Unifying weighting and case reduction methods based on Rough Sets to improve retrieval

Maria Salamó and Elisabet Golobardes

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

Abstract. Case-Based Reasoning systems usually retrieve cases using a similarity function based on K-NN or some derivatives. These functions are sensitive to irrelevant or noisy features. Weighting methods are used to extract the most important information present in the knowledge and determine the importance of each feature. However, this knowledge, can also be incorrect, redundant and inconsistent. In order to solve this problem there exist a great number of case reduction techniques in the literature. This paper analyses and justifies the relationship between weighting and case reduction methods, and also analyses their behaviour using different similarity metrics. We have focused this relation on Rough Sets approaches. Several experiments, using different domains from the UCI and our own repository, show that this integration maintain and even improve the performance over a simple CBR system and over case reduction techniques. However, the combined approach produces CBR system decrease if the weighting method declines its performance.

1 Introduction

The success of any Case-Based Reasoning (CBR) system depends on its ability to select the right case for the right target problem [Aamodt and Plaza, 1994]. The quality of the information in the case memory is one of the key issues. When the information is redundant, irrelevant, or noisy and/or unreliable, the success of the CBR system when classifying is more difficult. The case memory is a key piece in a CBR cycle because it is present in the whole process. Although we concentrate on CBR systems, the need for information with high quality is also present in other machine learning techniques (e.g. decision trees). Many researchers have addressed the improvement of this quality, most of them using two major approximations to it: (1) weighting or feature selection methods and (2) prototype selection or reduction techniques (identified as case reduction methods in the CBR community). Both approaches are focused on the information but they concentrate on different dimensions: the first one focuses on the features (attributes) while the second one focuses on the cases (instances).

The motivation of this paper is addressed after a previous analysis of three majors factors on the retrieval phase, see figure 1. The first factor is that weighting methods are influenced by the case memory size. At the same time, as a second continuous factor is the positive influence of weighting methods when using the similarity function to retrieve the most similar cases. Finally, the case memory itself is influenced by the similarity due to the policy applied when retaining which is dependant on the correct classification of the new case.

In this paper, we concentrate on an analysis of a combined approach between weighting and case reduction techniques, as a consequence of previous analysis. The combined approach include two of the main parts of the retrieval phase. In this case, we use Proportional Rough Sets method (PRS) as a weighting method and the Accuracy Classification Case Memory (ACCM) algorithm as a case reduction technique. Finally, we tested this combination using several similarity functions in order to test if there is some clear positive or negative interaction between the methods in the combination.

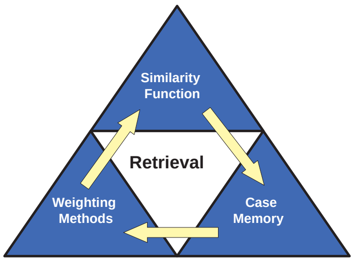


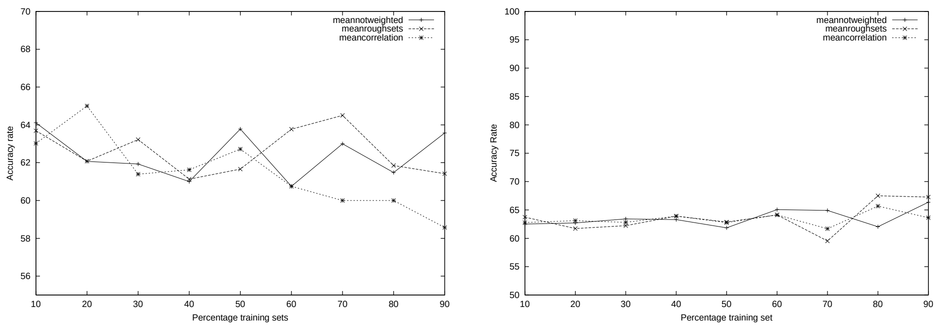
Fig. 1. Dependencies schema in the retrieval phase.

The paper is organized as follows. Section 2 describes the motivation for the paper. Next, Section 3 details the proposed unification between weighting methods and case reduction methods. Section 4 describes the experiments and analyses the results obtained. Then, Section 5 introduces some related work. Finally, Section 6 presents some conclusions and further work.

2 Motivation

The motivation of this paper originates in a previous analysis on the behaviour of weighting methods. This previous analysis [Salamó and Golobardes, 2002] demonstrated the positive influence of weighting methods and the relationship between weighting methods and case memory growth. Although the paper concentrated on weighting methods, the results can be extrapolated to feature selection methods.

In the previous paper, we tested the weighting methods using several datasets, each one with 9 proportions of the case memory. Proportions are in the range $X \in \{10\%, 20\%, \ldots, 90\%\}$ of the initial case memory for training where the remaining cases are used for testing. Each proportion was generated 10 times and the accuracy results of each test are averaged. Here we present the most significant results obtained in our previous paper for the *echocardiogram* (see figure 2(a)), *iris* (see table 1) and *mammogram* (see figure 2(b)) datasets. Details of datasets can be seen in section 4.1.



(a) Echocardiogram (b) Mammogram

Fig. 2. Figures describing the accuracy of the CBR system when increasing the training case memory size.

Table 1. Mean average accuracies results for the *Iris* dataset. The columns show the results for a non weighting approach (¬W), Proportional Rough Sets (PRS) and Sample Correlation (Corr) weighting methods.

Prop. Train	Mean ¬W	Mean PRS	Mean Corr
40%	96.22	96.00	96.22
60%	95.33	95.50	96.16
70%	95.11	95.33	95.77
80%	97.00	97.00	97.33
90%	96.66	96.66	97.33

As a summary of previous results, we can notice that the system performs better when using weighting methods. The most important point is that the CBR improves with enough case memory. However, it is noticeable that the case memory increase also produces a declination on performance when the case memory increases too much. Thus showing also that the number of cases included in the case memory influences the performance of the weighting methods. This influence can be seen in figures 2(a) and 2(b). These figures show the evolution of performance when the case memory increases. In conclusion, it is important to remark that the performance of the weighting seems to depend on the case memory size and also depend on the quality of the cases present in it, as reported in all datasets analysed.

Although the results are not conclusive in the previous paper, previous observations motivated us to perform the experiments and analysis described in this paper.

3 Unifying weighting and case reduction methods based on Rough Sets

After demonstrating the influence of the case memory over weighting methods, we present in this section a combined approach between weighting methods and case reduction methods. The proposed solution is based on Rough Sets. Here we present a unification of two kinds of algorithms that work well separately.

The most important point is that they have the same foundations, even though they use different policies: in one case to weight features and on the other hand to maintain or delete cases.

First of all, we present a summary of Rough Sets foundations of all algorithms. Next, we describe our weighting method and case base maintenance method tested and how they are unified in the CBR system.

3.1 Rough Sets foundations

The rough sets theory defined by Zdisław Pawlak, which is well detailed in [Pawlak, 1982,Pawlak, 1991], 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 IND defined on a certain universe of objects (cases) U and relations (attributes) R. IND defines a partition on U. The pair of the universe objects U and the associated equivalence relation IND forms an approximation space. The approximation space gives an approximate description of any subset $X \subseteq U$.

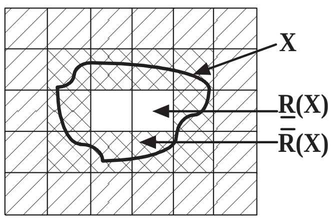


Fig. 3. The lower and upper approximations of a set X.

Two approximations are generated by the available data about the elements of the set X, called the lower and upper approximations (see figure 3). 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.

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.

3.2 Proportional Rough Sets weighting method (PRS)

The relevance of each feature in the system is computed using the proportional appearance at the reducts of information.

$$For\ each\ feature\ f\ computes : \mu(f) = \frac{card(appearance\ f\ in\ RED(R))}{card(\ all\ RED(R))} \quad (1)$$

An attribute f that does not appear in the reducts has a feature weight value $\mu(f) = 0.0$, whereas a feature that appears in the core has a feature value $\mu(f) = 1.0$. The remaining attributes have a feature weight value depending on the proportional appearance in the reducts. This weighting method has been selected because it has a good behaviour on different application areas. The comparison of this weighting method and well known weighting methods is detailed in [Salamó and Golobardes, 2002].

3.3 Accuracy-Classification Case Memory maintenance method (ACCM)

ACCM algorithm has been selected, from different Rough Sets case reduction techniques [Salamó and Golobardes, 2003], because in previous experiments it presents a good balance between reduction and accuracy. This algorithm uses a categorisation model of the case memory. Next, we briefly introduce the main definitions.

Categorisation model of case memory The distribution of the case memory is done using a categorisation in terms of their *coverage* and *reachability*, which are adapted to our needs. In the case of coverage it is measured using Rough Sets theory, equally it does the weighting method. The reachability is modified in order to be employed in classification tasks.

Definition 1 (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. When $AccurCoef$ value is 1.0, the $Coverage$ is 1.0 but when it is 0.0 value, the $Coverage$ is $ClassCoef$ value.

Definition 2 (AccurCoef)
This measure computes the *Accuracy* coefficient (**AccurCoef**) of each case t in the knowledge base (case memory) T as:

$$For\ each\ instance\ t\ \in\ T\ it\ computes : AccurCoef(t) = \frac{card\ (\ \underline{P}(t))}{card\ (\ \overline{P}\ (t))} \quad (2)$$

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* 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. 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 a 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 $\overline{P}$), the lower the accuracy of the set.

Definition 3 (ClassCoef)
In this measure we use the *quality of classification* coefficient (**ClassCoef)**. It is computed as:

$$\begin{gathered} For\ each\ instance\ t\ \in\ T\ it\ computes : \\ \mu(t) = \frac{card\ (\ \underline{P}(t))\ \cup\ card\ (\ \underline{P}(-t))}{card\ (\ all\ instances)} \end{gathered} \quad (3)$$

Where $ClassCoef(t)$ is the relevance of the instance t; T is the training set; $-t$ is $T - \{t\}$ set; $card$ is the cardinality of a set; P is a set that contains the reducts; 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 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.

Definition 4 (Reachability)
Let $T = \{t_1, t_2, ..., t_n\}$ be a training set of instances, $\forall t_i \in T$:

$$Reachability(t_i) = \begin{cases} Class\ (t_i) & if\ it\ is\ a\ classification\ task \\ Adaptable(t', t_i) & if\ it\ is\ not\ a\ classification\ task \end{cases} \quad (4)$$

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

Accuracy-Classification Case Memory (ACCM) algorithm Once we have computed the AccurCoef and ClassCoef, we apply for the original case memory algorithm 1 to select the cases that have to be deleted from the case memory. The cases not selected are maintained in the case memory. An extended explanation of this can be found in [Salamó and Golobardes, 2003].

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 AccurCoef = 1.0 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.

Algorithm 1 ACCM

1. SelectCasesACCM (CaseMemory T)
2. confidenceLevel = 1.0 and freeLevel = ConstantTuned *(set at 0.01)*
3. select all instances $t \in T$ as $SelectCase(t)$ if it satisfies: $coverage(t) \geq$ 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 it satisfies: $coverage(t) \geq$ confidenceLevel
7. **end while**
8. delete from CaseMemory the set of cases selected as $SelectCase$
9. **return** CaseMemory T

3.4 Unification of weighting and case reduction methods

The meta-level process of the unification can be described in three steps, as shown in figure 4. This process is performed in an initial phase prior to the CBR cycle. The first step discretises the initial training set of instances, using Fayyad and Irani's algorithm [Fayyad and Irani, 1993], in order to use Rough Sets theory. The second step searches for the reducts of knowledge using the Rough Sets theory. Finally, the third step uses the reducts of knowledge to extract the proportional appearance of each attribute and AccurCoef and ClassCoef measures. The last measures are used to compute the cases that have to be maintained and removed from the case memory using the algorithm 1 ACCM, thus reducing the initial training case memory. Weights are used when computing the similarity function.

Our approach based on the combination of PRS and ACCM has been done from the point of view of Rough Sets theory. The selection of this approach is done for two main reasons: (1) both methods share a common basis, what make it possible to obtain a higher speed because the central point of computations are the same; (2) both methods have demonstrated in previous papers their good behaviour in front of a great number of problems, PRS analysis versus well known weighting methods (e.g. ReliefF [Kononenko, 1994], CFS [Hall, 2000]) can be seen in [Salamó and Golobardes, 2002] and ACCM details can be found in [Salamó and Golobardes, 2003] where an analysis versus known case reduction techniques (e.g Instance Based learning (IB1-IB4) algorithms [Aha, 1992] and instance prunning techniques (DROP1-DROP5) [Wilson and Martinez, 2000b]) is performed.

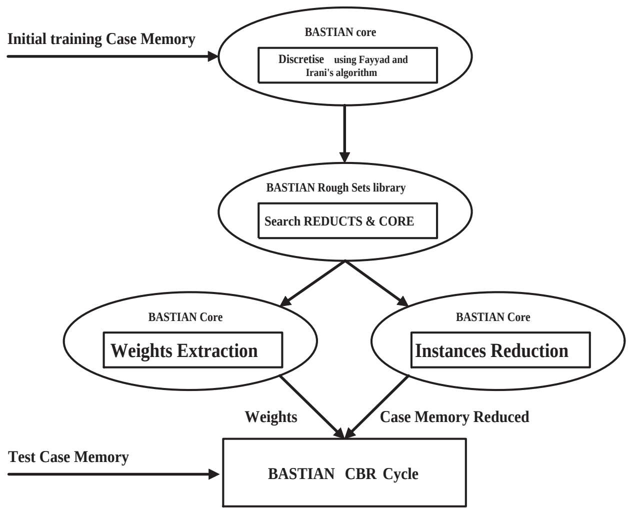


Fig. 4. Unification process in BASTIAN platform.

A sequential alternative, first case reduction and in second place weighting based on the reduced case memory, will be part of our future work. However, this sequential alternative have two main drawbacks at first sight: (1) it is not possible to improve execution time; (2) if the basis are the same, the reduced case memory will contain the same characteristics as the initial one to extract weights. Thus computing twice the same parameters. Last drawback will be true if the case memory reduction works well.

4 Experimental analysis

This section is structured as follows: first of all, we describe the testbed used in the experimental analysis; then we analyse the results obtained from the weighting methods, the case reduction technique and the the combined approach of both techniques in our CBR system.

4.1 Testbed

The evaluation of the performance rate is done using sixteen datasets which are detailed in table 2. Datasets can be grouped in two ways: *public* and *private*. **Public datasets** are obtained from the UCI repository [Merz and Murphy, 1998].

Private datasets [Golobardes et al., 2002] come from our own repository. They deal with *diagnosis* of breast cancer (*Biopsy* and *Mammogram*) and a *synthetic* dataset (TAO-*grid* which 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 2. Details of the datasets used in the analysis.

	Dataset	Ref.	Samples	Num. feat.	Sym. feat.	Classes	Inconsistent
1	*Audiology*	*AD*	226	61	2	24	Yes
2	*Biopsy (private)*	*BI*	1027	24	-	2	Yes
3	*Breast-w*	*BC*	699	9	-	2	Yes
4	*Credit-A*	*CA*	690	5	9	2	Yes
5	*Glass*	*GL*	214	9	-	6	No
6	*Heart-C*	*HC*	303	6	7	5	Yes
7	*Heart-H*	*HH*	294	6	7	5	Yes
8	*Heart-Statlog*	*HS*	270	13	-	2	No
9	*Ionosphere*	*IO*	351	34	-	2	No
10	*Iris*	*IR*	150	4	-	3	No
11	*Mammogram (private)*	*MA*	216	23	-	2	Yes
12	*Segment*	*SG*	2310	19	-	7	No
13	*Sonar*	*SO*	208	60	-	2	No
14	*TAO-Grid (private)*	*TG*	1888	2	-	2	No
15	*Vehicle*	*VE*	846	18	-	4	No
16	*Vote*	*VT*	435	-	16	2	Yes

The study described in this paper was carried out in the context of BASTIAN, a *case-***BA***sed* **S***ys***T***em for class***I***fic***A***tio***N**. 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 test the combination of weighting and case reduction methods. For this reason, we have not focused on the representation used by the system. The retain phase uses the following policy: *DifSim*, which only stores the new case if it has a different similarity from the retrieved case. Thus, the learning process is limited to this simple policy. Future work will be focused on improving the retain policy.

The configuration of BASTIAN system is different from previous papers, producing some changes on previous results. The percentage of correct classifications has been *averaged* over *stratified ten-fold cross-validation* runs. To study the performance we use paired *t-test* on these runs.

4.2 Experiment 1. Analysis of separated components and the unified approach for the retrieval phase

This section analyses each component studied (similarity function, weighting and case reduction method) in this paper versus the combined approach. The results are shown in table 3, where the similarity function analysed is an overlap metric for nominal attributes and normalised Euclidean distance function for linear attributes, the weighting approach is Proportional Rough Sets (PRS) method, the case reduction method is Accuracy Classification Case Memory (ACCM), and the combined approach is named ACCM+PRS.

Table 3. Results for all datasets showing the percentage of correct classifications. Last column shows the case memory size obtained when using ACCM in two previous columns. We use paired t-test at the 1% significance level, where a • and a ∘ stand for a significant improvement o degradation of PRS, ACCM and ACCM+PRS related to Euclidean. We also show paired t-test at the 5%, where a † or ‡ stand for a significant improvement or degradation.

Ref.	**Euclidean**	**PRS**	**ACCM**	**ACCM+PRS**	**size**
AD	75,36	77,93	71,84	72,58	70,00
BI	83,17	82,37	83,07	80,79	88,01
BC	95,86	96,14	94,99	95,00	77,36
CA	81,76	81,19	82,20	81,77	84,30
GL	66,30	76,56†	67,29	73,42	74,95
HC	74,20	76,19	73,58	73,91	82,02
HH	72,82	76,96•†	73,82	76,58•†	85,63
HS	74,07	81,11	76,29	78,89	79,67
IO	86,33	87,75	87,20	86,60	83,77
IR	95,33	96,00	96,66	96,66	89,03
MA	62,95	65,79	63,56	65,84	89,19
SG	97,35	97,31	97,40	97,10	57,59
SO	86,83	83,25	86,90	83,71	71,71
TG	96,13	96,66	96,29	96,29	95,87
VE	69,43	70,44	68,48	67,55	72,35
VT	86,65	88,23	90,78	91,49†	79,23
Mean	81,53	83,37	81,90	82,39	80,04

The results show that the combination, between PRS weighting method and ACCM case reduction method, obtains an average behaviour on performance for the majority of datasets. The behaviour of the combination depends initially on the behaviour of the weighting method. When PRS and ACCM increases the accuracy, in comparison with Euclidean, the combined approach also increases the prediction accuracy, as can be seen in *heart-h* and *tao-grid* datasets. The results also show that an increase combined with a decrease on performance in PRS or ACCM produces an increase on performance if one of the methods achieve a great difference on the initial performance (e.g. *glass* and *vote*). Another interesting observation is that when PRS or ACCM declines its performance, the combination performance loss is not as great as the first one.

All the results show that the combination of the case reduction and weighting methods is on average positive on the CBR system. The ACCM maintains some negative results obtained when weighting (e.g. *vote*), and at the same time, PRS maintains or improves some negative results obtained by ACCM (e.g. *mammogram*). Maybe the results are not so high as expected, but it is also important to note that the reduction of the case memory is performed while achieving a robust CBR system.

4.3 Experiment 2- Comparison of the combined approach versus Similarity Function

After considering the previous results, we want to analyse the influence of the similarity function in our combined approach. We want to observe if the similarity function produces an increment or decrement in the performance of the system. In this case, we test Camberra, Clark, Manhattan, Euclidean and Minkowski -set up r=3- (Cubic) similarity functions combined with ACCM+PRS approach.

Table 4. Results for all datasets showing the percentage of correct classifications.

Ref.	**Camberra**	**Clark**	**Manhattan**	**Euclidean**	**Cubic**
AD	75,58	**76,98**	72,58	72,58	72,53
BI	78,19	78,60	**80,80**	80,79	79,86
BC	94,29	93,32	**95,55**	95,00	93,99
CA	82,58	**82,87**	82,20	81,77	81,04
GL	70,63	67,12	**76,34**	73,42	72,00
HC	**76,86**	75,47	74,22	73,91	72,89
HH	78,59	**78,93**	77,24	76,58	76,58
HC	**79,63**	77,78	78,15	78,89	79,26
IO	**91,76**	91,47	90,62	86,60	82,91
IR	95,33	96,00	96,00	**96,66**	**96,66**
MA	60,28	60,14	61,21	**65,84**	64,46
SG	93,85	91,13	**97,45**	97,10	96,62
SO	76,66	70,11	82,35	83,71	**84,26**
TG	95,76	95,76	**96,29**	**96,29**	**96,29**
VE	68,31	67,48	**69,02**	67,55	65,79
VT	**91,49**	**91,49**	**91,49**	**91,49**	**91,49**
Mean	81,86	80,91	**82,59**	82,39	81,66

The results on table 4 show some great differences between different similarity functions. One of the major points to notice is that no one is able to achieve a maximum values in all datasets. Camberra function can deal well with datasets that contain a great number of missing values and at the same time a reduced set of cases, whereas Manhattan function is better than the usual Euclidean or Cubic distance functions.

Experiment 3- Comparison of the combined approach versus IDIBL Finally, we test a similar approach to our combined approach. However, in this case, the comparison has been performed using the information present in the paper that describes IDIBL method [Wilson and Martinez, 2000a]. So t-test can not be performed, and only a briefly comparison using nine datasets can be showed in table 5.

Table 5. Results for nine datasets showing the percentage of correct classifications.

Ref.	**IDIBL**	**Manhattan K=1**	**Manhattan K=3**
BC	97,00	95,55	95,12
GL	70,56	76,24	76,24
HC	83,83	74,22	74,22
HO	73,80	71,79	75,29
IO	87,76	90,62	90,62
IR	96,00	96,00	96,00
SO	84,12	82,35	83,70
VE	72,62	69,02	69,02
VO	95,62	91,49	90,08
Mean	84,59	83,03	83,37

The results are slightly lower in our combined approach. IDIBL approach has its own weighting method, its own case reduction method as our combined approach, but uses a different number of neighbours (K=3) and uses a different similarity function. Table 5 show the results for IDIBL in second column, our Manhattan similarity distance function using K=1 neighbours in third column and K=3 neighbours in the last column. The differences between IDIBL approach and our combined approach, in our opinion, is mainly produced by the similarity function used. Wilson and Martinez have reported that distance functions are not suitable for some kind of problems and the IVDM similarity function perform better than these kind of functions. For future work we will test their function in our combined approach. Another important difference between both approaches is that IDIBL tunes up parameters twice, while our approach does it only once. As explained in the unification process, this is part of our further work. Although the comparison is not fair in all parameters tested, we think that our results are promising and the IDIBL results address us to further investigate some improvements on the combined approach.

5 Related Work

There is little related work focused closely on the approach presented in this paper. One of the few closely ones to our proposal is the refinement of retrieval knowledge by optimising the feature/weights after case base maintenance

[Craw and Jarmulak, 2001]. The difference of this approach compared to our proposal is that the refinement of retrieval is performed all at the same time. Some work focused on similarity is also related to our proposal. The most similar approach is IDIBL algorithm [Wilson and Martinez, 2000a]. However, it uses a different similarity metric, uses K=3 neighbours and find parameters twice.

Many researchers have point out that it is important to obtain diversity in order to improve similarity, particularly in so-called *recommender systems* [Smyth and McClave, 2001]. McSherry in [McSherry, 2002] shows that it is possible to increase diversity without loss of similarity. Our present analysis argues that diversity maintained using CBM technique can help similarity and weighting during the retrieval phase.

Related work on weighting methods can be placed in two main categories: *Wrappers* and *Filters*. We concentrate on *filters* due to the fact that our PRS approach is a filter method. *Filters* use general characteristics of the data to evaluate features and operate independently of any learning algorithm. Many filter methods for feature selection have been proposed recently, and a review of them can be found in [Blum and Langley, 1997]. The simplest filtering scheme is to evaluate each feature individually measuring its correlation to the target function (e.g. using a mutual information measure) and then select K features with the highest value. The *Relief* algorithm, proposed by Kira and Rendell [Kira and Rendell, 1992], follows this general paradigm. Kononenko proposed an extension of it [Kononenko, 1994], called *ReliefF*, that can handle noisy and multiclass problems. Unlike *Relief*, *CFS* [Hall, 2000] evaluates and hence ranks feature subsets rather than individual features. The *CFS* algorithm is a subset evaluation heuristic that takes into account the usefulness of individual features for predicting the class along with the level of intercorrelation among them. In our weighting method, the relevant features are extracted using the reduction of feature space computed by the Rough Sets theory.

On the other hand, many researchers have addressed the problem of case memory reduction [Wilson and Martinez, 2000b] and different approaches have been proposed. The most similar methods to our approach are those focused on increasing the overall competence, *the range of target problems that can be successfully solved* [Smyth and Keane, 1995], of the case memory through case deletion. Strategies have been developed for controlling case memory growth. Several methods such as competence-preserving deletion [Smyth and Keane, 1995] and failure-driven deletion [Portinale et al., 1999], as well as for generating compact case memories [Smyth and McKenna, 2001] through competence-based case addition. Leake and Wilson [Leake and 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. ACCM approach uses a global policy to delete cases using a Rough Sets competence model [Salamó and Golobardes, 2003]. Reinartz and Iglezakis [Reinartz and Iglezakis, 2001] presented the maintenance integrated with the overall case-based reasoning process.

6 Conclusions

The aim of this paper has been to analyse and demonstrate the combination of weighting and case reduction techniques based on Rough Sets in the retrieval phase. First of all, this paper has presented the influence of case memory growth on weighting methods in a CBR system. Secondly, it has also presented a Rough Sets proposal that combines weighting and case reduction methods. The most important fact of the unification is that they share common foundations. Different experiments have shown the unification of both approaches produces maintenance or even an improvement on performance. The maintenance or improvement of the prediction accuracy is highly related to the initial behaviour of the weighting method, as denoted in the experiments, and not mainly to the case reduction method. The results also show that unification produces a robust system, because the system performance does not decrease too much if the weighting method does not perform good weights. Our further work will be focused on testing different Rough Sets case reduction methods and to combine different measures of feature relevance to improve the CBR system when the weighting method does not work efficiently. Also, to test different similarity functions not based on distance.

Acknowledgements

This work is supported by the *Ministerio de Ciencia y Tecnologia*, Grant No. TIC2002-04160-C02-02. We wish to thank *Enginyeria i Arquitectura La Salle - Ramon Llull University* for their support to our Research Group in Intelligent Systems. Finally, we wish to thank the anonymous reviewers for their useful comments.

References

[Aamodt and Plaza, 1994] Aamodt, A. and Plaza, E. (1994). Case-Based Reasoning: Foundations Issues, Methodological Variations, and System Approaches. In *AI Communications*, volume 7, pages 39–59.

[Aha, 1992] Aha, D. (1992). Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, (36):267–287.

[Blum and Langley, 1997] Blum, A. and Langley, P. (1997). Selection of Relevant features and Examples in Machine Learning. In *Artificial Intelligence*, volume 97, pages 245–271.

[Craw and Jarmulak, 2001] Craw, S. and Jarmulak, J. (2001). Maintaining Retrieval knowledge in a case-based reasoning system. *Computational Intelligence*, 17(2):346–363.

[Fayyad and Irani, 1993] Fayyad, U. and Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *19th International Joint Conference on Artificial Intelligence*, pages 1022–1027.

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

[Hall, 2000] Hall, M. (2000). Correlation-based feature selection of discrete and numeric class machine learning. In *Proc. International Conference on Machine Learning*, pages 359–366. Morgan Kaufmann.

[Kira and Rendell, 1992] Kira, K. and Rendell, L. (1992). A practical approach to feature selection. In *Proceedings of the 9th International Conference on Machine Learning*, pages 249–256.

[Kononenko, 1994] Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In *Proceedings of the Seventh European Conference on Machine Learning*, pages 171–182.

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

[McSherry, 2002] McSherry, D. (2002). Diversity-Conscious Retrieval. In *Proceedings of the 6th. European Conference on Case-Based Reasoning*, pages 219–233.

[Merz and Murphy, 1998] Merz, C. J. and Murphy, P. M. (1998). 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*.

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

[Pawlak, 1991] Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers.

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

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

[Salamó and Golobardes, 2002] Salamó, M. and Golobardes, E. (2002). Analysing rough sets weighting methods for case-based reasoning systems. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, (15):34–43.

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

[Smyth and Keane, 1995] Smyth, B. and Keane, M. (1995). 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.

[Smyth and McClave, 2001] Smyth, B. and McClave, P. (2001). Similarity vs. Diversity. In *Proceedings of the 4th. International Conference on Case-Based Reasoning*, pages 347–361.

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

[Wilson and Martinez, 2000a] Wilson, D. and Martinez, T. (2000a). An integrated instance-based learning algorithm. *Computational Intelligence*, 16(1):1–28.

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