2009-01-01

The Good, the Bad and the Incorrectly Classified: Profiling Cases for Case-Base Editing

Sarah Jane Delany
Technological University Dublin, sarahjane.delany@tudublin.ie

Recommended Citation

Delany, S. (2009) The Good, the Bad and the Incorrectly Classified: Profiling Cases for Case-Base Editing. L.McGinty & D. Wilson (eds) *International Conference on Case Based Reasoning (ICCBR 2009)*, LNCS 5650 p.135-149 Springer Verlag. doi:10.1007/978-3-642-02998-1_11

Funder: Science Foundation Ireland under Grant No. 07/RFP/CMSF718.

The Good, the Bad and the Incorrectly Classified: Profiling Cases for Case-base Editing

Sarah Jane Delany

Dublin Institute of Technology, Dublin, Ireland
sarahjane.delany@dit.ie

Abstract. Case-based approaches to classification, as instance-based learning techniques, have a particular reliance on training examples that other supervised learning techniques do not have. In this paper we present the RDCL case profiling technique that categorises each case in a case-base based on its classification by the case-base, the benefit it has and/or the damage it causes by its inclusion in the case-base. We show how these case profiles can identify the cases that should be removed from a case-base in order to improve generalisation accuracy and we show what aspects of existing noise reduction algorithms contribute to good performance and what do not.

1 Introduction

Unlike many other supervised learning techniques, lazy learning techniques are instance-based and depend greatly on individual training examples. This has motivated considerable research into the identification of appropriate training examples for case-based maintenance tasks. Case-base editing involves reducing a case-base or training set to a smaller number of cases while trying to maintain and even improve the generalisation accuracy. A key aspect of case-base editing is to identify and remove noisy or exceptional cases that can cause a degradation in the generalisation accuracy.

In this paper we present a technique for associating a competence profile with each case in a case-base. The case profile categorises each case based on three characteristics;

(i) whether the case is classified correctly or not by the rest of case-base,
(ii) what benefit (or good) if any, it brings to the case-base by its inclusion, and
(iii) whether or not it causes damage (or harm) to the case-base by its inclusion.

Different combinations of the three characteristics result in a case having one of eight possible profiles. Building on established case-base maintenance research [1, 2], these case profiles are derived from a competence model constructed on the case-base by a leave-one-out classification of all cases.

A key advantage of identifying the types of cases in a case-base is that it exposes the effect of removing cases of different types from a training set. This facilitates identifying which case types are the most useful in maintaining and improving generalisation accuracy.

A further important benefit of this profiling methodology is that it reveals the biases of existing noise reduction techniques and shows what aspects contribute to good performance and what aspects do not. Resulting from our analysis of these case profiles we present in this paper a simple noise reduction technique based on these profiles that consistently improves generalisation accuracy and we compare its performance to existing noise reduction techniques across a number of datasets.

The rest of this paper is structured as follows. Section 2 of this paper reviews the extensive existing case-editing literature. Section 3 presents our case profiling approach while section 4 describes the investigations we performed into showing which types of cases are most beneficial to remove from a case-base. This section also includes an investigation into what kinds of cases existing noise reduction algorithms remove and presents our profile-based noise reduction technique. Section 5 concludes with some directions for future work.

2 Case-base Editing

Case-base editing techniques have been categorised as *competence preservation* or *competence enhancement* techniques [3, 4]. Competence preservation corresponds to redundancy reduction, removing superfluous cases that do not contribute to classification competence. Competence enhancement is effectively noise reduction, removing noisy or corrupt cases from the training set. Competence preservation techniques aim to remove internal cases in a cluster of cases of the same class and can predispose towards preserving noisy cases as exceptions or border cases. Competence enhancement on the other hand aims to remove noisy or corrupt cases but can remove exceptional or border cases which may not be distinguishable from true noise, so a balance of both can be useful in a case editing algorithm.

Editing strategies normally operate in one of two ways; *incremental* which involves adding selected cases from the training set to an initially empty edited set and *decremental* which involves contracting the training set by removing selected cases.

An early competence preservation technique is Hart's Condensed Nearest Neighbour (CNN) [5]. CNN is an incremental technique which adds to an initially empty edited set any case from the training set that cannot be classified correctly by the edited set. This technique is very sensitive to noise and to the order of presentation of the training set cases, in fact CNN by definition will tend to preserve noisy cases. Ritter [6] reported improvements on the CNN with their Selective Nearest Neighbour (SNN) which imposes the rule that every case in the training set must be closer to a case of the same class in the edited set than to any other training case of a different class. Gates [7] introduced a decremental technique which starts with the edited set equal to the training set and removes a case from the edited set where its removal does not cause any other training case to be misclassified. This technique will allow for the removal of noisy cases but is sensitive to the order of presentation of cases. More recent improvements

to CNN have been proposed by Chou et al. [8] and Angiulli [9] with Hao et al. [10] proposing a variation appropriate for text classification.

Competence enhancement or noise reduction techniques start with Wilson's Edited Nearest Neighbour (ENN) algorithm [11], a decremental strategy, which removes cases from the training set which do not agree with their k nearest neighbours. These cases are considered to be noise and appear as exceptional cases in a group of cases of the same class.

Tomek [12] extended this with his repeated ENN (RENN) and his *all k-NN* algorithms. Both make multiple passes over the training set, RENN repeating the ENN algorithm until no further eliminations can be made from the training set while all k-NN uses incrementing values of k for each case and removes the case if a misclassification occurs for any value of k. These techniques focus on noisy or exceptional cases and do not result in the same storage reduction gains as the competence preservation approaches. A variation on ENN using k nearest centroid neighbours instead of k nearest neighbours was proposed by Sánchez et al. [13]. There is also work which considered relabelling examples rather than deleting them [14, 15].

Later editing techniques can be classified as hybrid techniques incorporating both competence preservation and competence enhancement stages. Aha [16] presented a series of Instance Based (IB) learning algorithms to reduce storage requirements and tolerate noisy instances. IB2 is similar to CNN adding only cases that cannot be classified correctly by the reduced training set. IB2's susceptibility to noise is handled by IB3 which records how well cases are classifying and only keeps those that classify correctly to a statistically significant degree. Other researchers have provided variations on the IBn algorithms [17–19].

More recently, Pan et al. [20] proposed a case-base mining algorithm Kernel-based Greedy Case-base Mining (KGCM) guided by theoretical results to edit a case-base. It involves using a kernel transformation to map the original case-base to a new feature space and using FDA to help remove noise and identify the predictive features. KGCM is an incremental approach that considers cases from this new space for addition based on their diversity.

Also, Massie et al.'s [21] recent work on case-base profiling introduces a decremental noise reduction strategy called Threshold Error Reduction (TER) that removes cases based on a complexity measure called the Friend:Enemy (F:E) Ratio. This measure compares the distances to a case's nearest like neighbours with distances to its nearest unlike neighbours. Their case-editing algorithm iteratively removes cases with F:E ratios higher than certain thresholds to remove noisy cases and to smooth out the boundary. They point out that different datasets are threshold dependent.

2.1 Competence-Based Case-Base Editing

More recent approaches to case-base editing build a competence model of the training data and use the competence properties of the cases to determine which cases to include in the edited set. Measuring and using case competence to guide case-base maintenance was first introduced by Smyth & Keane [1]. They

introduced two important competence properties, the *reachability* and *coverage* sets for a case in a case-base [1]. The *reachability set* of a case c is the set of all cases that can successfully classify c, and the *coverage set* of a case c is the set of all cases that c can successfully classify. The coverage and reachability sets represent the local competence characteristics of a case and are used as the basis of a number of editing techniques. Smyth & Keane first used these case competence properties in their Footprint Deletion policy which identified a series of case categories using these competence properties to provide a means of ordering cases for deletion. The *competence footprint* is a subset of the case-base that provides the same competence as the entire case-base.

McKenna & Smyth [22] later presented a family of competence-guided editing methods for case-bases which combine both incremental and decremental strategies. This family of algorithms is based on different combinations of policies for adding and removing cases, policies for presenting cases for consideration and for competence model update. These algorithms also include an RENN based initial pass to remove noise. Brighton & Mellish [3] also use the coverage and reachability properties of cases in their Iterative Case Filtering (ICF) algorithm. ICF is a decremental strategy contracting the training set by removing those cases c, where the number of other cases that can correctly classify c is higher that the number of cases that c can correctly classify. This strategy focuses on removing cases far from class borders. ICF also includes a pre-processing noise reduction stage, effectively RENN, to remove noisy cases.

Wilson & Martinez [23] presented a series of Reduction Technique (RT) algorithms which they later enhanced into the Decremental Reduction Optimisation Procedures (DROP) [4]. Although these were originally published before the definitions of coverage and reachability, they could also be considered to use a competence model. They define the set of *associates* of a case c which is comparable to the coverage set of McKenna and Smyth except that the associates set will include cases of a different class from case c whereas the coverage set will only include cases of the same class as c. The RTn and DROPn algorithms use a decremental strategy.

In contrast to the earlier approaches to noise reduction which tend to focus on removing the cases that are misclassified, Delany & Cunningham's Blame Based Noise Reduction (BBNR) [2] attempts to identify those cases causing the misclassifications and uses this information to identify training cases the case-base would be better off without. BBNR extends Smyth & Keane's case competence model by including an additional set, the *liability set* which is the set of all cases that c causes to be misclassified. This attempts to model the situation of a case being classified incorrectly because of the retrieved cases that contributed to its classification rather than the case being itself a noisy or mislabelled case.

3 Case Profiles

In this section we propose an approach to modelling the competence of a case-base with a view to categorising the competence of each case in the case-base. We then have the opportunity to investigate the effect that each type of case can have on case-base competence.

3.1 Enhanced Competence Model

Smyth & Keane's [1] case-base competence modelling approach proposed two sets to model the local competence properties of a case, the *reachability* set of a case c, the set of all cases that can successfully classify c and the *coverage set* of a case c, the set of all cases that c can classify. Using the case-base itself as a representative of the target problem space, these sets can be estimated as shown in Equations 1 and 2. Delany & Cunningham's [2] extension included an additional property; the liability set of a case c, the set of all cases that c causes to be misclassified and can be estimated by Equation 3.

$$ReachabilitySet(c \in C) = \{t \in C : Classifies(c, t)\} \quad (1)$$

$$CoverageSet(c \in C) = \{t \in C : Classifies(t, c)\} \quad (2)$$

$$LiabilitySet(c \in C) = \{t \in C : Misclassifies(t, c)\} \quad (3)$$

In the above equations $Classifies(t, c)$ means that case c contributes to the correct classification of target case t. This means that target case t is successfully classified and case c is returned as a nearest neighbour of case t and has the same classification as case t. $Misclassifies(t, c)$ means that case c contributes in some way to the incorrect classification of target case t. In effect this means that when target case t is misclassified by the case-base, case c is returned as a neighbour of t but has a different classification to case t. For k-NN with $k = 1$, case c causes the misclassification but for $k > 1$ case c contributes to the misclassification. Case t is therefore a member of the liability set of case c.

We propose to further extend the competence properties of a case to include an additional property, the *dissimilarity set*, which complements the reachability set in the same way as the liability set complements the coverage set. The dissimilarity set of a case c is the set of cases that misclassify case c and can be represented by Equation 4.

$$DissimilaritySet(c \in C) = \{t \in C : Misclassifies(c, t)\} \quad (4)$$

The first point to note about these sets is that one of the reachability set or the dissimilarity set will always be empty. In effect, if we consider that a set exists for a case only if that set is non empty, then the reachability set and the dissimilarity set are mutually exclusive. If a case has a non empty reachability set then it has been classified correctly by the case-base and, as such, will have an empty dissimilarity set and vice versa. However, a case can have one, none

or both of the coverage and liability sets. The coverage set of a case c identifies the potential benefit or usefulness of c in the case-base, represented by the cases that c contributes to classifying correctly. On the other hand the liability set of a case c identifies the damage or harm that c causes in the case-base represented by the cases that it causes to be misclassified. It is possible for a case to be both useful for some targets and damaging for others.

3.2 Categorising Cases

This leads us to being able to associate an individual case profile with each case in a case-base. We call the profile the *RDCL* profile of a case; it is derived from the competence model of the case-base and includes three characteristics:

(i) Firstly, it is possible to indicate whether the case is correctly or incorrectly classified by the case-base. This is identified by the case having either a reachability set (R) or a dissimilarity set (D).
(ii) Secondly, we can consider whether the case is useful, by the existence of a coverage set (C), and
(iii) Finally whether the case is harmful and causes damage, by the existence of a liability set (L).

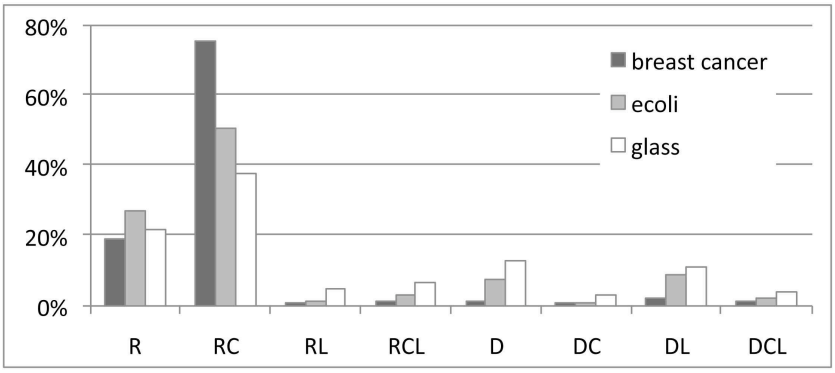

Fig. 1. Composition of datasets with different generalisation accuracies showing the proportion of cases of each profile type. *Breast Cancer* has 95% 10-fold cross validation accuracy, *ecoli* has 81% accuracy while *glass* has 66% accuracy.

Taking all possible combinations of these characteristics, each case in a case-base can have one of eight different case profiles as described below. Fig 1 helps to interpret these by giving the proportion of the different case profiles in different case-bases.

R A case which is correctly classified but is not used for classifying any other case in the case-base.

D A case which is misclassified but is not used for classifying any other case in the case-base.

RC A case which is correctly classified and is useful in that it has contributed to the correct classification of other cases in the case-base. This profile and the R profile are generally the majority case profile types as illustrated in Fig 1.

RL A case which is correctly classified but is harmful in the case-base causing damage by contributing to other cases being misclassified.

DC A case which is misclassified but is useful in the case-base.

DL A case which is misclassified and is harmful in the case-base, (more of these occur in the case-base with poorer generalisation accuracy in Fig 1).

RCL A case which is correctly classified and is both useful and harmful in the case-base.

DCL A case which is misclassified and is both useful and harmful in the case-base.

4 Experimental Analysis

The ability to associate a competence case profile with each case in a case-base offers the opportunity to investigate the structure of case-bases at a case level and the effect of removing different types of cases from a case-base. This section outlines a number of different investigations and evaluations performed using case profile information on a variety of datasets. The datasets used throughout this paper are listed in Table 1 with a description of their characteristics. All datasets are available in the UCI repository [24].

Table 1 also includes the baseline 10-fold cross validation accuracy achieved on each dataset using a k-NN classifier with a Euclidean distance measure and $k = 1$. As the objective is to consider the effect of different case editing strategies, $k = 1$ was selected as the effect of noise in the data will be more evident with this value since higher values of k are more noise tolerant.

4.1 Removal of Different Types of Cases

Table 2 shows the effect of removing all the cases of each different type of case profile from each dataset. The accuracy was calculated using 10-fold cross validation, using the same folds as the original baseline accuracy given in Table 1. A competence model was build on each training set of nine folds and the training set was edited to remove all cases with the specified case profile. The cases in the remaining fold were classified using the edited training set.

Fig 2 shows the difference, averaged across all datasets, of removing cases with the specified profile. This figure illustrates some interesting facts which are discussed below:

Table 1. Datasets

	#cases	#classes	#features	class distribution(%)	cv accuracy(%)
breastcancer	683	2	9	65/35	95.5
cmc	1473	3	9	28/16/56	43.3
glass	214	6	9	33/8/35/6/4/14	65.9
musk2	6598	2	166	15/85	73.8
waveform	5000	3	21	33/33/33	77.2
spectf	267	2	44	21/79	71.5
ecoli	336	8	7	42/23/15/10/6/2/1/1	81.0
wine	178	3	13	33/40/27	94.9
ionosphere	351	2	34	68/32	86.0
hill-valley	1212	2	100	50/50	59.4

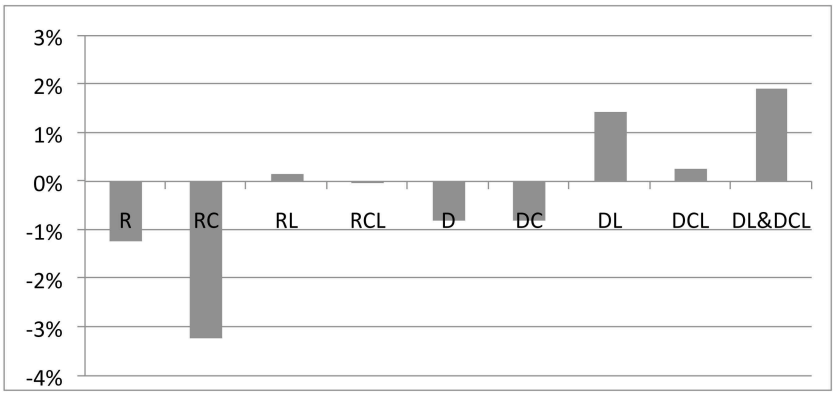

Fig. 2. Accuracy difference, averaged across all datasets, between the unedited dataset and the dataset edited with all cases of the specified case profile removed.

DL/DCL cases: The cases that cause the greatest improvement in accuracy by their removal are the DL cases. DL cases are cases that are misclassified by the rest of the case-base and are themselves also causing harm as they are misclassifying other cases. It is to be expected that removing cases such as these would have a beneficial effect on the generalisation accuracy. Considering the individual results in Table 2, the removal of the DL cases is damaging in just one of the datasets where the decrease in accuracy is less than 1%.

DCL cases are somewhat similar to DL cases. The only difference is that these cases, in spite of being misclassified and doing harm, also do some good in that they are used to correctly classify other cases. Looking at the individual dataset results the removal of these cases doesn't typically decrease generalisation accuracy, in all cases but one accuracy remains constant or increases. Overall the

Table 2. Accuracy values (%) on editing the datasets by removing cases with a specific case profile. Accuracy values which show an increase over the baseline are highlighted in bold. Decreases in generalisation accuracy are in italic.

Dataset	R	RC	RL	RCL	D	DC	DL	DCL	DL&DCL
breastcancer	95.5	*94.4*	**95.9**	95.5	**95.9**	**95.6**	**96.1**	95.5	**95.9**
cmc	*42.6*	*42.8*	*42.9*	*42.8*	*42.8*	*42.7*	**46.2**	**44.1**	**47.0**
glass	65.9	*62.2*	**66.4**	*64.5*	*65.4*	65.9	*65.0*	65.9	*65.4*
musk2	*72.4*	*70.5*	**74.0**	**74.5**	**74.9**	*73.7*	**74.9**	73.8	**74.8**
waveform	*76.7*	*74.6*	77.2	*76.9*	77.2	*77.0*	**78.0**	**77.6**	**78.5**
spectf	*70.4*	*67.8*	**71.9**	**72.7**	**72.7**	*70.8*	**73.4**	71.5	**74.2**
ecoli	*78.1*	*77.1*	*80.4*	**81.0**	**82.1**	*80.1*	**83.3**	81.0	**83.0**
wine	*93.8*	*93.8*	94.9	94.9	*94.4*	94.9	94.9	94.9	94.9
ionosphere	**87.2**	*81.5*	**86.3**	**86.3**	*85.8*	*84.6*	**86.3**	**86.6**	**86.9**
hill-valley	*57.7*	*59.3*	**59.7**	**59.7**	*53.1*	*57.8*	**59.5**	*59.1*	**59.7**

effect is beneficial, albeit marginal. This suggests that we are better off without these cases.

Considering the beneficial effect of removing DC and DCL cases separately, Fig 2 and Table 2 also include the results of removing both DC and DCL cases from each dataset. This only has the effect of decreasing, marginally, the accuracy for one dataset, glass. It also results in a higher average increase than removing cases of either profile.

R cases: Cases with an R profile are cases that are classified correctly by the rest of the case-base but are not used in the classification of any other case in that case-base. This suggests that these cases are redundant cases and are not needed. Removing such cases should show no change in generalisation accuracy. Interestingly enough, the removal of these cases causes a decrease in generalisation accuracy on average, with only one of the datasets showing an actual increase in accuracy. This indicates that these cases, although not used to classify the training data, are useful in the classification of unseen data and necessary to maintain good generalisation accuracy. This suggests that R cases may be outlier cases, not well covered by other cases and should not be removed from the case-base.

D/DC cases: Cases with a D profile are cases that are misclassified by the rest of the case-base but are not used in the classification of any other case in that case-base. Corresponding to the reasoning above for the R cases, it might be considered that these cases are redundant cases, as they are not used for classification and could be removed. As these cases are also misclassified, there is an even stronger impetus to remove them from the case-base. However, as Fig 2 shows, overall the removal of these cases has a surprisingly detrimental effect on

the generalisation accuracy. In only four of the datasets is the generalisation accuracy increased. This suggests, analogous to the R cases, that D cases may be border cases, situated near the decision boundary which are not well covered by other cases in the case-base.

DC cases are cases that are themselves misclassified by the case-base but that are useful as they contribute to the correct classification of other cases. Similar to the D cases, the removal of these in general decreases the generalisation accuracy of the case-base with a marginal increase of 0.1% shown in only one dataset. This suggests that these cases should not be deleted.

It is also interesting to note that both these types of case, D and DC, are removed by the standard Wilson noise reduction technique used by many case editing algorithms.

RC cases: RC cases are cases that are correctly classified and are used to correctly classify other cases. It is to be expected that the removal of these cases would have a detrimental effect on the generalisation accuracy which we can see is the case. In fact all datasets show a decrease in generalisation accuracy by removing RC cases, with some very significant harmful effects including a drop of over 5% for the ionosphere, spectf and glass datasets.

RL/RCL cases: RL cases are classified correctly but cause harm contributing to the misclassification of other cases. RCL cases are the same but also do good by contributing to the correct classification of other cases. This suggests that these cases are border cases, but considering Fig 2 and Table 2 there is no strong evidence to support the removal of such cases.

It is interesting to note here that both these types of cases, RL & RCL, are removed by another noise reduction algorithm, the BBNR algorithm.

4.2 What existing noise reduction algorithms do

The original ENN noise reduction technique proposed by Wilson [11] is the algorithm upon which the noise reduction phases of many of the existing case-base editing techniques are based, with a number of them using RENN as a noise reduction stage [3, 22, 23]. Wilson's technique removes cases that would be misclassified by the other cases in a training set, assuming that these are incorrectly labelled and are therefore noisy cases. In terms of our profiling, this would be cases that have a dissimilarity set. The later BBNR approach [2] focusses more on the 'unhelpful' or harmful cases that *cause* misclassification, i.e. the cases with a liability set. The differences between the principles behind these approaches is evident in Fig 3 which shows for both the RENN and the BBNR algorithms the proportion of deleted cases that were of each case profile type, averaged across all datasets.

RENN removes all cases which are misclassified, which means they contain a D in their profile as they have a dissimilarity set (i.e. it removes 100% of cases with a D, DC, DC or DCL profile). As RENN is an iterative algorithm repeating

until no more changes are made to the dataset, it can also remove cases with profiles other than those containing D. This can be seen in Fig 3 where on average 7.3% of the deleted cases have profile R with smaller percentages for RL, RC and RCL cases. Overall, RENN removes on average 10% of the total R cases, 20% of the RL cases, less than 1% of the RC cases and almost 2% of the RCL cases. A high proportion of the cases removed by RENN are, on average, D or R cases (just over 50%); we saw from our analysis of case profiles that removing these cases tends on average to have a bad effect on generalisation accuracy.

BBNR removes all cases which do harm, meaning they have L in their profile, as they contain a liability set. It is obvious from Fig 3 that there is only a small likelihood of overlap in the cases deleted by the two algorithms. If we follow our conclusions from section 4.1 above, BBNR focusses more on the types of cases that are beneficial to generalisation accuracy than RENN (59% of the deleted cases are DL and DCL cases for BBNR whereas 39% are for RENN). In section 4.3 below, we investigate the effect of implementing both algorithms on the selected datasets and will see that RENN has quite inconsistent behaviour which may be due to its focus on case types which are not beneficial to generalisation accuracy.

It is also worth noting that neither algorithms removes the RC cases, which our investigations show would have a bad impact on accuracy if removed.

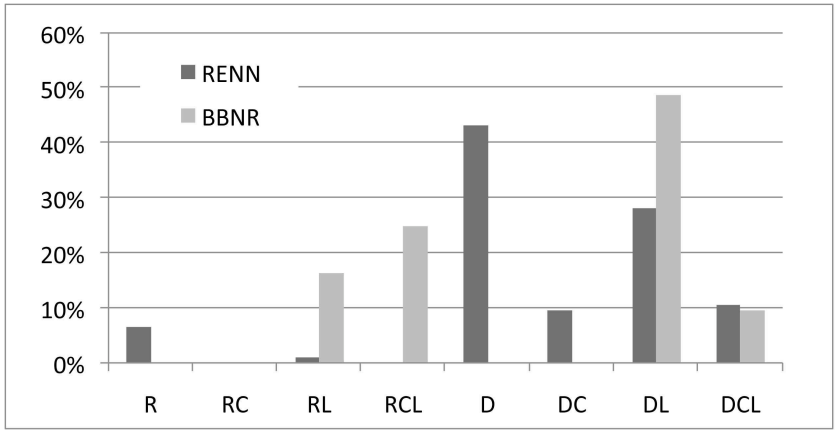

Fig. 3. Proportion of deleted cases that are of the specified profile for each noise reduction algorithm averaged across all datasets.

The BBNR algorithm is straightforward in dealing with cases that just do damage, regardless of how they are classified. These cases, DLs and RLs are simply removed. Where a case has a liability set but also a coverage set the BBNR algorithm adopts the principle of 'not causing even more harm'. BBNR only removes a DCL or RCL case if its removal will not cause even more harm, in other words all cases in its coverage set will still be classified correctly if the

DCL or RCL case is removed. This is evident in our experiments, as although 100% of DL and RL cases are removed by BBNR, only on average 55% and 61% of DCL and RCL cases respectively are removed.

We ran an experiment to see the effect of this 'not causing even more harm' principle. We only removed a DCL or RCL case if its removal did not result in any of its coverage set being misclassified. The overall results are displayed in Fig 4. It shows that the effect of keeping some of the DCL and RCL cases, those that cause even more harm (labelled v2 in Fig 4) has a worse effect, albeit small, than removing all of them. This raises questions about the benefit of this aspect of the BBNR algorithm.

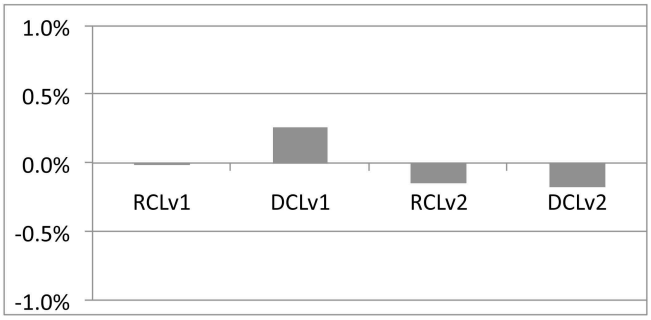

Fig. 4. Percentage accuracy difference, averaged across all datasets, between the unedited datasets and the datasets edited with the specified case profile removed; v1 figures show the result of deleting all cases with the specified profile, v2 figures show the results of keeping those profile cases that cause even more harm by their removal.

4.3 Comparison of editing algorithms

The analysis shown in Fig 2 suggests another noise reduction algorithm, the identification and removal of the DL and DCL cases in a dataset. In this section we compare the performance of existing noise reduction algorithms, RENN and BBNR and this new algorithm. Table 3 presents the results of a 10-fold cross validation accuracy (on the same folds) across these noise reduction algorithms. The statistical significance of the differences in accuracy of each algorithm against the unedited alternative was calculated using McNemar's test [25]. Where there were small levels of disagreement (15 cases or less) an exact sign test was used.

The main point to note here is the more consistent performance of removing just DL & DCL cases from the datasets. In just one dataset the generalisation accuracy of the dataset reduces; in all others but one the accuracy actually increases. Compare this with the performance of RENN. For a number of the datasets the generalisation accuracy is in fact higher for RENN than for DL&DCL, with a highest difference of 2.7% for the ecoli dataset. However the

Table 3. Comparison between different noise reduction algorithms. Algorithms which show an increase in accuracy over the unedited case-base are highlighted in bold. Decreases in generalisation accuracy are in italic. Differences significant at the 95% level using McNemar's test [25] are highlighted with an asterisk.

	Unedited cv accuracy (%)	**RENN cv accuracy (%)**	**BBNR cv accuracy (%)**	**DL & DCL cv accuracy (%)**
breastcancer	95.5	**96.6***	**96.5***	**95.9**
cmc	43.3	**46.1***	**45.1***	**47.0***
glass	65.9	*63.1*	*65.4*	*65.4*
musk2	73.8	**76.0***	**75.1***	**74.8***
waveform	77.2	**78.9***	**78.4***	**78.5***
spectf	71.5	**75.3**	**71.9**	**74.2**
ecoli	81.0	**85.7***	**82.7**	**83.0***
wine	94.9	*93.8*	94.9	94.9
ionosphere	86.0	*84.9*	**86.9**	**86.9**
hill-valley	59.4	*48.5**	*58.3*	**59.7**
Avg Diff		-0.1%	1.0%	1.9%

performance of the RENN algorithm is not consistent with considerable decreases in generalisation accuracy for a number of datasets, including a significant 18% drop for hill-valley and more than 3% for glass and ionosphere.

The performance of BBNR seems better, recording only a lower generalisation accuracy over the unedited case-base in two datasets. However, removing DC & DCL cases records equivalent or higher generalisation accuracy than BBNR for all but one of the datasets.

5 Conclusions & Future Work

A methodology for categorising cases in a case-base into individual case profile types was presented in this paper. The profile is based on three characteristics that are derived from constructing a competence model of the case-base. These characteristics indicate whether the case has been classified correctly or not, whether the case is helpful to the case-base by its inclusion and/or whether it causes damage in the case-base. Using these profiles we investigated the effect of removing the different types of cases from the case-base. Based on this analysis a simple noise reduction algorithm based on case profiles, was proposed which was more consistent at improving generalisation accuracy on a number of evaluation datasets that the other noise reduction techniques considered.

The ability to categorise cases within a casebase allowed the identification of the types of cases that are removed by existing noise reduction algorithms. Knowing the effect of removing different types of cases, we were able to identify

the aspects of the algorithms that contribute to good performance and those that do not.

The work presented in this paper offers opportunities for further work in a number of directions. With a case profiling strategy available, we hope to gain further insights into other case editing algorithms, such as Massie et al.[21]'s TER algorithm, investigating exactly which types of cases are removed. We would like to investigate further into the case profiles, by quantifying the good and harm performed by the cases to allow prioritising cases within a profile for deletion.

Acknowledgements

The author is grateful to Pádraig Cunningham for discussions about this work. This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/RFP/CMSF718.

References

1. Smyth, B., Keane, M.: Remembering to forget: A competence preserving case deletion policy for CBR systems. In Mellish, C., ed.: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI (1995), Morgan Kaufmann (1995) 337–382
2. Delany, S.J., Cunningham, P.: An analysis of case-based editing in a spam filtering system. In Funk, P., González-Calero, P., eds.: 7th European Conference on Case-Based Reasoning (ECCBR 2004). Volume 3155 of LNAI., Springer (2004) 128–141
3. Brighton, H., Mellish, C.: Advances in instance selection for instance-based learning algorithms. Data Mining and Knowledge Discovery **6** (2002) 153–172
4. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning **38** (2000) 257–286
5. Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory **14** (1968) 515–516
6. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An algorithm for a selective nearest neighbor decision rule. IEEE Transactions on Information Theory **21** (1975) 665–669
7. Gates, G.W.: The reduced nearest neighbor rule. IEEE Transactions on Information Theory **18** (1972) 431–433
8. Chou, C.H., Kuo, B.H., Chang, F.: The generalized condensed nearest neighbor rule as a data reduction method. In: ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition, Washington, DC, USA, IEEE Computer Society (2006) 556–559
9. Angiulli, F.: Fast nearest neighbor condensation for large data sets classification. IEEE Transactions on Knowledge and Data Engineering **19** (2007) 1450–1464
10. Hao, X., Zhang, C., Xu, H., Tao, X., Wang, S., Hu, Y.: An improved condensing algorithm. In: Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on. (2008) 316–321
11. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics **2** (1972) 408–421
12. Tomek, I.: An experiment with the nearest neighbor rule. IEEE Transactions on Information Theory **6** (1976) 448–452

13. Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters **24** (2003) 1015–1022
14. Jiang, Y., Zhou, Z.: Editing training data for knn classifiers with neural network ensemble. In: Lecture Notes in Computer Science, Vol.3173. (2004) 356–361
15. Koplowitz, J., Brown, T.A.: On the relation of performance to editing in nearest neighbor rules. Pattern Recognition **13** (1981) 251–255
16. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6** (1991) 37–66
17. Brodley, C.: Addressing the selective superiority problem: Automatic algorithm/mode class selection. In: Proceedings of the 10th International Conference on Machine Learning (ICML 93), Morgan Kaufmann Publishers Inc. (1993) 17–24
18. Cameron-Jones, R.M.: Minimum description length instance-based learning. In: Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers Inc. (1992) 368–373
19. Zhang, J.: Selecting typical instances in instance-based learning. In: Proceedings of the 9th International Conference on Machine Learning (ICML 92), Morgan Kaufmann Publishers Inc. (1992) 470–479
20. Pan, R., Yang, Q., Pan, S.J.: Mining competent case bases for case-based reasoning. Artificial Intelligence **171** (2007) 1039–1068
21. Massie, S., Craw, S., Wiratunga, N.: When similar problems don't have similar solutions. In: ICCBR '07: Proceedings of the 7th international conference on Case-Based Reasoning, Berlin, Heidelberg, Springer-Verlag (2007) 92–106
22. McKenna, E., Smyth, B.: Competence-guided editing methods for lazy learning. In Horn, W., ed.: ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, IOS Press (2000) 60–64
23. Wilson, D., Martinez, T.: Instance pruning techniques. In: ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc. (1997) 403–411
24. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
25. Dietterich, D.T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computing **10** (1998) 1895–1923