# Learning from Imbalanced Sets Through Resampling and Weighting *

R. Barandela[1,4], J.S. Sánchez[2], V García[1], and F.J. Ferri[3]

[1] Instituto Tecnológico de Toluca, Av. Tecnológico s/n, 52140 Metepec, México
[2] Dept. Llenguatges i Sistemes Informàtics, U. Jaume I, 12071 Castelló, Spain
[3] Dept. d'Informàtica, U. València, 46100 Burjassot (València), Spain
[4] Instituto de Geografía, Vedado, La Habana, Cuba

**Abstract.** The problem of imbalanced training sets in supervised pattern recognition methods is receiving growing attention. Imbalanced training sample means that one class is represented by a large number of examples while the other is represented by only a few. It has been observed that this situation, which arises in several practical situations, may produce an important deterioration of the classification accuracy, in particular with patterns belonging to the less represented classes. In the present paper, we introduce a new approach to design an instance-based classifier in such imbalanced environments.

## 1 Introduction

Design of supervised pattern recognition methods is based on a training sample (TS), that is, a collection of examples previously analyzed by a human expert. Performance of the resulting classification system depends on both the quantity and the quality of the information contained in the TS. This dependency is particularly strong in the case of non-parametric classifiers since these systems do not rest upon any probabilistic assumption about the class models. Researchers have very early realized that the TS must satisfy some requirements in order to guarantee good classification results. From the start, two assumptions were established: 1) the set of $c$ classes present in the TS covers the whole space of the relevant classes, and 2) the training instances used to teach the classifier how to identify each class are actually members of that class.

As the number of practical applications of these methods grows, experience has gradually indicated the necessity of some requisites for the system to reach satisfactory results. Among others, one can remark: 3) the TS represents the population, 4) the considered features must permit discrimination, and 5) the size/dimensionality rate of the sample is high enough.

An additional and interesting complication arises when the TS is *imbalanced*. A TS is said to be imbalanced if one of the classes is represented by a very small number of instances compared to the other classes. Throughout this paper,

* Partially supported by grants 32016-A (Mexican CONACyT), TIC2000-1703-C03-03 (Spanish CICYT), and P1-1B2002-07 (Fundació Caixa Castelló-Bancaixa).

and consistently with the common practice [10, 16], we consider only two-class problems and therefore, the examples are said to be either positive or negative (that is, either from the minority class or the majority class, respectively). It has been observed that class imbalance may cause a significant deterioration in the performance attainable by standard supervised methods. High imbalance occurs in real-world domains where the decision system is to detect a rare but important case, such as fraudulent telephone calls [12], oil spills in satellite images [17], an infrequent disease [24], or text categorization [18, 20].

Most of the research efforts addressing this problem can be organized into three categories. One is to assign distinct costs to the classification errors for positive and negative examples [8, 14]. The second is to resample the original TS, either by over-sampling the minority class [19] and/or under-sampling the majority class [16] until the classes are approximately equally represented. The third consists in internally biasing the discrimination-based process so as to compensate for the class imbalance [11, 12, 21].

In an earlier study [2], we provided preliminary results of several techniques addressing the class imbalance problem. In such a work, we focused on resampling (by under-sampling the majority class) the TS and also on internally biasing the discrimination process, as well as on a combination of both methods. In the present paper, we introduce a new approach for a better and higher decrease in the number of negative examples. The technique proposed here is evaluated over four real datasets using a Nearest Neighbour (NN) classifier [6].

## 2 Related Works

The two basic methods for resampling the TS cause the class distribution to become more balanced. Nevertheless, both strategies have shown important drawbacks. Under-sampling throws out potentially useful data, while over-sampling increases the TS size and hence the time to design a classifier. Furthermore, since over-sampling typically replicates examples in the minority class, overfitting is more likely to occur. In the last years, research has focused on improving these basic methods. Kubat and Matwin [16] proposed an under-sampling technique that intelligently removes only those negative instances that are "redundant" or that "border" the minority prototypes (they assume that these bordering cases are noisy examples).

Chawla et al. [4] combine under-sampling and over-sampling methods and, instead of over-sampling by merely replicating positive prototypes, they form new minority instances by interpolating between several positive examples that lie close together. On the other hand, Chan and Stolfo [3] first run preliminary experiments to determine the best class distribution for learning and then generate multiple TSs with such a distribution. This is accomplished by including all the positive examples and some of the negative prototypes in each TS. Afterwards, they run a learning algorithm on each of the datasets and combine the induced classifiers to form a composite learner. This method ensures that all

of the available training instances are used, since each negative example will be found in at least one of the TSs.

Pazzani et al. [21] take a slightly different approach when learning from an imbalanced TS by assigning different weights to prototypes of the different classes. On the other hand, Ezawa et al. [11] bias the classifier in favour of certain attribute relationships. Kubat et al. [17] use some counter-examples to bias the recognition process.

In a previous work [2], we presented some methods for under-sampling the majority class in the TS and a technique for biasing the classification procedure. Since downsizing the majority class can result in throwing away some useful information, this size reduction must be done carefully. One should be interested in using the removal of negative examples to eliminate the less valuable prototypes, that is, noisy or atypical cases, instances that are close to the decision boundaries, and redundant examples. For these purposes, we employed several well-known editing and condensing schemes [7] that offer a good alternative for removing all these examples. In [2], we tried three prototype selection algorithms. Two of them are in the group of editing: the classical Wilson's proposal [23] and the $k$-NCN (Nearest Centroid Neighbourhood) scheme [13]. Both aim at filtering the TS by deletion of noisy or atypical instances, generally increasing the NN accuracy. These two techniques were also used in an iterative manner.

For elimination of redundant prototypes, we have employed the Modified Selective (MS) [1] condensing. This method is based on the idea of creating a consistent subset [15], and guarantees a suitable approximation to the NN decision boundaries as they are defined by the whole TS. Finally, employment of the combined editing-condensing (Wilson + MS and $k$-NCN + MS) was also proposed as a way of downsizing the majority class in the TS to balance the class distribution.

For internally biasing the discrimination procedure, we proposed in [2] a weighted distance function to be used in the classification phase. Let $d_E(\cdot)$ be the Euclidean metric, and let $Y$ be a new sample to be classified. Let $x_0$ be a training prototype from class $i$, let $N_i$ be the number of prototypes from class $i$, let $N$ be the TS size, and let $m$ be the dimensionality of the feature space. Then, the weighted distance measure is defined as:

$$d_W(Y, x_0) = (\tfrac{N_i}{N})^{1/m} \cdot d_E(Y, x_0)$$

The basic idea behind this weighted distance is to compensate for the imbalance in the TS without actually altering the class distribution. Thus, weights are assigned, unlike in the usual weighted $k$-NN rule [9], to the respective classes and not to the individual prototypes. In such a way, since the weighting factor is higher for the majority class than for the minority one, the distance to positive instances becomes much lower than the distance to negative examples. This produces a tendency for the new patterns to find their nearest neighbour among the prototypes from the minority class.

## 3 Classifier Performance in Class Imbalance Problems

To evaluate the performance of learning systems, a confusion matrix like the one in Table 1 (for a two-class problem) is usually employed. The elements in this table characterize the classification behaviour of the given system. The columns are the actual class and the rows correspond to the predicted class. The sum of the two columns gives the total number of samples in each class which is $n^+ = TP + FN$ and $n^- = FP + TN$, respectively.

**Table 1.** Confusion matrix.

| | Actual Positive | Actual Negative |
|---|---|---|
| Predict Positive | True Positive (TP) | False Positive (FP) |
| Predict Negative | False Negative (FN) | True Negative (TN) |

The standard evaluation measure in pattern recognition domain is the classification accuracy, defined as $acc = \frac{TP+TN}{n^+ + n^-}$. However, this form of classification accuracy assumes that the error costs (that is, the cost of a false positive and a false negative) are equal. This assumption has been criticized as being unrealistic. For instance, consider a domain where only 0.2% patterns are positive. In such a situation, labeling all new patterns as negative would give an accuracy of 99.8%, but failing on all positive cases. Classifiers that optimize for accuracy in these problems are of questionable value since they rarely predict the minority class. Consequently, in the presence of imbalanced datasets, it is more appropriate to use other performance measures.

Alternative criteria for evaluating classifier performance include ROC curves [22] and the geometric mean of accuracies [16]. These are good indicators of performance on imbalanced datasets because they are independent of the distribution of prototypes between classes, and are thus robust in circumstances where such a distribution might change with time or be different in the training and test sets. In particular, the geometric mean of accuracies measured separately on each class [16] is defined as $g = \sqrt{acc^+ \cdot acc^-}$, where $acc^+ = \frac{TP}{n^+}$ is the accuracy on the positive examples, and $acc^- = \frac{TN}{n^-}$ denotes the accuracy on the negative examples. This measure closely relates with the distance to perfect classification in the ROC space.

The rationale behind this measure is to maximize the accuracy on each of the two classes while keeping these accuracies balanced. For instance, a high $acc^+$ by a low $acc^-$ will result in a poor $g$ value. The $g$ measure has the distinctive property of being nonlinear, that is, a change in $acc^+$ (or $acc^-$) has a different effect on $g$ depending on the magnitude of $acc^+$: the smaller the value of $acc^+$, the greater the change of $g$. This means that the cost of misclassifying each positive pattern increases the more often positive examples are misclassified.

In this work, the $g$ criterion will be used to evaluate the learning algorithms both because the interesting general properties of $g$ and also because the pro-

posed classifiers do not directly have a changing parameter which properly justifies a ROC analysis.

## 4 The Weighted Wilson's Editing

As already explained, we have experimented with several methods [2] aimed at reducing the size of the majority class. Out of concern for the possibility of eliminating useful information, we have used the well-known Wilson's editing algorithm [23]. One of the contributions of our previous paper to the imbalance problem has been the application of this editing technique only to the majority class. Another idea also explored in [2] is the employment of a weighted distance when looking for the nearest prototype of a new pattern to be classified. Both proposals have produced a significant increase in performance.

Despite these important results, it was observed in [2] that the editing technique does not produce significant reductions in the size of the majority class. Accordingly, the imbalance in the TS is not diminished in an important way. It is worthy to consider that Wilson's technique essentially consists in a sort of classification system. The corresponding procedure works by applying the $k$-NN classifier to estimate the class label of all prototypes in the TS. Afterwards, those prototypes whose class label does not agree with the associated with the largest number of the $k$ neighbours are discarded.

Of course, the $k$-NN classifier is also affected by the imbalance problem. When applied to prototypes from the majority class, the imbalance in the TS will cause a tendency to find most of their $k$ neighbours into that majority class. Consequently, only a few of the negative instances will be removed from the TS. This means that the majority class is not completely cleaned of atypical cases and also that the balance in the TS is far from being reached.

To cope with this difficulty, in the present paper we introduce the employment of the weighted distance previously mentioned, not only in the classification phase but also in editing the majority class. That is, we apply the Wilson's editing procedure, but using the weighted distance function instead of the Euclidean metric. In such a way, the already explained tendency will be overturned.

This proposal is assessed with experiments carried out over four datasets from the UCI Database Repository (`http://www.ics.uci.edu/~mlearn/`). Five-fold cross-validation is used to obtain averaged results of the $g$ criterion. Some datasets have required to be transformed into two-class problems, both to have a minority class and also to facilitate comparison with other published results [16].

The experimental results are shown in Table 2. The average $g$ values obtained when classifying with the original TS, and with this TS after being processed with the idea of Kubat and Matwin [16], are also included for comparison purposes. Weighted editing of the majority class yields an improvement in performance (as measured by the $g$ criterion). This improvement is more remarkable when the weighted distance is employed both in editing and classification. It is also important to note that the results from the procedure of Kubat and Matwin are excelled in all datasets.

**Table 2.** Average $g$ value by processing the majority class.

| | Phoneme | Satimage | Glass | Vehicle |
|---|---|---|---|---|
| Original TS | 73.8 | 70.9 | 86.7 | 55.8 |
| Euclidean editing and classification | 74.9 | 73.0 | 86.2 | 64.0 |
| Euclidean editing and weighted classif. | 75.7 | 76.2 | 87.9 | 65.8 |
| Weighted editing and Euclidean classif. | 75.0 | 74.5 | 86.2 | 65.6 |
| Weighted editing and classification | 75.3 | 77.8 | 87.9 | 67.2 |
| Kubat and Matwin | 74.4 | 71.7 | 86.4 | 61.0 |

The effects of the weighted Wilson's editing can be better analyzed by considering the number of negative examples that remain in the TS after its application (see Table 3). Results in this table suggest a higher decrease in the size of the majority class when it is processed with the weighted editing.

**Table 3.** Average size before and after processing the majority class.

| | Phoneme | Satimage | Glass | Vehicle |
|---|---|---|---|---|
| Original TS | 3,054.0 | 4,647.0 | 150.0 | 508.0 |
| After Euclidean editing | 2,882.8 | 4,471.6 | 147.2 | 414.8 |
| After weighted editing | 2,729.8 | 4,320.6 | 144.6 | 392.0 |

On the other hand, there is no reason to consider that the minority class is free from atypical prototypes, which certainly affect the classifier performance. However, none of the previously published works has reported attempts to eliminate noisy positive examples. Because of the relative small size of the minority class, positive prototypes are considered as very important and therefore, elimination of some of them is usually regarded as a very risky undertaking.

To explore the convenience of editing also the minority class, we have done some experiments applying the usual and the weighted editings to both classes simultaneously. In these experiments, both editing procedures have been applied only once since more iterations may lead to removal of all examples in the minority class. As can be seen in Table 4, both editing methods have produced an increase in the imbalance between the classes, although this increment is patently lower when the weighted editing was applied.

Despite this imbalance intensification, weighted editing of both classes produces enhancement of the $g$ values, when compared with the usual editing technique (see Table 5). This is particularly true when the weighted distance is also employed to classify new patterns. These results indicate that the weighted distance for classification is able to cope with the imbalance increase (with the weighted editing) when it is moderate, as in Phoneme and Glass databases. In these datasets, the $g$ values now obtained exceed the best results shown in Table 2 (editing only the majority class).

**Table 4.** Majority to minority ratio when both classes are processed.

| | Phoneme | Satimage | Glass | Vehicle |
|---|---|---|---|---|
| Original TS | 2.41 | 9.29 | 6.25 | 2.99 |
| After Euclidean editing | 2.85 | 12.06 | 8.00 | 6.90 |
| After weighted editing | 2.52 | 10.37 | 7.23 | 5.49 |

**Table 5.** Average $g$ values when processing both classes.

| | Phoneme | Satimage | Glass | Vehicle |
|---|---|---|---|---|
| Euclidean editing and classification | 73.8 | 66.4 | 84.6 | 47.5 |
| Euclidean editing and weighted classif. | 76.7 | 69.5 | 86.4 | 51.5 |
| Weighted editing and Euclidean classif. | 75.1 | 70.1 | 84.6 | 52.3 |
| Weighted editing and classification | 76.4 | 72.2 | 88.7 | 56.1 |

## 5 Concluding Remarks and Further Work

In some real-world applications, the learning system has to work with just a few positive examples and a great number of negative instances. Traditional learning systems such as the NN rule can be misled when applied to such practical problems. This effect can become moderate by using some simple prototype selection techniques to under-sample the majority class and/or some kind of weighted distance to compensate the imbalance. In these directions, a new approach has been proposed in this paper. The idea of employing a weighted distance when editing the majority class has yield promising results: majority class gets a higher size reduction and the resulting TS is better cleaned from atypical prototypes.

The issue of cleaning also the minority class, through removal of noisy and redundant prototypes, deserves further attention. The resulting increase in the imbalance when both classes are processed may be diminished if the minority class is over-sampled after the application of the editing procedure. In our paper, we have shown that, when this increase is moderate, employment of the weighted distance in the classification stage is able to obtain accuracy improvement.

Despite the successful results, a problem common to most of the downsizing techniques is that they do not permit control on the number of prototypes to be removed. Therefore, eliminated examples can be too many or too few to adequately solve the class imbalance problem. Hence, experimentation with schemes that allow to control the number of resulting examples [5] could be of interest.

## References


1. Barandela, R., Cortés, N., Palacios, A.: The nearest neighbor rule and the reduction of the training sample size, In: *Proc. 9th Spanish Symp. on Pattern Recognition and Image Analysis* **1** (2001) 103-108.
2. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for learning in class imbalance problems, *Pattern Recognition* **36** (2003) 849-851.

3. Chan, P., Stolfo, S.: Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection, In: *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining* (1998) 164-168.
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research* **16** (2000) 321-357.
5. Chen, C.H., Józwik, A.: A sample set condensation algorithm for the class sensitive artificial neural network, *Pattern Recognition Letters* **17** (1996) 819-823.
6. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification, *IEEE Trans. on Information Theory* **13** (1967) 21-27.
7. Dasarathy, B.V.: *Nearest Neighbor Norms: NN Pattern Classification Techniques*, IEEE Computer Society Press, Los Alamos, CA, 1991.
8. Domingos, P.: Metacost: a general method for making classifiers cost-sensitive, In: *Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining* (1999) 155-164.
9. Dudani, S.A.: The distance-weighted $k$-nearest neighbor rule, *IEEE Trans. on Systems, Man, and Cybernetics* **6** (1976) 325-327.
10. Eavis, T., Japkowicz, N.: A recognition-based alternative to discrimination-based multi-layer perceptrons, In: *Advances in Artificial Intelligence* **LNCS 1822**, Springer-Verlag (2000) 280-292.
11. Ezawa, K.J., Singh, M., Norton, S.W.: Learning goal oriented Bayesian networks for telecommunications management, In: *Proc. 13th Int. Conf. on Machine Learning* (1996) 139-147.
12. Fawcett, T., Provost, F.: Adaptive fraud detection, *Data Mining and Knowledge Discovery* **1** (1996) 291-316.
13. Ferri, F.J., Sánchez, J.S., Pla, F.: Editing prototypes in the finite sample size case using alternative neighbourhoods, In: *Advances in Pattern Recognition* **LNCS 1451**, Springer-Verlag (1998) 620-629.
14. Gordon, D.F., Perlis, D.: Explicitly biased generalization, *Computational Intelligence* **5** (1989) 67-81.
15. Hart, P.E.: The condensed nearest neighbor rule, *IEEE Trans. on Information Theory* **14** (1968) 515-516.
16. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection, In: *Proc. 14th Int. Conf. on Machine Learning* (1997) 179-186.
17. Kubat, M., Holte, R., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images, *Machine Learning* **30** (1998) 195-215.
18. Lewis, D, Catlett, J.: Heterogeneous uncertainity sampling for supervised learning, In: *Proc. 11th Int. Conf. on Machine Learning* (1994) 148-156.
19. Ling, C.X., Li, C.: Data mining for direct marketing: problems and solutions, In: *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining* (1998) 73-79.
20. Mladenic, D., Grobelnik, M.: Feature selection for unbalanced class distribution and naive Bayes, In: *Proc. 16th Int. Conf. on Machine Learning* (1999) 258-267.
21. Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., Brunk, C.: Reducing misclassification costs, In: *Proc. 11th Int. Conf. on Machine Learning* (1994) 217-225.
22. Swets, J., Dawes, R., Monahan, J.: Better decisions through science, *Scientific American* (2000) 82-87.
23. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data sets, *IEEE Trans. on Systems, Man and Cybernetics* **2** (1972) 408-421.
24. Woods, K., Doss, C., Bowyer, K.W., Solka, J., Priebe, C., Kegelmeyer, W.P.: Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography, *International Journal of Pattern Recognition and Artificial Intelligence* **7** (1993) 1417-1436.