Global, Local and Mixed Rough Sets Case Base Maintenance Techniques

Maria Salamó and Elisabet Golobardes
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull
Quatre Camins 2, 08022 Barcelona, Spain
{*mariasal,elisabet*}*l@salleurl.edu*

Abstract.
A major key to success for case-based reasoning systems widely recognized are the definition of suitable case base maintenance policies. The aim of this paper is two fold: (1) to analyse global versus local case base maintenance policies and (2) to develop a unified technique between local and global policies. All the case base maintenance techniques studied in this paper share a common foundation: the Rough Sets theory and its measures of coverage integrated in a Rough Sets competence model. The main purpose of all Rough Sets case base maintenance techniques is to maintain the competence and reduce, as much as possible, its size. Several experiments, using different domains from the UCI and from our repositories, denote that the relation between local and global are quite similar and the unified approach can lead to successful systems in some domains while maintains the competence in other domains.

1 Introduction

Case-Based Reasoning (CBR) is the process of reasoning and learning by storing prior *cases* −records of specific prior reasoning episodes− and retrieving and adapting them to aid new problem solving or interpretation in similar situations ([3],[9]). A major key to success of large-scale and long-term case base reasoning application systems widely recognised are the definition of suitable maintenance techniques.

The aim of this paper is two fold: (1) to analyse the behaviour of global and local maintenance techniques, and (2) to present a unified approach between global and local techniques. All the case base maintenance techniques presented and proposed in this paper are based on our Rough Sets competence model [12]. The main issue of all the case base maintenance techniques is to reduce the case memory, as much as possible, while maintaining and improving -if possible- the competence of the system. The analysis of our global and local approaches promotes the development of a mixed approach between them. Our motivation on the mixed approach was mainly produced by the possibility to obtain a more reduced case memory, but without losing our main proposal, to reduce the case base while maintaining −at least− its competence.

Rough Sets case base maintenance approaches have been introduced into our Case-Based Classifier System called BASTIAN [14]. This paper continues the initial Rough Sets approaches presented in our previous work [11, 12]. The global reduction technique analysed is named *Negative Accuracy-Classification Case Memory (NACCM)*. The local approach studied and improved is named Sort Out Internal Case Memory (SortOutInternalCM).

The paper is organised as follows. Section 2 introduces some relevant related work. Next, section 3 describes the foundations of the Rough Sets Theory used in our reduction techniques. Then, section 4 details the Rough Sets case base maintenance techniques based on the Rough Sets competence model. Section 5 describes the test suite of the experiments and the results obtained. Finally, section 6 presents some conclusions and further work.

2 Related Work

Many researchers have addressed the problem of case memory reduction [19, 18] and different approaches have been proposed. One kind of approaches are related to Instance Based Learning algorithms (IBL) [1].

Another kind of approaches have focused researchers on increasing the overall competence, *the range of target problems that can be successfully solved* [15], of the case memory through case deletion. Strategies have been developed for controlling case memory growth. Several methods such as competence-preserving deletion [15] and failure-driven deletion [7], as well as for generating compact case memories through competence-based case addition [16, 21, 17]. Leake and Wilson [4] 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 case-based reasoning process was presented in [8]. Finally, a case-base maintenance method that avoids building sophisticated structures around a case-base or complex operations is presented by Yang and Wu [20]. Their method partitions cases into clusters that can be converted to new smaller case-bases.

3 Rough Sets Theory

Zdzislaw Pawlak introduced Rough Sets theory in 1982 [6]. We use Rough Sets theory for extracting the dependencies of knowledge. These dependencies are the basis for computing the relevance of features and instances into the Case-Based Classifier System.

3.1 Introduction to Rough Sets Theory

We have a **Universe** (U) (finite not null set of cases that describes our problem, i.e. the case memory). We compute from our universe the **concepts** (cases) that form partitions. The union of all the *concepts* make the entire Universe. Using *all the concepts* we can describe all the **equivalence relations** (R) over the universe U. Let an equivalence relation be a *set of features* that describe a specific concept. U/R is the family of all **equivalence classes** of R. The universe and the relations form the **knowledge base** (K), defined as $K =< U, \hat{R} >$, where $\hat{R}$ is the family of equivalence relations over U. Every relation over the universe is an elementary concept in K. All the concepts are formed by a set of equivalence relations that describe them. Thus, the goal is to search for the minimal set of R that defines the same concept as the initial set.

Definition (Indiscernibility Relations) $IND(\hat{P}) = \bigcap \hat{R}$ where $\hat{P} \subseteq \hat{R}$. The indiscernibility relation is an equivalence relation over U. Hence, it partitions the concepts (cases) into equivalence classes. These sets of classes are sets of instances indiscernible with respect to

the features in P. Such a partition is denoted as $U/IND(P)$. In supervised machine learning the sets of cases indiscernible, with respect to the class attribute, contain the cases of each class.

Approximations of Set. The idea of Rough Sets relies on the approximation of a set by a pair of sets. These sets are known as the lower and the upper approximation. These approximations are generated by the available data about the elements of the set.

Let $K =< U, \hat{R} >$ be a knowledge base. For any subset of cases $X \subseteq U$ and an equivalence relation $R \in \hat{R}$, $R \subseteq IND(K)$ we associate two subsets called: Lower $\underline{R}X$; and Upper $\overline{R}X$ approximations. If $\underline{R}X$=$\overline{R}X$ then X is an *exact set* (definable using subset R), otherwise X is a ***rough set*** with respect to R.

Definition (Lower approximation) The lower approximation, defined as: $\underline{R}X = \bigcup\{Y \in U/R : Y \subseteq X\}$ is the set of all elements of U which can *certainly* be classified as elements of X in knowledge R.

Definition (Upper approximation) The upper approximation, $\overline{R}X = \bigcup\{Y \in U/R : X \bigcap Y \neq \emptyset\}$ is the set of elements of U which can *possibly* be classified as elements of X, employing knowledge R.

Reduct and Core of knowledge This part is related to the concept of reduction of the feature search space that defines the initial knowledge base. Next, this reduced space is used to extract the relevance of each case. Intuitively, a **reduct** of knowledge is its essential part which suffices to define all concepts occurring in the knowledge, whereas the **core** is the most important part.

Let $\hat{R}$ be a family of equivalence relations and $R \in \hat{R}$. We will say that:

- R is *indispensable* if $IND(\hat{R}) \neq IND(\hat{R} - \{R\})$; otherwise it is *dispensable*. $IND(\hat{R} - \{R\})$ is the family of equivalence $\hat{R}$ extracting R.
- The family $\hat{R}$ is *independent* if each $R \in \hat{R}$ is *indispensable* in R; otherwise it is *dependent*.

Definition (Reduct) $\hat{Q} \in \hat{R}$ is a reduct of $\hat{R}$ if : $\hat{Q}$ is *independent* and $IND(\hat{Q}) = IND(\hat{R})$. Obviously, $\hat{R}$ may have many reducts. Using $\hat{Q}$ it is possible to approximate the same concept as using $\hat{R}$. Each reduct has the property that a feature can not be removed from it without changing the indiscernibility relation.

Definition (Core) The set of all indispensable relations in $\hat{R}$ will be called the *core* of $\hat{R}$, and will be denoted as: $CORE(\hat{R}) = \bigcap RED(\hat{R})$. Where $RED(\hat{R})$ is the family of all reducts of $\hat{R}$. It is the most characteristic part of knowledge and can not be eliminated.

3.2 Measures of relevance based on Rough Sets

AccurCoef and *ClassCoef* measures use the information of reducts and the core to compute the relevance of each case.

Definition (AccurCoef) This measure computes the *Accuracy* coefficient (**AccurCoef**) of each case t in the knowledge base (case memory T) as:

$$For\ each\ instance\ t\ \in\ T\ it\ computes:$$
$$AccurCoef(t) = \frac{card\ (\ \underline{P}(t))}{card\ (\ \overline{P}\ (t))} \tag{1}$$

Where $AccurCoef(t)$ is the relevance of the instance t; T is the training set; $card$ is the cardinality of one set; P is the set that contains the *reducts* and *core* obtained from the original data; and finally $\underline{P}(t)$ 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. The accuracy coefficient explains if an instance is on an internal region or on a border line region, thus $AccurCoef(t)$ is a binary value. When the value is 0 it means an internal case, and a value of 1 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, the lower the accuracy of the set. The accuracy expresses the percentage of possible correct decisions made when classifying cases employing knowledge P.

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

$$For\ each\ instance\ t\ \in\ T\ it\ computes:$$
$$\mu(t) = \frac{card\ (\ \underline{P}(t))\ \cup\ card\ (\ \underline{P}(-t))}{card\ (\ all\ instances)} \tag{2}$$

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

The $ClassCoef$ coefficient expresses the percentage of cases which can be correctly classified employing the knowledge t. This coefficient has a range of values between 0 to 1, where 0 and 1 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.

4 Global, Local and Mixed approaches for CBM

This section presents the competence model and the local and global case base maintenance techniques analyzed. Finally, it introduces the mixed approach between local and global techniques. All these reduction techniques are based on the Rough Sets measures described along this section, but their basis are described in section 3.2.

4.1 Rough Sets Competence Model

First of all, we present the key concepts in categorising the cases in the sort out case memory model (see figure 1). The *coverage* and *reachability* concepts are modified, for our *coverage* coefficients and to our problem task, with regard to B. Smyth and M. Keane [15]. However, we maintain as far as possible the essence of the original ones. The *coverage* is computed using the Rough Sets coefficients. On the other hand, the *reachability* in this case is adapted to classification tasks.

Definition (Coverage) Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances, $\forall\, t_i \in T$: $Coverage(t_i)$= $AccurCoef(t_i) \oplus ClassCoef(t_i)$

The $\oplus$ operation is the logical sum of both values. The $coverage$ of a case is the accuracy and quality when it is used to solve a target problem.

Definition (Reachability) Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances, $\forall\, t_i \in T$:

$$Reachability(t_i) = \begin{cases} class\ (t_i) & if\ it\ is\ a\ classification\ task \\ adaptable(t_i, T) & if\ it\ is\ not\ a\ classification\ task \end{cases} \quad (3)$$

The original definition is maintained and extended to classification tasks. The $reachability$ of a target problem is the set of cases that can be used to provide its solution.

Definition (Coverage group]) Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances and let S be a subset of instances where $S \in T$. For all instances i and j in S: $CoverageGroup(S)$ = $Coverage(i) = Coverage(j)$

A coverage group (see figure 1) is a set of cases from the case memory where all the cases have the same *coverage* without taking into account the class of each case. The coverage group shows space regions of our knowledge. The bigger a coverage group, the higher outlier the set of cases. The lower the coverage group, the higher an internal set of cases.

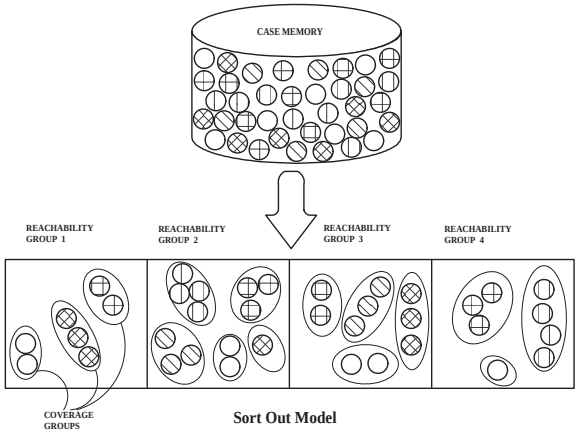

Figure 1: Sort Out Case Memory

Definition (Reachability group) Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances and let S be a subset of instances where $S \in T$. For all instances i and j in S: $ReachabilityGroup(S)$ = $Reachability(i) = Reachability(j)$

A reachability group (see figure 1) is the set of instances that can be used to provide a solution for the target. The reachability group produce the sort out of the case memory. However, a reachability group can contain different coverage groups. Every coverage group shows the levels of information (border line regions) in the reachability group.

Definition (Master case) Let $S = \{s_1, s_2, ..., s_n\}$ and $T = \{t_1, t_2, ..., t_n\}$ be two sets of instances, where $S \in T$. For each $CoverageGroup(s) \in ReachabilityGroup(S)$ we have a: $MasterCase(t)$ = A selected case t from $ReachabilityGroup(S) \wedge CoverageGroup(s)$

Each coverage group contains a master case. Thus each reachability group contains as many master cases as coverage groups. The master cases will depend on the selection policies we use in our reduction techniques. These will be explained in the following sections.

4.2 Global approach named Negative Accuracy-Classification Case Memory

The global approach treats all the cases that belong to the case memory at the same time, independently from the knowledge included in each case. The *Negative Accuracy-Classification Case Memory* (NACCM) is an algorithm that combines $AccurCoef$ and $ClassCoef$ measures to decide which cases have to be maintained in the case memory. The main idea of this reduction technique is to take benefit from the advantages obtained when applied both measures separately. There is also another possibility to combine both measures, it is an algorithm called *Accuracy-Classification Case Memory* (ACCM). We decided to use NACCM algorithm because it uses in its foundations the same concepts of ACCM algorithm and it also obtains a higher reduction of the case memory. In a graphical manner, the process is represented in figure 2, where it can be seen that the NACCM algorithm is based on ACCM, doing the complementary process. An extended explanation of it can be found in [13].

NACCM

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

The motivation for NACCM algorithm is to select a wider range of cases than ACCM algorithm. The main process in ACCM is to select all the cases that are near to the outliers to delete them and maintain those cases that are completely internal and have not any case whose competence are contained. In NACCM the process is to select cases to be maintained in the case memory until all the classes contain almost one case.

The NACCM algorithm is divided in two steps:
Step 1 converts $coverage$ measure of each case to its negation measure in order to let us to modify the selection process from internal to outliers points.
Step 2 uses the algorithm 4.2 that describes the SelectCases in NACCM process.

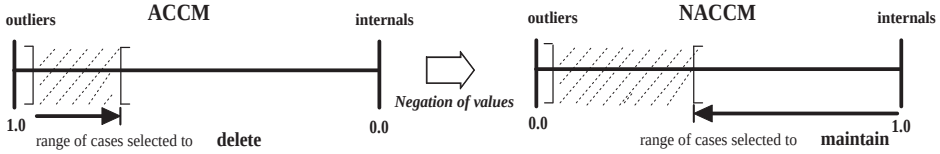

Figure 2: Description of ACCM and NACCM process.

SortOutInternalCM

SortOutInternalCM (CaseMemory T)

1. Sort out each instance $t \in T$ in its corresponding $ReachabilityGroup(S)$
2. Order decremented each $ReachabilityGroup(S)$ by $CoverageGroup(s) \in ReachabilityGroup(S)$
3. **for** each $ReachabilityGroup(S)$
4. **for** each $CoverageGroup(s) \in ReachabilityGroup(S)$
5. Select the first instance as a $MasterCase(t)$
6. **if** $Coverage(t) \neq 1.0$, Delete the rest of instances from T in the $CoverageGroup(s)$
7. **elseif** $Coverage(t) = 1.0$
   Select the rest of instances as a $MasterCase(t)$ to maintain in T
8. **endif**
9. **end for**
10. **end for**
11. **return** CaseMemory T

Thus, the selection of cases starts from internal cases to outliers ones. The aim is to maintain the minimal set of cases in the case memory. The behaviour of this reduction technique will be very similar to ACCM, but NACCM allows less cases to be maintained in the case memory. Figure 3 shows the behaviour of this technique using the example case memory shown in figure 1. The experimental analysis of this global case base maintenance method can be seen in section 5.2.

4.3 Local approach named Sort Out Internal Case Memory

The local approach uses the information of the knowledge to sort out the case memory. The information of the case memory allows the algorithm to separate between different groups of information, called $ReachabilityGroups$. After each $ReachabilityGroup$ is formed, different $CoverageGroups$ are builded using the information of the $coverage$ measure. There exists different case base maintenance approaches to use in the $SortOut$ categorisation model [12]. We have decided to use the *Sort Out Internal Case Memory* algorithm to test its behaviour in front of the global approach because it follows a similar approximation to the global approach in its foundations but it changes the way of extract the most important part of the knowledge, without modifying the instances as done in the building sort out techniques.

The behaviour of algorithm in the *Sort Out Case Memory* categorisation can be seen in

figure 3. The algorithm 4.3 modifies only the internal $CoverageGroups$ and maintains all the cases present in an outlier $CoverageGroup$. The outlier cases are isolated cases that no other case but itself can solve. Thus, it is important to maintain them because a $MasterCase$ can not be a good representative of the $CoverageGroup$. In this case, each case in a outlier $CoverageGroup$ is an isolated space region of each class. It could be possible to find an outlier coverage group whose $MasterCase$ could be a good representative $MasterCase$, but this part involves further work.

Thus, the algorithm 4.3 selects from each internal $CoverageGroups$ a representative case, denoted as $MasterCase$ to be maintained in the case memory. Also, it selects as $MasterCases$ all the cases that are in an outlier $CoverageGroup$. Those cases not selected are removed from the case memory. The behaviour of this technique are shown in figure 3. The experimental analysis of the algorithm is described in section 5.2.

4.4 Mixed approach between SOI and NACCM

The reader will notice in figure 3 that algorithm 4.3 SOI selects -at least- one case for each $CoverageGroup$. On the other hand, the global approach -algorithm 4.2 NACCM- selects cases that are internal and the outlier ones. Thus, deleting all the cases that belong to some $CoverageGroups$ and maintaining all the cases of the remaining $CoverageGroups$. In both approaches, we prefer to maintain or even improve the competence, selecting a fewer number of cases to be deleted from the case memory. Thus, the reduction of the case memory is not great enough.

After an analysis of the global and local techniques, detailed in section 5.2, the idea of a great reduction promotes a unified approach between the previous techniques. At the same time, our motivation is to analyse the behaviour of such a combined technique. The aim is twofold: first, to reduce the case base; second to improve utility of our case memory maintaining its diversity. Thus, an extension of the previous algorithm to include the global behaviour is algorithm 4.4 SOI-NACCM.

The motivation of our mixed approach is founded in the analysis of the previous techniques. Let see their behaviour in figure 3 using the example case memory of figure 1. As it can be seen, the NACCM algorithm, figure 3(a), selects a subset of the training instances to be maintained and a subset to be removed using the $confidenceLevel$ computed but the cases that belong to each subsets have, all of them, the same $coverage$ measure. Thus, the system maintains a group of $coverageGroups$ while removing the remaining ones. Looking carefully the behaviour of SOI algorithm, detailed in figure 3(b), it organise the case memory in $ReachabilityGroups$ and $CoverageGroups$. Once the sort out of the case memory is performed it selects one case as a $MasterCase$ for each $CoverageGroup$, except for the outlier cases, while deleting the remaining ones. Thus, the system maintain at least one case for each $CoverageGroup$. However, a question can arise when looking the way of doing the process in both algorithms: it is necessary to maintain at least one case of each $CoverageGroup$ in the sort out model when the global approach does not maintain some of them?

The question suggests us to analyse the unified approach in order to improve the sort out internal case memory. If we apply the global approach in the Sort out model, the resulting case memory will be reduced. The unified approach behaviour can be seen in figure 3(c), where some of the previous $CoverageGroups$ have not a $MasterCase$ selected to be maintained. Also, the algorithm maintains the previous policy to do not remove outlier cases because it is

SortOutInternalCM

SortOutInternalCM (CaseMemory T)

1. Sort out each instance $t \in T$ in its corresponding $ReachabilityGroup(S)$
2. confidenceLevel = 1.0 and freeLevel = ConstantTuned *(set at 0.01)*
3. **while** not $\exists$ at least a t in $SelectCase$ for each class c that $reachability(t) = c$
4. select all instances $t \in T$ as $SelectCase(t)$ if accomplish:
5. confidenceLevel = confidenceLevel - freeLevel
6. **end while**
7. Order decremented each $ReachabilityGroup(S)$ by $CoverageGroup(s) \in ReachabilityGroup(S)$
8. **for** each $ReachabilityGroup(S)$
9. **for** each $CoverageGroup(s) \in ReachabilityGroup(S)$
10. Select the first instance as a $MasterCase(t)$
11. **if** $Coverage(t) \neq 1.0$, Delete the rest of instances from T in the $CoverageGroup(s)$
12. **elseif** $Coverage(t) = 1.0$
    Select the rest of instances as a $MasterCase(t)$ to maintain in T
13. **endif**
14. **end for**
15. **end for**
16. **return** CaseMemory T

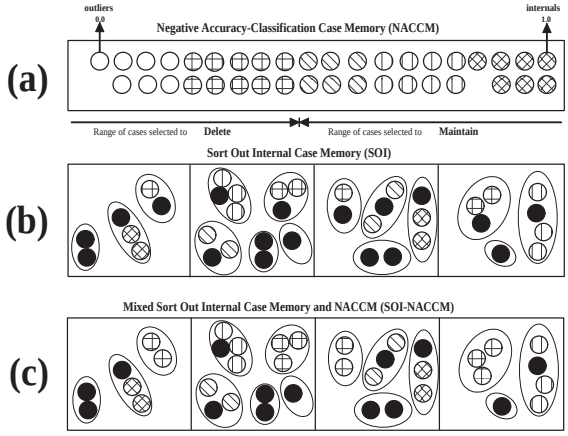

Figure 3: Behaviour of the Global, Local and Mixed CBM techniques.

known that they contribute greatly to the competence of a system.

5 Experimental analysis

This section is structured as follows: first, we describe the test suite used in the experimental analysis; then we discuss the results obtained from the ACCM method, the Sort Out Internal method and the unification of the global approach in the local case base maintenance method.

5.1 Test suite

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 [5]. They are: *Breast Cancer Wisconsin (Breast-w), Glass, Ionosphere, Iris, Sonar and Vehicle*. **Private datasets** [2] come from our own repository. They deal with *diagnosis* of breast cancer (*Biopsy* and *Mammogram*) and *synthetic* datasets (*MX11* is the eleven input multiplexer and TAO-*grid* is obtained from sampling the TAO figure using a grid). 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	Ref.	Samples	Numeric feat.	Symbolic feat.	Classes	Inconsistent
1	*Biopsy*	*BI*	1027	24	-	2	Yes
2	*Breast-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	*MX11*	*MX*	2048	-	11	2	No
8	*Sonar*	*SO*	208	60	-	2	No
9	*TAO-Grid*	*TG*	1888	2	-	2	No
10	*Vehicle*	*VE*	846	18	-	4	No

The study described in this paper was carried out in the context of BASTIAN, a *case*-**BA***sed* **S***ys***T***em* **I***n cl***A***ssificatio***N**. BASTIAN has been developed in JAVA, for details see [10]. All techniques were run using the same set of parameters for all datasets. The configuration of BASTIAN platform for this paper is set as follows. It uses a 1-Nearest Neighbour Algorithm. The case memory is represented as a list of cases. Each case contains the set of attributes, its class and the AccurCoef and ClassCoef coefficients. Our goal is to test the reliability and feasibility of the reduction techniques. Therefore, we have not focused on the case representation used by the system. The retain phase use two policies: *DifSim* that only store the new case if it has different similarity from the retrieved case, and *DifClass*) that only store the new case if it has different class from the retrieved one. Thus, the learning process is limited to two simple policies. Future work will be focused on improving the retain policy. Finally, no weighting method is used in this paper in order to test the reliability of our reduction techniques.

The percentage of correct classifications has been averaged over stratified ten-fold cross-validation runs. We analyse the significance of the performance using two-sided paired *t*-test (p=0.1) on these runs.

5.2 Experiment 1- Analysis of global vs. local algorithms

First analysis corresponds to the comparison between global and local algorithms. The aim of both reduction techniques is to reduce the case memory while maintaining the competence of the system. This priority guides our reduction techniques based on Rough Sets competence model. That fact is detected in the results shown in table 2, where NACCM and SOI algorithms obtain on average a higher generalisation on competence than IBL. NACCM and SOI algorithm improve on a significant level in some datasets (e.g. *sonar,vehicle*) while maintaining the competence, with the exception of the *TAO-grid* example because it reduces too much the case base. The performance of IBL algorithms declines, in almost all datasets (e.g.

Table 2: Mean percentage of correct classifications (%PA) and mean storage size (%CM). A ○ and • stand for a significant improvement or degradation of the reduction techniques related to the CBR. Bold font indicates the best prediction accuracy and smallest case base.

Ref.	**CBR**		**NACCM**		**SOI**		**IB2**		**IB3**		**IB4**	
	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM
BI	83.15	100.0	83.66	99.3	**83.75**	88.74	75.77•	26.65	78.51•	13.62	76.46•	**12.82**
BC	**96.28**	100.0	95.72	59.52	95.85	29.42	91.86•	8.18	94.98	2.86	94.86	**2.65**
GL	**72.42**	100.0	64.48	**33.91**	64.48	37.89	62.53•	42.99	65.56	44.34	66.40	39.40
IO	90.59	100.0	90.30	56.80	**91.16**	50.68	86.61•	15.82	90.62	**13.89**	90.35	15.44
IR	96.0	100.0	93.33	42.88	91.33•	13.18	93.98	**9.85**	91.33•	11.26	**96.66**	12.00
MA	64.81	100.0	60.18	44.80	58.04	25.36	**66.19**	42.28	60.16	**14.30**	60.03	21.55
MX	78.61	100.0	78.61	99.90	78.61	99.90	**87.07**○	18.99	81.59	15.76	81.34	**15.84**
SO	84.61	100.0	**86.90**○	78.24	86.42○	65.15	80.72	27.30	62.11•	**22.70**	63.06•	22.92
TG	**95.76**	100.0	90.25•	1.54	89.66•	**1.37**	94.87•	7.38	95.04•	5.63	93.96•	5.79
VE	67.37	100.0	69.10○	72.35	**69.70**○	68.33	65.46	40.01	63.21•	33.36	63.68•	**31.66**

Breast-w, Biopsy), when case memory is reduced. On the other side, the mean storage size obtained is higher in our reduction techniques than those using IBL schemes.

5.3 *Experiment 2- Analysing the unified algorithm*

Table 3 shows the results for all the reduction techniques explained in this paper. As it can be seen, the unified approach (SOI-NACCM) has obtained a higher reduction of the case memory, but the competence of the system varies from one dataset to another. However, the competence is maintained in all datasets in a significance level. In some domains it is improved (e.g. *Glass* and in other domains it is maintained while achieving a higher reduction. It is interesting to worth noting that the competence is always between the competence values of the NACCM and SOI algorithms. However, the maximum reduction is always obtained by the new unified approach, the SOI-NACCM algorithm.

Table 3: Mean percentage of correct classifications (%PA) and mean storage size (%CM). A ○ and ● stand for a significant improvement or degradation of the reduction techniques related to CBR. Bold font indicates the best prediction accuracy and smallest case base.

Ref.	**CBR**		**NACCM**		**SOI**		**SOI-NACCM**	
	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM
BI	83.15	100.0	83.66	99.3	**83.75**	88.74	**83.75**	**88.74**
BC	**96.28**	100.0	95.72	59.52	95.85	29.42	95.85	**27.87**
GL	**72.42**	100.0	64.48	33.91	64.48	37.89	64.93	**29.33**
IO	90.59	100.0	90.30	56.80	**91.16**	50.68	90.32	**49.44**
IR	96.0	100.0	93.33	42.88	91.33●	13.18	91.33●	**4.59**
MA	64.81	100.0	60.18	44.80	58.04	25.36	58.63	**19.44**
MX	78.61	100.0	78.61	99.90	78.61	99.90	78.61	99.90
SO	84.61	100.0	**86.90**○	78.24	86.42○	65.15	85.95○	**64.90**
TG	**95.76**○	100.0	90.25●	1.54	89.66●	**1.37**	87.97●	**0.11**
VE	67.37	100.0	69.10○	72.35	**69.70**○	68.33	69.35○	**66.07**

Table 4: Mean percentage of correct classifications (%PA) and mean storage size (%CM). Two-sided paired t-test (p = 0.1) is performed, where a ○ and ● stand for a significant improvement or degradation of the CBM techniques related to CBR system. Bold font indicates the best prediction accuracy.

Ref.	**CBR**		**SOI-NACCM**		**IB2**		**IB3**		**IB4**	
	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM
BI	83.15	100.0	**83.75**	88.74	75.77●	26.65	78.51●	13.62	76.46●	**12.82**
BC	**96.28**	100.0	95.85	27.87	91.86●	8.18	94.98	2.86	94.86	**2.65**
GL	**72.42**	100.0	64.93	**29.33**	62.53●	42.99	65.56	44.34	66.40	39.40
IO	90.59	100.0	90.32	49.44	86.61●	15.82	90.62	**13.89**	90.35	15.44
IR	96.0	100.0	91.33●	**4.59**	93.98	**9.85**	91.33●	11.26	**96.66**	12.00
MA	64.81	100.0	58.63	19.44	**66.19**	42.28	60.16	**14.30**	60.03	21.55
MX	78.61	100.0	78.61	99.90	**87.07**○	18.99	81.59	15.76	81.34	**15.84**
SO	84.61	100.0	**85.95**○	64.90	80.72	27.30	62.11●	**22.70**	63.06●	22.92
TG	**95.76**○	100.0	87.97●	**0.11**	94.87●	7.38	95.04●	5.63	93.96●	5.79
VE	67.37	100.0	**69.35**○	66.07	65.46	40.01	63.21●	33.36	63.68●	**31.66**

The unified approach does not achieve the best competence but it guarantees that the system is robust between a maximum and minimum values, while achieving the most reduced case memory. In order to finish the comparison, we also compared the unified approach with the IBL schemes.

The competence of the SOI-NACCM algorithm is on average similar to those obtained using the global and the local approaches, and in consequence better than those obtained using the IBL schemes. The main difference between this approach and the previous ones is that SOI-NACCM obtains a most similar reduction to IBL schemes. In some datasets the reduction obtained is higher (e.g. *Glass, iris, TAO-grid*).

5.4 Discussion

The unified approach does not obtain a higher competence than the original reduction techniques but it has obtained better reduction. It is worth noting that there are some facets that have to be taken into account.

1. the analysis has been performed without using weighting methods. It could be interesting to test the influence of such methods when applied at the same time as the reduction techniques.

2. in some domains the reduction is extreme. This fact produce that the competence decrease because the retain phase used is the most simple policies. As denoted by Leake , two cases are better than one. Thus, it is also important to combine the reduction techniques with a properly retain phase.

3. the analysis of the combined approach has to be extended to recommenders systems.

4. The most important part of the unified approach is that it is a starting point to combine the *Sort out* techniques with different global reduction techniques.

6 Conclusions and further work

This paper presents an analysis between global and local approaches to case base maintenance, and a unified approach. All the reduction techniques are developed under a competence model based on Rough Sets theory. The aim of the paper was twofold: (1) to denote that the global and local approaches are focused in a different space regions, and (2) to show that the unified approach can be an alternative for a robust CBR system. Experimental analysis show that these reduction techniques produces a higher or equal generalisation accuracy on classification tasks. We can conclude that the mixed approach between global and local case base maintenance techniques can be a good alternative to maintain robust CBR systems. However, it could be improved in some facets and it is necessary to increase the study to quickly changing environments over time. This kind of environments are the focus of our future work. Also, as shown in our discussion it is necessary to have a good retain policy to improve the system. Finally, we want to analyse the influence of the weighting methods and similarity functions in these reduction techniques.

Acknowledgements

This work is supported by the *Ministerio de Ciencia y Tecnologia*, Grant No. TIC2002-04160-C02-02. We also wish to thank D.W. Aha for providing the IBL code.

References

[1] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning, Vol. 6*, pages 37–66, 1991.

[2] E. Golobardes, X. Llorà, M. Salamó, and J. Martí. Computer Aided Diagnosis with Case-Based Reasoning and Genetic Algorithms. *Knowledge-Based Systems*, (15):45–52, 2002.

[3] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, Inc., 1993.

[4] D. Leake and D. Wilson. Remembering Why to Remember: Performance-Guided Case-Base Maintenance. In *Proceedings of the Fifth European Workshop on Case-Based Reasoning*, pages 161–172, 2000.

[5] 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.

[6] Z. Pawlak. Rough Sets. In *International Journal of Information and Computer Science*, volume 11, 1982.

[7] L. Portinale, P. Torasso, and P. Tavano. Speed-up, quality and competence in multi-modal reasoning. In *Proceedings of the Third International Conference on Case-Based Reasoning*, pages 303–317, 1999.

[8] T. Reinartz and I. Iglezakis. Review and Restore for Case-Base Maintenance. *Computational Intelligence*, 17(2):214–234, 2001.

[9] C.K. Riesbeck and R.C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Hillsdale, NJ, US, 1989.

[10] M. Salamó and E. Golobardes. BASTIAN: Incorporating the Rough Sets theory into a Case-Based Classifier System. In *Butlletí de l'acia: III Congrés Català d'Intel·ligència Artificial (CCIA'00)*, pages 284–293, Barcelona, Spain, October 2000.

[11] M. Salamó and E. Golobardes. Rough sets reduction techniques for case-based reasoning. In *Proceedings 4th. International Conference on Case-Based Reasoning, ICCBR 2001*, pages 467–482, Vancouver, BC, Canada, 2001.

[12] M. Salamó and E. Golobardes. Deleting and building sort out techniques for case base maintenance. In *European Conference on Case-Based Reasoning*, 2002.

[13] M. Salamó and E. Golobardes. Hybrid deletion policies for case base maintenance. In *FLAIRS-2003*, page To appear, 2003.

[14] M. Salamó, E. Golobardes, D. Vernet, and M. Nieto. Weighting methods for a Case-Based Classifier System. In *LEARNING'00*, Madrid, Spain, October 2000. IEEE.

[15] B. Smyth and M. Keane. Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In *Proceedings of the Thirteen International Joint Conference on Artificial Intelligence*, pages 377–382, 1995.

[16] B. Smyth and E. McKenna. Building compact competent case-bases. In *Proceedings of the Third International Conference on Case-Based Reasoning*, pages 329–342, 1999.

[17] B. Smyth and E. McKenna. Competence Models and the maintenance problem. *Computational Intelligence*, 17(2):235–249, 2001.

[18] D.C. Wilson and D.B. Leake. Maintaining Case-Based Reasoners:Dimensions and Directions. *Computational Intelligence*, 17(2):196–213, 2001.

[19] D.R. Wilson and T.R. Martinez. Reduction techniques for Instance-Based Learning Algorithms. *Machine Learning, 38*, pages 257–286, 2000.

[20] 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.

[21] J. Zhu and Q. Yang. Remembering to add: Competence-preserving case-addition policies for case base maintenance. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 234–239, 1999.