complexity.

The proposed methodology gives us a framework to understand the data mining capabilities of the clustering technique used to organize the case memory for a particular dataset characterized by its complexity, which heavily influences the case retrieval strategy. Therefore, there is not an absolute best strategy, the selection depends on the performance desired by the user.

The empirical test of the methodology is applied in a case memory organized by a Self-Organization Map (SOM) [12] over 56 datasets. We select SOM as clustering technique due to our experience using it [8,9,10]. However, this study could be easily extended to other clustering techniques.

The paper is organized as follows. Section 2 summarizes some related work about strategies for organizing the case memory and data complexity. Section 3 presents the methodology for setting up the case retrieval. Section 4 describes the experiments and discusses the results. Finally, Section 5 ends with the conclusions and further research.

2 Related Work

This section contains a brief review of the case memory organization and the importance of studying the data complexity.

The Case Memory Organization. This issue is tackled from several points of view in order to improve the computational time.

K-d trees [21] organize the features in nodes, which split the cases by their values. The main drawbacks are the treatment of missing values and the reduced flexibility of the method because of the tree structure. Both problems are successfully solved in Case Retrieval Nets [13], which organize the case memory as a graph of feature-value pairs. They employ a spreading activation process to select only the cases with similar values. Decision Diagrams [15] work in a similar way to the k-d trees but using a directed graph.

Other approaches link the cases by means of the similarity between them such as Fish-and-sink [19,22], or using relationships defined by the knowledge of the domain such as CRASH system [5]. In both cases, these links allow CBR to find out the similarity of cases in the case base.

The reduction of the number of operations can also be done by indexing the case memory using the knowledge from the domain like in the BankXX system [18], which is based on a conceptualization of legal argument as heuristic search. Another way of indexing the information is by the identification of clusters by means of clustering algorithms: X-means [16] in ULIC [20] or SOM [12] in [6,9].

On the other hand, there are approaches based on distributing the case memory through multi-agent architectures [17], or applying massive parallel solutions [14]. These solutions let CBR reduce the execution time, but they do not reduce the number of operations.

The Utility of the Complexity Measures. Complexity measures highlight the data geometry distribution offering an indicator that estimates to what extent the classes are interleaved, a factor that affects the accuracy. The dataset analysis allows us to understand the classifier behavior on a given dataset. Nowadays, the complexity measures are used to: (1) predict the classifier's error on a particular dataset, based on a study [3] where a linear relation was found between the estimated complexity of a dataset and the classifier's error; and (2) characterize the difficulty of a classification problem and provide a map that illustrates the domain of competence of classifiers in the complexity space. Basu and Ho [2] presented many metrics that measure the problem complexity from several aspects (power of discriminant attributes, class separability, degree of overlap, topology, etc.). However, it is difficult to set the complexity with only one measure. For this reason, their combination is a more reliable tool [8].

3 Description of the Methodology

This section explains the different parts of the methodology proposed for understanding the behavior of the case retrieval strategies from a clustered case memory. First, we present the strategy map as a taxonomy of the different ways in which the retrieval can be performed considering the clusters and cases used. Next, we detail the scatter plot for analyzing the results obtained from running over the strategies of the taxonomy. Finally, we introduce the characterization of the datasets according to its complexity.

3.1 The Strategy Map

The strategy map is a taxonomy of the case retrieval strategies from a clustered case memory based on two factors as Fig. 1 shows. The **factor of the selected clusters** identifies three possible situations on the basis of the number of clusters selected. Areas numbered 1 and 2 are situations where only the best cluster is retrieved. In contrast, areas numbered 5 and 6 represent the opposite situation where all the clusters are used. Finally, the intermediate situation is defined by the areas numbered 3 and 4, where a set of the clusters is selected. Note that area number 6 corresponds to a situation where all the cases are used in the same way as a CBR system that carries out a linear search over the whole case base: all the cases from all the clusters (*All_All*).

Although the number of selected clusters for retrieval can be set by the user, we could use a threshold (ϑ) for requiring the minimum similarity accepted between the input case C and a cluster M_X to select it. This similarity can be computed as the complement of the normalized Euclidean distance (see Eq. 1), and other metrics can be applied. N is the number of attributes.

$$similarity(C, M_X) = |1 - distance(C, M_X)| = \left| 1 - \sqrt{\frac{\sum_{n:1}^{N}(C(n) - M_X(n))^2}{N}} \right| \quad (1)$$

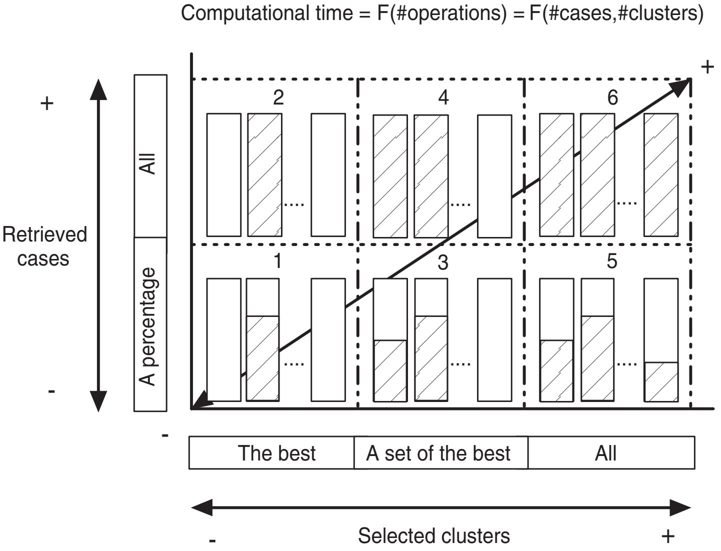


Fig. 1. The strategy map classifies the case retrieval strategies into six areas. Each one represents a combination of the number of clusters and cases selected for applying the case retrieval. The rectangles are the clusters and the lined area the retrieved cases from each cluster. The diagonal arrow from area number 1 to area number 6 shows the increase of the computational time as more cases are used.

Anyway, the selection of one of these situations depends on three issues: (1) the capability of the cluster for representing the data; (2) the desired computational time improvement; and (3) the maximum reduction of the accuracy rate accepted due to reduction of cases. For example, a high reduction of the computational time implies to select few clusters but the accuracy rate can be degraded if clusters are not representative. Therefore, the selection of the clusters is a compromise between issues 2 and 3, which are highly influenced by the capability of modeling the data complexity (issue 1).

On the other hand, the **factor of the retrieved cases** represents how many cases from the clusters are compared to C in the retrieve phase. The cases retrieved can be: (1) an arbitrary percentage or (2) all the cases. This issue is the difference between the areas 1-2, 3-4, 5-6 previously explained. Thus, the computational time can be reduced while the capability of exploring new clusters remains intact. To compute the percentage of retrieved cases, we propose two metrics based on the goodness of the clusters.

The first proposal defines a linear relation between the cluster contribution and its goodness. Eq. 2 computes the percentage as a normalized percentage between the similarity of the selected clusters K_M.

$$\%\ of\ cases\ from\ M_X = \frac{similarity(C, M_X)}{\sum_{m \in K_M} similarity(C, m)} \cdot 100 \tag{2}$$

Furthermore, it could be interesting to promote the contribution of clusters with high goodness values and, at the same time, penalizing the contribution of clusters with lower goodness values. This is exactly the behavior of an *arctangent* function: linear in the central zone, restrictive in one extreme, and permissive in the other. Moreover, other interesting aspects to consider are the possibility of adjusting the gradient of the curve and defining for which similarity values the contribution of elements has to be more or less important (the inflection point). These behaviors are parametrized by the μ and x_0 arguments. Finally, the arctangent domain is transferred from [-$\pi/2, \pi/2$] to [0, 1] by dividing by π, and adding 0.5. These requirements define Eq. 3.

$$\%\ of\ cases\ from\ M_X = 0.5 + \frac{arctg(\mu * (similarity(C, M_X) - x_0))}{\pi} \cdot 100 \tag{3}$$

Fig. 2 shows how the μ and x_0 arguments in Eq. 3 determine the percentage of the contribution. High values of μ and x_0 imply a highly restrictive selection. Alternatively, low values imply lower levels of restrictiveness.

However, if the selected clusters are not similar with respect to the input case, the global sum of the percentage will be less than 100%. In contrast, the sum will be greater than 100% if they are very similar. Therefore, the normalization of the last equation can help to adjust (increasing or decreasing) the total amount of cases to retrieve. Eq. 4 normalizes Eq. 3.

$$\%\ of\ cases\ from\ M_X\ (normalized) = \frac{\%\ of\ cases\ from\ M_X}{\sum_{m \in K_M} \%\ of\ cases\ from\ m} \cdot 100 \tag{4}$$

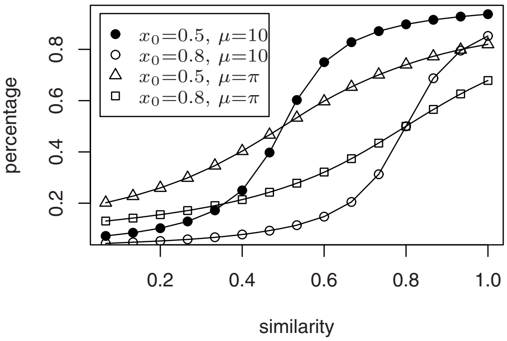
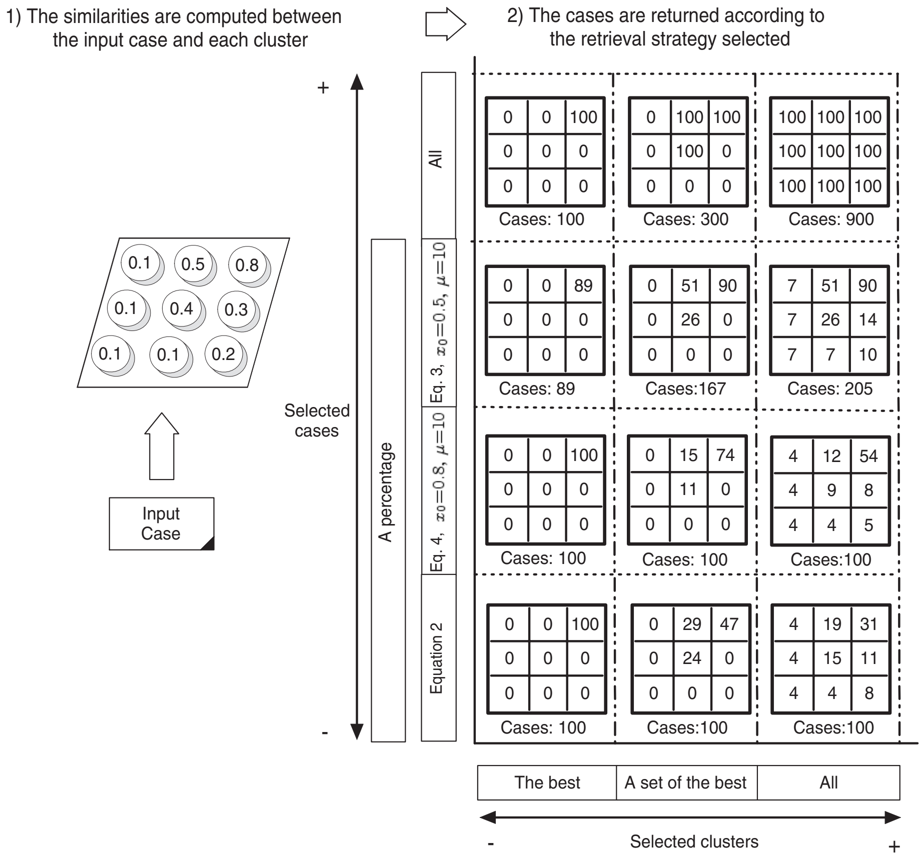


Fig. 2. Graphical representation of Eq. 3. The μ and x_0 arguments adjust the function according to the gradient and the inflection point desired. The pair $x_0 = 0.8$ and $\mu = 10$ is the most restrictive, and the pair $x_0 = 0.5$ and $\mu = 10$ is the most permissive. The other two configurations are intermediate situations.

Example. Fig. 3 illustrates the impact of the case retrieval strategies defined by Eq. 2, 3, and 4 through a case study. The left part represents a case memory clustered in nine clusters. Each cluster contains 100 cases and its goodness is computed by Eq. 1. The right part describes the behavior of twelve strategies, where each area is a combination of the two factors previously explained. Moreover, each area shows how many cases are retrieved from each one of the nine clusters. A value equal to zero means that the cluster is not selected. Therefore, the combination of both factors determines the degree of dispersion in which system explores the case memory. The definition of both issues depends on the performance desired by the user according to the capability of clusters for representing the domain, which is related to the data complexity. The greater the computational time improvement, the fewer clusters and cases have to be used.

Let's suppose a situation in which the user wants to improve the computational time but without reducing the accuracy rate. If the clusters are well defined, the best strategy is to select only the best cluster because it contains all the potentially useful cases.

Nevertheless, the clusters can present a lack of precision due to the data complexity. In this scenario, the best solution is to retrieve more than one cluster. Although this decision affects the computational time improvement, it can be compensated by applying strategies which focus on retrieving a percentage of cases from the selected clusters. The strategies based on Eq. 2 and 4 provide the same number of cases as the strategy that retrieves all the cases, the difference being that they explore other data clusters. The strategy built from Eq. 2 uses a linear contribution, and Eq. 4 uses a contribution weighted by the goodness of the cluster. On the other hand, Eq. 3 follows the same philosophy as the strategy based on Eq. 4 but increasing the total amount of cases as a consequence of the contribution not being normalized.


1) The similarities are computed between the input case and each cluster
2) The cases are returned according to the retrieval strategy selected
0.1
0.5
0.8
0.1
0.4
0.3
0.1
0.1
0.2
Input Case
+
Selected cases
-
A percentage
All
Eq. 3, x_0=0.5, μ=10
Eq. 4, x_0=0.8, μ=10
Equation 2
The best
A set of the best
All
-
Selected clusters
+

	The best	A set of the best	All
All	0 0 100 0 0 0 0 0 0 Cases: 100	0 100 100 0 100 0 0 0 0 Cases: 300	100 100 100 100 100 100 100 100 100 Cases: 900
Eq. 3, x_0=0.5, μ=10	0 0 89 0 0 0 0 0 0 Cases: 89	0 51 90 0 26 0 0 0 0 Cases:167	7 51 90 7 26 14 7 7 10 Cases: 205
Eq. 4, x_0=0.8, μ=10	0 0 100 0 0 0 0 0 0 Cases: 100	0 15 74 0 11 0 0 0 0 Cases: 100	4 12 54 4 9 8 4 4 5 Cases:100
Equation 2	0 0 100 0 0 0 0 0 0 Cases: 100	0 29 47 0 24 0 0 0 0 Cases:100	4 19 31 4 15 11 4 4 8 Cases:100

Fig. 3. The left part exemplifies a case memory clustered in nine clusters, and the right part shows the behavior of several case retrieval strategies. Each matrix corresponds to the cases retrieved from each cluster for a given configuration. A zero value means that the cluster is not selected. The total number of cases retrieved is below the matrix.

The extreme situation appears when the goodness of the clusters is small, and a full exploration of all the clusters is needed. In this case, the strategies of Eq. 2, 3, and 4 explore the case memory in different degrees of dispersion without utilizing all the case memory.

In summary, the performance is a balance between the computational time and the accuracy rate, where the goodness of clusters plays a crucial role.

3.2 Evaluation of the Case Retrieval Strategies

The evaluation of all strategies for a wide set of datasets implies the generation of huge tables which are complex to interpret. For this reason, we propose a 2-dimensional scatter plot to represent its performance as shown in Fig. 4.

The x axis depicts the ratio of the computational time of a case retrieval strategy (in this case $S2$ or $S3$) with respect to another strategy (in this case $S1$, which is the *All_All* situation featured by a linear search of the case memory) in logarithmic scale. A value closer to 0 indicates that there is no reduction in the number of operations, while growth in the negative direction implies a high reduction in this magnitude. A logarithm scale gives a better visual representation: for example, the value -1 of the logarithm in $S3$ means that $S3$ does 10% less operations than $S1$. As we can observe, the reduction in $S3$ is higher than in $S2$.

The y axis depicts the rank of each strategy averaged over all datasets. That is, if we consider m strategies tested over n datasets, $R_{i,j}$ is the rank-order assigned to the strategy i in comparison to the other ones, tested over the dataset j. From here, R_i is the medium rank for the strategy i, calculated as:

$$R_i = \frac{\sum_{j=1}^{n} R_{i,j}}{n} \tag{5}$$

Values next to 1 of this measure indicate that the strategy i is usually the best of the m tested, while values next to m indicate the opposite. Fig. 4 shows that S1 is better than S2, and S2 is better than S3 in terms of how many times each one has the best accuracy rate. The size of the drawn circumferences is proportional to the standard deviation of R_i. Thus, the bigger the circumference ($S2$, for example), the higher variability in the values obtained of $R_{i,j}$, and the smaller the circumference ($S1$ or $S3$), the lower the variability with respect to the medium rank.

Furthermore, the concept of critical distance (CD) is introduced to define the minimum distance from which the existence of a significant difference can be considered between the values of R_i, for a given confidence level [7]. The horizontal lines delimit the zone of equivalence between strategies. In this case, $S1$ and $S2$ are not significantly different but $S2$ reduces the cases used by almost by 50%. In contrast, $S3$ is significantly worse than $S1$ and $S2$.

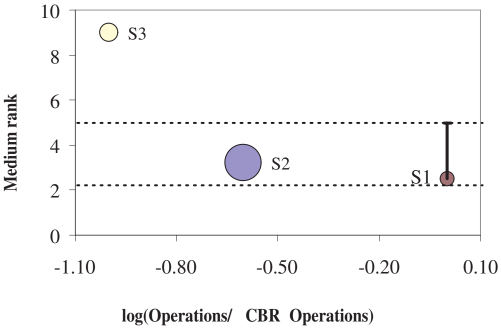


Fig. 4. Comparison of the performance between the strategies $S1$, $S2$, and $S3$. The x axis measures the computational time improvement, and the y axis represents how many times each strategy has the best accuracy rate. The vertical error bar is the CD value.

3.3 Data Complexity in the Case Retrieval Strategies

The data complexity influences the building of clusters and the strategy's behavior. We consider the boundary complexity [11] in order to evaluate how data geometry may affect the behavior of retrieval strategy.

The complexity space is defined by the complexity measures F3, N1, and N2 [8]. F3 is the feature efficiency, and it defines the efficiency of each feature individually describing to what degree the feature takes part in the class separability. The higher the value, the higher the power of class discrimination, implying a linear separation. N1 and N2 are the length of the class boundary and the intra/inter class nearest neighbor distances respectively. Both measures compute the distance between the opposite classes. Our metric is composed of the N1·N2 product because it emphasizes extreme behaviors. While a low value of these measures indicates a high class separability, a high value does not necessarily provide a conclusion about complexity. Thus, the dataset properties are evaluated by the discriminant power of features and the class separability.

Fig. 5 depicts the complexity space where the point (1,0) is considered the point of minimum complexity (mCP) whereas the point (0,1) corresponds to the maximum possible complexity (MCP). This is due to the meaning of each of the metrics and allows us to sort the complexity space into zones of low complexity (next to mCP) and zones of high complexity (next to MCP). As a matter of fact, the distance to the point mCP, in this space, distinguishes the studied datasets into three groups: (1) problems with a low complexity (type A, corresponding to distances to mCP less than 0.5), (2) problems with high complexity (type C, with value greater or equal than 1), and (3) problems in the middle of the two extremes (type B). Thus, we can evaluate the performance of each strategy in a more precise way.

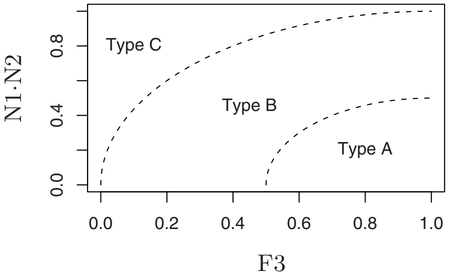


Fig. 5. The complexity space is divided into three types of complexity, where *A* is the less complex, and *C* the most complex

4 Experiments, Results, and Discussion

This section tests the methodology outlined in section 3. First, we briefly review how to integrate SOM in a Case-Based Reasoning system. Then, the datasets selected for the experimentation are described and classified by the complexity map. Finally, we present and discuss the results.

4.1 Self-Organization Map in a Case-Based Reasoning System

SOM projects the original N-dimensional input space into a new space with less dimensions by highlighting the most important data features to identify groups of similar cases. SOM is constituted of two layers: (1) the input layer composed of N neurons, where each neuron represents one of the N-dimensional features of the input case; and (2) the output layer composed of $M \times M$ neurons, where each one represents a set of similar cases by a director vector of N dimensions. Each input neuron is connected to all the output neurons. When a new input case C is introduced in the input layer, each neuron X from the output layer computes a degree of similarity between its director vector and the input case C applying a metric such as the normalized Euclidean distance (see Eq. 1). Thus, CBR can determine the clusters most similar to the input case. SOM is integrated into CBR in the SOMCBR framework (Self-Organization Map in a Case-Based Reasoning system) [9].

4.2 Testbed

The setting up of the case retrieval strategy according to the required performance is studied over several datasets of different domains and characteristics. There are 56 discrimination problems where miasbi, mias3c, ddsm, and μCa are related to breast cancer diagnosis [9] and the remaining datasets belong to the UCI Repository [4]. The datasets of D-classes are split in D datasets of two classes (each class versus all other classes) to increase the testbed. The dataset name, the number of features and instances, and the complexity type are described in table 1. The complexity map of datasets is drawn in Fig. 6.

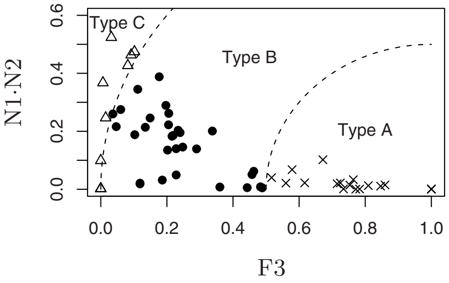


Fig. 6. Complexity map of the 56 analyzed datasets

4.3 Assessing the Performance of the Case Retrieval Strategies

The configurations from the strategy map studied for analyzing the behavior of the case retrieval strategy are summarized in Fig. 7. SOM is used for organizing the case memory. The strategies are executed applying a 10-fold stratified cross-validation with the following common configuration: (1) The retrieve phase uses

Table 1. Description of test datasets: name, number of attributes and instances, and complexity type. The suffix $2cX$ means that the dataset classifies the classes X versus the rest of classes. The datasets are sorted by their complexity.

Dataset	Attributes	Instances	Type	Dataset	Attributes	Instances	Type
segment2c2	19	2310	A	wav2c3	40	5000	B
iris2c2	4	150	A	wav2c1	40	5000	B
glass2c1	9	214	A	miasbi2c3	152	320	B
thy2c1	5	215	A	ddsm2c1	142	501	B
thy2c2	5	215	A	mias3c2c2	152	322	B
segment2c6	19	2310	A	thy2c3	5	215	B
segment2c7	19	2310	A	mias3c2c1	152	322	B
wine2c2	13	178	A	ddsm2c4	142	501	B
iris2c1	4	150	A	miasbi2c2	152	320	B
segment2c1	19	2310	A	wisconsin	9	699	B
wine2c1	13	178	A	wbcd	9	699	B
glass2c2	9	214	A	wav2c2	40	5000	B
miasbi2c4	152	320	A	sonar	60	208	B
glass2c4	9	214	A	wpbc	33	198	B
wine2c3	13	178	A	glass2c6	9	214	B
iris2c3	4	150	A	mias3c2c3	152	322	B
wdbc	30	569	A	biopsia	24	1027	B
segment2c3	19	2310	B	vehicle2c3	18	846	B
segment2c5	19	2310	B	vehicle2c2	18	846	B
glass2c3	9	214	B	bal2c3	4	625	C
vehicle2c1	18	846	B	bal2c2	4	625	C
segment2c4	19	2310	B	bal2c1	4	625	C
tao	2	1888	B	ddsm2c3	142	501	C
hepatitis	19	155	B	heartstatlog	13	270	C
glass2c5	9	214	B	μCa	21	216	C
ionosphere	34	351	B	ddsm2c2	142	501	C
vehicle2c4	18	846	B	pim	8	768	C
miasbi2c1	152	320	B	bpa	6	345	C

the Euclidean distance as similarity function. (2) The reuse phase proposes a solution using the most similar case. (3) The retain phase does not store new cases. Moreover, SOMCBR is tested with 10 random seeds and the size map is automatically computed as the map with the lowest error [9].

Next, the scatter plot for analyzing the strategies is built considering the complexity characterization in Fig. 8(a), 8(b), and 8(c). They show the accuracy (measured through the medium rank) versus the computational time (represented by the logarithm of the quotient of the number of operations). The strategy of reference is the *All_All* because it works like a CBR system with linear search of the case memory.

Fig. 8(a) represents the low complexity problems (type A). We observe a linear correlation between the values of the two axes, which indicates that the effect of the SOM is weak: the accuracy of the method is directly proportional to the number of retrieved cases (the correlation coefficient is 0.96 for the strategies

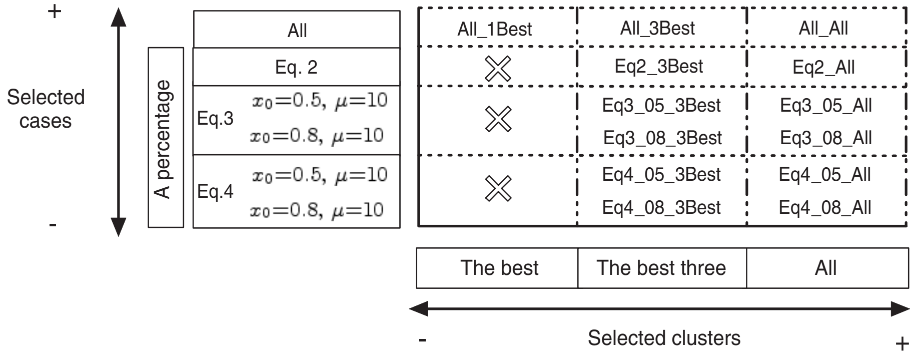


Fig. 7. Test experiments. The configurations marked with a cross has not been tested because they behave like *All_1Best*.

with a noticeable reduction in the number of operations). Even so, we note a set of strategies with values of medium rank inside the limit marked by CD, two of which have an important reduction in the number of operations while the accuracy rate is maintained: *Eq3_05_3Best* and *All_3Best*. Although the strategies *Eq3_05_All* and *Eq3_08_All* have a similar accuracy rate like *All_All*, they do not provide a significant computational time improvement.

Fig. 8(b) represents datasets with a complexity of type B, which has higher complexity than type A. The increase of the complexity entails two effects: (1) the number of operations is reduced in most strategies and (2) the linearity between the two variables is also decreased (the correlation coefficient is now 0.86). Similarly as before, *All_3Best* is the most suitable strategy because it maintains the accuracy rate while the computational is reduced. *Eq3_05_All* and *Eq3_08_All* works like *All_All* again.

Finally, Fig. 8(c) refers to datasets of the highest complexity (type C). In this case, the complexity accentuates the previous effects: (1) the mean number of operations continues to be reduced and (2) the linear correlation between both variables is even less than before (coefficient in 0.76). Although the strategies *Eq3_05_All* and *Eq3_08_All* continue without improving the computational time, they improve the accuracy rate of the *All_All* configuration. The strategy *All_3Best* improves the computational time and the accuracy rate, while the strategies *Eq3_05_3Best*, *Eq4_05_All* improve only the computational time and maintain the accuracy.

The analysis of SOMCBR using the proposed methodology can be summarized in the following aspects: (1) SOM is a suitable clustering technique for organizing the case memory because it is able to successfully index it. (2) SOM works best in complex domains. This idea corroborates a previous work of ours [8]. (3) The best configurations are those in which the retrieve phase uses all the cases from more than one cluster, or it uses a weighting percentage of cases from all the clusters. We understand such a good configuration those in which the computational time is improved and the accuracy rate is maintained. Notwithstanding, the rest of

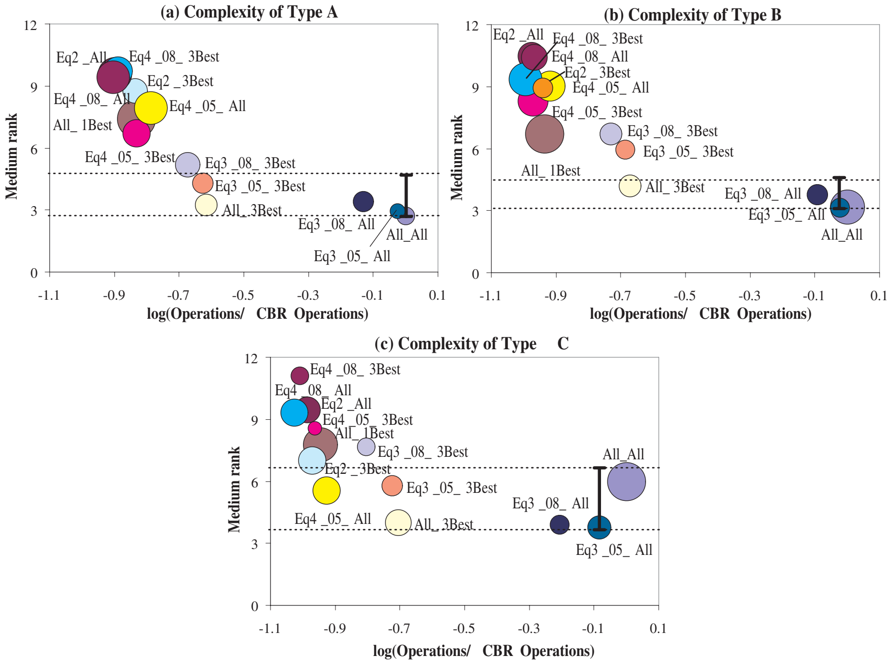


Fig. 8. Analysis of the case retrieval strategies according to the complexity types (A, B, and C)

configurations improve greatly the computational time because they use few cases, but this has a negative influence on the accuracy rate. The final selection of the strategy will depend on the user requirements.

5 Conclusions and Further Research

In this paper, we have presented a methodology for analyzing the behavior of the different ways in which the case retrieval from a clustered case memory can be performed while taking into account the performance expected by the user. The performance is a balance between the desired computational time improvement and the maximum acceptable reduction of the accuracy rate. Additionally, we have offered an innovative and intuitive way for analyzing the performance of the case retrieval strategies over a large set of datasets.

The proposed methodology is divided into three steps. The first step consists in running over all the possible case retrieval strategies from the clustered case memory. All the configurations are extracted from a previously taxonomy of several case retrieval ways. The taxonomy considers the number of clusters and cases used. Next, the datasets are split according to the three levels of complexity (A, B, or C) using the complexity measures N1, N2, and F3. Finally, the scatter plot is drawn for each one of the complexity types. The graphical representation compares the average rank with respect to the computational time improvement. These steps are applicable for any case memory organized by a clustering technique.

This methodology has been successfully tested using SOMCBR, which is a CBR system characterized by organizing the case memory by means of the SOM approach. The main conclusions of the analysis are that SOMCBR works better in complex domains, and that the best solution (improving the computational time while maintaining the accuracy rate) is often to use all the cases from more than one cluster or a part of cases from all the clusters. Anyway, the performed desired depends on the final user requirements: more speed, less accuracy.

The further work is focused on applying this methodology over other case memory organizations based on clusters, and trying to define a meta-relation level between the case memory organizations.

Acknowledgments

We would like to thank the Spanish Government for the support under grants TIN2006-15140-C03-03, TIN2005-08386-C05-04, and the *Generalitat de Catalunya* for the support under grants 2005SGR-302 and 2006FIC-0043. Also, we would like to thank *Enginyeria i Arquitectura La Salle* of Ramon Llull University for the support to our research group.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundations issues, methodological variations, and system approaches. AI Communications 7, 39–59 (1994)
2. Basu, M., Ho, T.K.: Data Complexity in Pattern Recognition. In: Advanced Information and Knowledge Processing, Springer, Heidelberg (2006)
3. Bernadó, E., Ho, T.K.: Domain of competence of XCS classifier system in complexity measurement space. IEEE Transaction Evolutionary Computation 9(1), 82–104 (2005)
4. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)
5. Brown, M.: A Memory Model for Case Retrieval by Activation Passing. PhD thesis, University of Manchester (1994)
6. Chang, P., Lai, C.: A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. Expert Syst. Appl. 29(1), 183–192 (2005)
7. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
8. Fornells, A., Golobardes, E., Martorell, J.M., Garrell, J.M., Bernadó, E., Macià, N.: Measuring the applicability of self-organization maps in a case-based reasoning system. In: 3rd Iberian Conference on Pattern Recognition and Image Analysis. LNCS, vol. 4478, pp. 532–539. Springer, Heidelberg (2007)
9. Fornells, A., Golobardes, E., Vernet, D., Corral, G.: Unsupervised case memory organization: Analysing computational time and soft computing capabilities. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 241–255. Springer, Heidelberg (2006)

10. Fornells, A., Golobardes, E., Vilasís, X., Martí, J.: Integration of strategies based on relevance feedback into a tool for retrieval of mammographic images. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) IDEAL 2006. LNCS, vol. 4224, pp. 116–124. Springer, Heidelberg (2006) (Selected to be published in the International Journal of Neural Systems)
11. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(3), 289–300 (2002)
12. Kohonen, T.: Self-Organization and Associative Memory, 3rd edn. Springer Series in Information Sciences, vol. 8. Springer, Heidelberg (1984)
13. Lenz, M., Burkhard, H.D., Brückner, S.: Applying case retrieval nets to diagnostic tasks in technical domains. In: Smith, I., Faltings, B.V. (eds.) Advances in Case-Based Reasoning. LNCS, vol. 1168, pp. 219–233. Springer, Heidelberg (1996)
14. Myllymaki, P., Tirri, H.: Massively parallel case-based reasoning with probabilistic similarity metrics (1993)
15. Nicholson, R., Bridge, D., Wilson, N.: Decision diagrams: Fast and flexible support for case retrieval and recommendation. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 136–150. Springer, Heidelberg (2006)
16. Pelleg, D., Moore, A.: X-means: Extending K-means with efficient estimation of the number of clusters. In: Proceedings of the 17th International Conference of Machine Learning, pp. 727–734. Morgan Kaufmann, San Francisco (2000)
17. Plaza, E., McGinty, L.: Distributed case-based reasoning. The Knowledge engineering review 20(3), 261–265 (2006)
18. Rissland, E.L., Skalak, D.B., Friedman, M.: Case retrieval through multiple indexing and heuristic search. In: International Joint Conferences on Artificial Intelligence, pp. 902–908 (1993)
19. Schaaf, J.W.: Fish and Sink - an anytime-algorithm to retrieve adequate cases. In: Aamodt, A., Veloso, M.M. (eds.) Case-Based Reasoning Research and Development. LNCS, vol. 1010, pp. 538–547. Springer, Heidelberg (1995)
20. Vernet, D., Golobardes, E.: An unsupervised learning approach for case-based classifier systems. Expert Update. The Specialist Group on Artificial Intelligence 6(2), 37–42 (2003)
21. Wess, S., Althoff, K.D., Derwand, G.: Using k-d trees to improve the retrieval step in case-based reasoning. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) Topics in Case-Based Reasoning. LNCS, vol. 837, pp. 167–181. Springer, Heidelberg (1994)
22. Yang, Q., Wu, J.: Enhancing the effectiveness of interactive cas-based reasoning with clustering and decision forests. Applied Intelligence 14(1) (2001)