Hybrid Deletion Policies for Case Base Maintenance

Maria Salamó and Elisabet Golobardes
Enginyeria i Arquitectura La Salle
Universitat Ramon Llull
Psg. Bonanova, 8 - 08022, Barcelona, Spain
{mariasal,elisabet}@salleurl.edu

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 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 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.

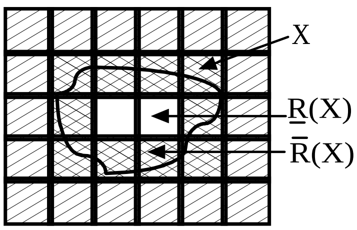

Figure 1: The lower and upper approximations of a set X.

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 (**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 *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 2 (Class Rough Sets)
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; $-t$ 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.

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 ($AccurCoef$ and $ClassCoef$). 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, $\forall\, t_i \in T$:
$Coverage(t_i)$= $AccurCoef(t_i) \oplus ClassCoef(t_i)$

The $\oplus$ operation is the logical sum of both values. When $AccurCoef$ value is 1, the $Coverage$ is 1.0 but when it is 0 value, the $Coverage$ is $ClassCoef$ value.

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

$$Reachability(t_i) = \begin{cases} Class\ (t_i) & if\ it\ is\ a\ C_t \\ Adaptable(t', t_i) & if\ it\ is\ not\ a\ C_t \end{cases} \tag{3}$$

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

Accuracy-Classification Case Memory (ACCM)

Once we have computed the AccurCoef and ClassCoef, we apply for the original case memory the 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. In a graphical manner, the process is represented in figure 2.

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 $Coverage = 1.0$ 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 t 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 t satisfies:* $coverage(t) \geq$ *confidenceLevel*
7. **end while**
8. ***Delete*** *from CaseMemory the set of cases selected as $SelectCase$*
9. **return** *CaseMemory T*

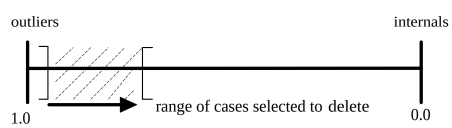

Figure 2: Graphical description of ACCM process.

Negative Accuracy-Classification Case Memory - (NACCM)

This reduction technique is based on the previous one, doing the complementary process as shown in figure 3. The motivation for this technique is to select a wider range of cases than the ACCM technique. The main process in ACCM is to select all the cases that are near to the outliers and maintain those cases that are completely internal and do not have any cases 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** convert coverage measure of each case to its negation measure in order to let us modify the selection process from internal to outlier points. **Step 2** use algorithm 2 that describes the SelectCasesNACCM process.

Algorithm 2 **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 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 t satisfies:* $coverage(t) \geq$ *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*

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.

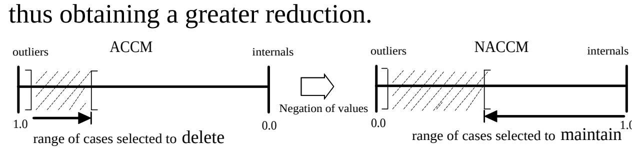

Figure 3: Graphical description of NACCM process.

Experimental study

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

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	Num. feat.	Sym. feat.	Classes	Inconsistent
1	*Biopsy*	*BI*	1027	24	-	2	Yes
2	*Breast-w*	*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**. 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. Thus, the learning process is limited to the reduced training set. 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.

The percentage of correct classifications has been *averaged* over *stratified ten-fold cross-validation* runs, with their corresponding standard deviations. To study the performance **we use two-sided paired t-test (p = 0.1)** on these runs, where ○ and ● stand for a significant improvement or degradation of the reduction techniques related to the first method of the table. Mean percentage of correct classifications is showed as **%PA** and mean storage size as **%CM**. Bold font indicates the best prediction accuracy.

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.

Table 2: Comparing Rough Sets reduction (ACCM, NACCM) techniques to IBL schemes (Aha & Kibler 1991).

Ref.	CBR		ACCM		NACCM		IB2		IB3		IB4	
	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM
BI	83.15	100.0	83.65	88.01	**83.66**	99.3	75.77●	26.65	78.51●	13.62	76.46●	12.82
BC	**96.28**	100.0	95.71	77.36	95.72	59.52	91.86●	8.18	94.98	2.86	94.86	2.65
GL	**72.42**	100.0	69.83	74.95	64.48	33.91	62.53●	42.99	65.56●	44.34	66.40●	39.40
IO	**90.59**	100.0	**90.59**	83.77	90.30	56.80	86.61●	15.82	90.62	13.89	90.35	15.44
IR	96.0	100.0	**96.66**	89.03	93.33	42.88	93.98	9.85	91.33●	11.26	**96.66**	12.00
MA	64.81	100.0	**66.34**	89.19	60.18	44.80	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	86.45○	71.71	**86.90**○	78.24	80.72	27.30	62.11●	22.70	63.06●	22.92
TG	95.76	100.0	**96.13**○	97.59	90.25●	1.54	94.87●	7.38	95.04●	5.63	93.96●	5.79
VE	67.37	100.0	**69.10**○	72.35	**69.10**○	72.35	65.46	40.01	63.21●	33.36	63.68●	31.66

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 DROP1, 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.

Table 3: Comparing ACCM technique to well known reduction techniques (Wilson & Martinez 2000).

Ref.	ACCM		CNN		SNN		DEL		ENN		RENN	
	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM
BI	**83.65**	88.01	79.57●	17.82	78.41●	14.51	82.79●	0.35	77.82●	16.52	81.03●	84.51
BC	95.71	77.36	95.57	5.87	95.42	3.72	96.57○	0.32	95.28	3.61	**97.00**○	96.34
GL	**69.83**	74.95	67.64	24.97	67.73	20.51	64.87●	4.47	68.23	19.32	68.66	72.90
IO	**90.59**	83.77	88.89●	9.94	85.75●	7.00	80.34●	1.01	88.31●	7.79	85.18●	86.39
IR	**96.66**	89.03	96.00	14.00	94.00●	9.93	96.00	2.52	91.33●	8.59	96.00	94.44
MA	**66.34**	89.19	61.04	25.06	63.42●	18.05	62.53●	1.03	63.85●	21.66	65.32	66.92
MX	78.61	99.90	89.01○	37.17	89.01○	37.15	68.99●	0.55	85.05○	32.54	**99.80**○	99.89
SO	**86.45**	71.71	83.26	23.45	80.38	20.52	77.45●	1.12	85.62	19.34	82.74	86.49
TG	96.13	97.59	94.39●	7.15	94.76●	6.38	87.66●	0.26	**96.77**	3.75	95.18	96.51
VE	69.10	72.35	**69.74**	23.30	69.27	19.90	62.29●	2.55	66.91	20.70	68.67	74.56

In table 4 the results obtained using ACCM and DROP algorithms show that ACCM has better competence for some 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 conservative. 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).

Ref.	**ACCM**		**DROP1**		**DROP2**		**DROP3**		**DROP4**		**DROP5**	
	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM	%PA	%CM
BI	**83.65**	88.01	76.36●	26.84	76.95●	29.38	77.34●	15.16	76.16●	28.11	76.17●	27.03
BC	95.71	77.36	93.28	8.79	92.56●	8.35	**96.28**	2.70	95.00	4.37	93.28	8.79
GL	**69.83**	74.95	66.39	40.86	69.57	42.94	67.27	33.28	69.18	43.30	65.02●	40.65
IO	**90.59**	83.77	81.20●	23.04	87.73●	19.21	88.89●	14.24	88.02●	15.83	81.20●	23.04
IR	**96.66**	89.03	91.33	12.44	90.00●	14.07	92.66●	12.07	88.67●	7.93	91.33●	12.44
MA	**66.34**	89.19	61.60	42.69	58.33●	51.34	58.51●	12.60	58.29●	50.77	61.60●	42.64
MX	78.61	99.90	87.94○	19.02	**100.00**○	98.37	82.37○	17.10	86.52 ○	25.47	86.52○	18.89
SO	86.45	71.71	84.64	25.05	**87.07**	28.26	76.57●	16.93	84.64●	26.82	84.64●	25.11
TG	**96.13**	97.59	94.76●	8.03	95.23●	8.95	94.49●	6.76	89.41●	2.18	94.76●	8.03
VE	**69.10**	72.35	64.66●	38.69	67.16	43.21	66.21	29.42	68.21	43.85	64.66●	38.69

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 Tecnologia*, 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

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

Brodley, C. 1993. Addressing the selective superiority problem: Automatic algorithm/model class selection. In *Proceedings of the 10th International Conference on Machine Learning*, 17–24.

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.

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*, 161–172.

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

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*, 303–317.

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

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

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

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*, 377–382.

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

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

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