Abstract

Case memory maintenance in a Case-Based Reasoning system is important for two main reasons: (1) to control the case memory size; (2) to reduce irrelevant and redundant instances that may produce inconsistencies in the Case-Based Reasoning system. In this paper we present two approaches based on deletion policies to the maintenance of case memories. The foundations of both approaches are the Rough Sets Theory, but each one applies a different policy to delete or maintain cases. The main purpose of these methods is to maintain the competence of the system and reduce, as much as possible, the size of the case memory. Experiments using different domains, most of them from the UCI repository, show that the reduction techniques maintain the competence obtained by the original case memory. The results obtained are compared with those obtained using well-known reduction techniques.

Introduction and Motivation

Case-Based Reasoning (CBR) systems solve problems by reusing the solutions to similar problems stored as cases in a case memory (Riesbeck & Schank 1989) (also known as case-base). However, these systems are sensitive to the cases present in the case memory and often its good competence depends on the significance of the cases stored. The aim of this paper is twofold: (1) to remove noisy cases and (2) to achieve a good generalisation accuracy. This paper presents two hybrid deletion techniques based on Rough Sets Theory. In a previous paper, we presented two reduction techniques based on these measures (Salamó & Golobardes 2001). This paper continues the initial approaches presented in the previous one, defining a competence model based on Rough sets and presenting new hybrid approaches to improve the weak points. The conclusion of the previous work was that the proposed reduction techniques were complementary, so hybrid methods will achieve a higher reduction and better competence case memories. Thus, in this paper, we present two hybrid approaches: Accuracy-Classification Case Memory (ACCM) and Negative Accuracy-Classification Case Memory (NACCM). Both reduction techniques have been introduced into our Case-Based Classifier System called BASTIAN.

The paper is organized as follows. Next section introduces related work. In the next section, we explain the foundations of Rough Sets Theory used in our reduction techniques. After this section we detail the proposed reduction techniques based on deletion policies, continuing in next section describing the testbed of the experiments and the results obtained. Finally, in the last section, we present the conclusions and further work.

Related Work

Many researchers have addressed the problem of case memory reduction (Wilson & Martinez 2000; Wilson & Leake 2001) and different approaches have been proposed. One kind of approaches are related to Instance Based Learning algorithms (IBL) (Aha & Kibler 1991). Another approach to instance pruning systems are those that take into account the order in which instances are removed (DROP1 to DROP5)(Wilson & Martinez 2000). The most similar methods to our approaches, some of them inspired us, are those focused on limiting the overall competence loss of the case memory through case deletion. Where competence is the range of target problems that can be successfully solved (Smyth & Keane 1995). Strategies have been developed for controlling case memory growth. Several methods such as competence-preserving deletion (Smyth & Keane 1995) and failure-driven deletion (Portinale, Torasso, & Tavano 1999), as well as for generating compact case memories through competence-based case addition (Smyth & McKenna 1999). Leake and Wilson (Leake & Wilson 2000) examine the benefits of using fine-grained performance metrics to directly guide case addition or deletion. These methods are specially important for task domains with non-uniform problem distributions. The maintenance integrated with the overall CBR process was presented in (Reinartz & Iglezakis 2001).

Rough Sets theory

The rough sets theory defined by Pawlak, which is well detailed in (Pawlak 1982), is one of the techniques for the identification and recognition of common patterns in data, especially in the case of uncertain and incomplete data. The mathematical foundations of this method are based on the set approximation of the classification space.

Within the framework of rough sets the term classification describes the subdivision of the universal set of all possible
categories into a number of distinguishable categories called elementary sets. Each elementary set can be regarded as a rule describing the object of the classification. Each object is then classified using the elementary set of features which can not be split up any further, although other elementary sets of features may exist. In the rough set model the classification knowledge (the model of the data) is represented by an equivalence relation $\text{IND}$ defined on a certain universe of objects (cases) $U$ and relations (attributes) $R$. $\text{IND}$ defines a partition on $U$. The pair of the universe objects $U$ and the associated equivalence relation $\text{IND}$ forms an approximation space. The approximation space gives an approximate description of any subset $X$ of $U$. Two approximations are generated by the available data about the elements of the set $X$, called the lower and upper approximations (see figure 1). The lower approximation $\underline{R}(X)$ is the set of all elements of $U$ which can certainly be classified as elements of $X$ in knowledge $R$. The upper approximation $\overline{R}(X)$ is the set of elements of $U$ which can possibly be classified as elements of $X$, employing knowledge $R$.

![Figure 1: The lower and upper approximations of a set X.](image)

In order to discover patterns of data we should look for similarities and differences of values of the relation $R$. So we have to search for combinations of attributes with which we can discern objects and object classes from each other. The minimal set of attributes that forms such a combination is called a reduct. Reducts are the most concise way in which we can discern objects classes and which suffices to define all the concepts occurring in the knowledge.

**Measures of relevance based on Rough Sets**

The reduced space, composed by the set of reducts ($P$) and core, is used to extract the relevance of each case.

**Definition 1 (Accuracy Rough Sets)**

This measure computes the Accuracy coefficient ($\text{AccurCoef}$) of each case $t$ in the knowledge base (case memory) $T$ as:

\[
\text{AccurCoef}(t) = \frac{\text{card} (P(t))}{\text{card} (\overline{P}(t))}
\]  

(1)

Where $\text{AccurCoef}(t)$ is the relevance of the instance $t$; $T$ is the training set; $\text{card}$ is the cardinality of a set; $P$ is a set that contains the reducts; and finally $P(t)$ and $\overline{P}(t)$ are the presence of $t$ in the lower and upper approximations, respectively.

The accuracy measure expresses the degree of completeness of our knowledge about the set $P$. It is the percentage of possible correct decisions when classifying cases employing $t$. We use the accuracy coefficient to explain if an instance $t$ is on an internal region or on an outlier region. The values of the measure when there exists only one case $t$ as input is limited to $\{0,1\}$. When the value is 0.0 it means an internal case, and a value of 1.0 means an outlier case. Inexactness of a set of cases is due to the existence of a borderline region. The greater a borderline region of a set (greater $P$), the lower the accuracy of the set.

**Definition 2 (Class Rough Sets)**

In this measure we use the quality of classification coefficient (ClassCoef). It is computed as:

\[
\mu(t) = \frac{\text{card} (P(t)) \cup \text{card} (P(\neg t))}{\text{card} (\text{all instances})}
\]  

(2)

Where $\text{ClassCoef}(t)$ is the relevance of the instance $t$; $T$ is the training set; $\neg t$ is $T - t$ set; $\text{card}$ is the cardinality of a set; $P$ is a set that contains the reducts; and finally $P(t)$ is the presence of $t$ in the lower approximation.

The $\text{ClassCoef}$ coefficient expresses the percentage of cases which can be correctly classified employing the knowledge $t$. This coefficient has a range of real values in the interval $[0.0, 1.0]$. Where 0.0 and 1.0 mean that the instance classifies incorrectly and correctly respectively, the range of cases that belong to its class. The higher the quality, the nearer to the outlier region.

**Reduction Techniques**

This section presents two hybrid reduction techniques based on the Rough Sets measures described in the previous section. The difference between them is to facilitate the usage of the coverage when selecting the cases that are deleted from the original case memory.

**Categorisation model of case memory**

The aim of these reduction techniques is to take advantage of the benefits of each coverage measure ($\text{AccurCoef}$ and $\text{ClassCoef}$). In order to make understanding the algorithms and the environment of application easier, we introduce different concepts and definitions.

We use these techniques on classification tasks. For this reason, we modify some definitions. The distribution of the case memory is done using a new categorisation in terms of their coverage and reachability. The coverage and reachability concepts are modified with regard to (Smyth & Keane 1995). However, we maintain as far as possible the essence of the original ones, but it is modified to our coverage measures (explained previously) and to our problem task.

**Definition 3 (Coverage)**

Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances, $\forall t_i \in T$;

\[
\text{Coverage}(t_i) = \text{AccurCoef}(t_i) \oplus \text{ClassCoef}(t_i)
\]
The $\oplus$ operation is the logical sum of both values. When $\text{AccurCoef}$ value is 1, the $\text{Coverage}$ is 1.0 but when it is 0, the $\text{Coverage}$ is $\text{ClassCoef}$ value.

**Definition 4 (Reachability)**

Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances and $C_i$ be a classification task, $\forall i \in T$:

$$\text{Reachability}(t_i) = \begin{cases} 
\text{Class}(t_i) & \text{if it is a } C_i \\
\text{Adaptable}(t', t_i) & \text{if it is not a } C_i
\end{cases}$$

Where $\text{class}(t_i)$ is the class that classifies case $t_i$ and $t' \in T$.

**Accuracy-Classification Case Memory (ACCM)**

The main idea of this reduction technique is to benefit from the advantages of both measures separately. Firstly, it maintains all the cases that are outliers, so cases with an $\text{Coverage}$ value are not removed. This assumption is made because if a case is isolated, there is no other case that can solve it. Secondly, the cases selected are those that are nearest to the outliers and other cases nearby can be used to solve it because their coverage is higher.

Thus, the selection of cases starts from internal cases to outlier ones. However, this algorithm maintains the selected cases. The aim is to maintain the minimal set of cases in the case memory. The behaviour of this reduction technique is similar to ACCM because it removes also cases near the outlier region but NACCM allows fewer cases to be maintained, thus obtaining a greater reduction.

**Algorithm 1 ACCM**

1. SelectCasesACCM (CaseMemory $T$)
2. $\text{confidenceLevel} = 1.0$ and $\text{freeLevel} = \text{ConstantTuned}$ (set at 0.01)
3. select all instances $t \in T$ as $\text{SelectCase}(t)$ if $t$ satisfies: $\text{coverage}(t) \geq \text{confidenceLevel}$
4. while not $\exists$ at least a $t$ in $\text{SelectCase}$ for each class $c$ that $\text{reachability}(t) = c$
5. $\text{confidenceLevel} = \text{confidenceLevel} - \text{freeLevel}$
6. select all instances $t \in T$ as $\text{SelectCase}(t)$ if $t$ satisfies: $\text{coverage}(t) \geq \text{confidenceLevel}$
7. end while
8. Delete from CaseMemory the set of cases selected as $\text{SelectCase}$
9. return CaseMemory $T$

**Algorithm 2 NACCM**

1. SelectCasesNACCM (CaseMemory $T$)
2. $\text{confidenceLevel} = 1.0$ and $\text{freeLevel} = \text{ConstantTuned}$ (set at 0.01)
3. select all instances $t \in T$ as $\text{SelectCase}(t)$ if $t$ satisfies: $\text{coverage}(t) \geq \text{confidenceLevel}$
4. while not $\exists$ at least a $t$ in $\text{SelectCase}$ for each class $c$ that $\text{reachability}(t) = c$
5. $\text{confidenceLevel} = \text{confidenceLevel} - \text{freeLevel}$
6. select all instances $t \in T$ as $\text{SelectCase}(t)$ if $t$ satisfies: $\text{coverage}(t) \geq \text{confidenceLevel}$
7. end while
8. Maintain in CaseMemory the set of cases selected as $\text{SelectCase}$, those cases not selected are deleted from CaseMemory
9. return CaseMemory $T$

This section describes the testbed used in the experimental study and discuss the results obtained from our reduction techniques. Finally, we also compare our results with some related reduction techniques.

**Testbed**

In order to evaluate the performance rate, we use ten datasets. Datasets can be grouped in two ways: public and private (details in table 1). Public datasets are obtained from the UCI repository (Merz & Murphy 1998). They are: Breast Cancer Wisconsin (Breast-w), Glass, Ionosphere, Iris, Sonar and Vehicle. Private datasets (Golobardes et al. 2002) come from our own repository. They deal with diagnosis of breast cancer (Biopsy and Mammogram). Synthetic datasets (MX11) is the eleven input multiplexer and...
The study described in this paper was carried out in the context of BASTIAN, a case-Based System in Classification. All techniques were run using the same set of parameters for all datasets: a 1-Nearest Neighbour Algorithm that uses a list of cases to represent the case memory. Each case contains the set of attributes, the class, the AccurCoef and ClassCoef coefficients. Our goal in this paper is to reduce the case memory. For this reason, we have not focused on the representation used by the system. The retain phase does not store any new case in the case memory. Finally, weighting methods are not used in this paper in order to test the reliability of our reduction techniques. Further work will consist of testing the influence of these methods in conjunction with weighting methods.

Experimental analysis of reduction techniques

The aim of our reduction techniques is to reduce the case memory while maintaining the competence of the system. This priority guides our deletion policies. That fact is detected in the results. Table 2 shows the results for the IBL's algorithms and the Rough Sets reduction techniques. For example, the Vehicle dataset obtains a good competence as well as reducing the case memory, in both reduction techniques. The results related to ACCM show competence maintenance and improvement in some datasets, but the case memory size has not been reduced too much. These results show that ACCM is able to remove inconsistency and redundant cases from the case memory, enabling to be improved the competence. The NACCM technique shows, as expected in its description due to a more restrictive behaviour, a higher reduction of the case memory. However, the reduction in NACCM is not very large. The behaviour is similar to ACCM. This is due to the fact that both reduction techniques share the same foundations. The NACCM obtains higher reduction while producing a competence loss, although it is not a significant loss.

In summary, the results obtained using ACCM and NACCM maintain or even improve in a significance level the competence while reducing the case memory.

Comparing rough sets reduction techniques to IBL, ACCM and NACCM obtain on average a higher generalisation on competence than IBL, as can be seen in table 2. The performance of IBL algorithms declines, in almost all datasets (e.g. Breast-w, Biopsy), when case memory is reduced. CBR obtains on average higher prediction competence than IB2, IB3 and IB4. On the other hand, the mean storage size obtained is higher in our reduction techniques than those obtained using IBL schemes.

To finish the empirical study, we also run additional well-known reduction schemes on the previous data sets. Table 3 compares ACCM to CNN, SNN, DEL, ENN, RENN. Table 4 compares ACCM to DROPI, DROP2, DROP3, DROP4 and DROP5 (a complete explanation of them can be found in (Wilson & Martinez 2000)). We use the same datasets described above but with different ten-fold cross validation sets.

Table 3 shows that the results of ACCM are on average better than those obtained by the reduction techniques studied. RENN improves the results of ACCM in some data sets (e.g. Breast-w) but its reduction on the case memory is lower than ACCM.
data sets (e.g. Biopsy, Breast-w, Ionosphere, Sonar), although its results are also worse in others (e.g. Mx11). The behaviour of these reduction techniques are similar to those previously studied. ACCM obtains a balanced behaviour between competence and size. There are some reduction techniques that obtain best competence for some data sets while reducing less the case memory size.

All the experiments (tables 2, 3 and 4) point to some interesting observations. First of all, it is worth noting that the individual ACCM and NACCM work well in all data sets, obtaining better results on ACCM because its deletion policy is more competitive. Secondly, the mean storage obtained using ACCM and NACCM is reduced while maintaining the competence on the CBR system. Finally, the results in all tables suggest that all the reduction techniques work well in some, but not all, domains. This has been termed the selective superiority problem (Brodley 1993). Consequently, future work consists of improving the selection of cases in order to be eliminated or maintained in the case memory while maintaining, as well as ACCM and NACCM techniques, the CBR competence.

Table 4: Comparing ACCM reduction technique to DROP algorithms (Wilson & Martinez 2000).

<table>
<thead>
<tr>
<th>Method</th>
<th>ACCM</th>
<th>DROP1</th>
<th>DROP2</th>
<th>DROP3</th>
<th>DROP4</th>
<th>DROP5</th>
</tr>
</thead>
<tbody>
<tr>
<td>%PA</td>
<td>%CM</td>
<td>%PA</td>
<td>%CM</td>
<td>%PA</td>
<td>%CM</td>
<td>%PA</td>
</tr>
<tr>
<td>BC</td>
<td>83.65</td>
<td>58.01</td>
<td>76.95</td>
<td>28.34</td>
<td>73.94</td>
<td>20.30</td>
</tr>
<tr>
<td>BR</td>
<td>95.71</td>
<td>73.16</td>
<td>93.28</td>
<td>8.79</td>
<td>92.56a</td>
<td>8.35</td>
</tr>
<tr>
<td>GZ</td>
<td>69.83</td>
<td>74.95</td>
<td>66.59</td>
<td>40.86</td>
<td>69.57</td>
<td>42.94</td>
</tr>
<tr>
<td>B0</td>
<td>90.59</td>
<td>83.77</td>
<td>81.20</td>
<td>23.04</td>
<td>87.74</td>
<td>19.21</td>
</tr>
<tr>
<td>BR</td>
<td>96.66</td>
<td>89.03</td>
<td>91.33</td>
<td>12.44</td>
<td>90.04a</td>
<td>14.07</td>
</tr>
<tr>
<td>MA</td>
<td>66.34</td>
<td>91.19</td>
<td>61.60</td>
<td>42.69</td>
<td>58.31a</td>
<td>31.34</td>
</tr>
<tr>
<td>MX</td>
<td>78.61</td>
<td>99.97</td>
<td>80.46</td>
<td>19.02</td>
<td>90.00a</td>
<td>98.17</td>
</tr>
<tr>
<td>SO</td>
<td>86.45</td>
<td>77.31</td>
<td>84.44</td>
<td>25.05</td>
<td>87.07</td>
<td>28.26</td>
</tr>
<tr>
<td>TG</td>
<td>96.13</td>
<td>97.59</td>
<td>94.70</td>
<td>3.83</td>
<td>95.23</td>
<td>8.39</td>
</tr>
<tr>
<td>YE</td>
<td>69.10</td>
<td>72.35</td>
<td>64.64</td>
<td>38.69</td>
<td>67.16</td>
<td>45.21</td>
</tr>
</tbody>
</table>

Conclusions and Further Work

This paper presents two reduction techniques whose foundations are the Rough Sets Theory. The aim of this paper is twofold: (1) to avoid inconsistent and redundant instances and to obtain compact case memories; and (2) to maintain or improve the competence of the CBR system. Empirical study shows that these reduction techniques produce compact competent case memories. Although the case memory reduction is not large, the competence of the CBR system is improved or maintained on average. Thus, the generalisation accuracy on classification tasks is guaranteed.

We conclude that the deletion policies could be improved in some points which our further work will be focus on. Firstly, we can modify the competence model presented in this paper to assure a higher reduction on the case memory. Secondly, it is necessary to study the influence of the learning process. Finally, we want to analyse the influence of the weighting methods in these reduction techniques.

Acknowledgements

This work is supported by the Ministerio de Ciencia y Tecnología, Grant No. TIC2002-04160-C02-02. This work is also supported by Generalitat de Catalunya by grant No. 2002SGR 00155 for our Consolidate Research Group in Intelligent Systems. We wish to thank Enginyeria i Arquitectura La Salle for their support of our research group. We also wish to thank: D.W. Aha for providing the IBL code and D. Randall Wilson and Tony R. Martinez who provided the code of the other reduction techniques.

References


