Imbalanced Training Set Reduction and Feature Selection Through Genetic Optimization

R. Barandela[a], J. K. Hernández[a], J. S. Sánchez[b] , F. J. Ferri[c]
[a]*Instituto Tecnológico de Toluca, Av. Tecnológico s/n, 52140 Metepec, México*
[b]*Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I, 12071 Castelló, Spain*
[c]*Dept. d'Informàtica, Universitat de València, 46100 Burjassot (València), Spain*

Abstract. Despite its simplicity and good classification performance, the Nearest Neighbor (NN) rule is not applied in many practical tasks because of the high amount of computational resources that it requires. Besides, when working with imbalanced training samples, its classification accuracy can be seriously degraded. In the present paper we propose two genetic algorithms to cope with these two issues. The purpose is to obtain complexity reduction while at the same time, to get a better balance in the training sample. Experimental results showing the benefits of our proposals are also reported.

Keywords: Imbalanced sample; Prototype and Feature Selection; Genetic algorithms

1. Introduction

The Nearest Neighbor (NN) classification rule is one of the most important and well-known non-parametric classification methods. As with any supervised pattern recognition method, the design of the NN rule is based on a training sample (TS): a collection of examples, instances or training patterns, previously analyzed and identified by a human expert. When working with the NN rule, the entire TS is stored in the computer memory. Then, to classify a new pattern, its distance to each one of the stored training patterns is computed. The new pattern is then assigned to the class represented by its nearest neighboring training pattern.

Despite its simplicity and effectiveness, practical use of the NN rule has been limited due to several drawbacks. Firstly, it is slow in the classification phase and demands high storage requirements. On the other hand, it shows high sensitivity to outlier patterns. Besides, theoretical analyses suggest potential problems for the NN rule when working with large dimensionalities and not correspondingly large number of training patterns [1].

Performance of the NN rule can be also affected when the class distribution of the data in the TS is *imbalanced*. A TS is said to be imbalanced when one of the classes

Work partially supported by grants SEP-2003-C02-44225 from the Mexican CONACYT, and TIC2003-08496 from the Spanish CICYT.
Correspondence to: Francesc J. Ferri (ferri@uv.es) or J. Salvador Sánchez (sanchez@uji.es).

(the minority one) is heavily under-represented in comparison to the other (the majority) class. This issue is particularly important in those applications where it is costly to misclassify minority-class examples. For simplicity, and consistently with the common practice [2], only two-class problems are here considered. High imbalance occurs in real-world domains where the decision system is aimed to detect a rare but important case, such as fraudulent telephone calls [3], oil spills in satellite images of the sea surface [4], an infrequent disease [5], or text categorization [6]. Several of the proposals to handle the imbalance issue are aimed at downsizing (in some way) the majority class, looking for a match in size with the minority one [2, 7]. It has been acknowledged that the overall accuracy is not the best criterion to assess the classifier's performance such imbalanced domains. Consequently, other criteria have been proposed as the geometric mean, $g=(a^+ \cdot a)^{1/2}$, where a^+ is the accuracy on cases from the minority class and a^- is the accuracy on cases from the majority one. This measure constitutes a convenient and simple tradeoff which tries to maximize the accuracy on each of the two classes while keeping these accuracies balanced.

The present paper focuses on complexity reduction through removal of instances in the majority class and elimination of features in both classes. By doing that, we are downsizing the majority class for also dealing with the imbalance problem, in an effective and efficient manner, as will be shown in the experimental results below reported. To this end, we have employed a genetic algorithm based on the idea of Kuncheva and Jain [8], conveniently adapted for the imbalanced training sample problem. We have also explored the convenience of allowing the genetic algorithm to eliminate some (supposedly noisy) examples in the minority class, to enhance the performance of the resulting classifier.

2. Complexity reduction

Since the whole TS has to be stored and used in the classification process, the main problem using the NN rule is the significant time and memory resources that it requires. In order to lessen this computational burden, it is possible to employ suitable data structures and fast search algorithms. In the context of the present paper, we focus on a different line: the removal of some patterns. The latter has also the advantage of leading to both time and memory requirements attenuation. Within this approach, the goal is to design a good TS with a smaller cardinality that will ideally allow for the lowest possible error rate of the classifier.

Prototype selection techniques that find subsets S guaranteeing zero errors when used to classify the original TS are called *condensing* techniques, and the produced subset is said to be *consistent* with the TS. This idea was first proposed by Hart [9]. Many different algorithms for prototype selection using different approaches and contexts have been proposed [11].

Features used to describe instances are not necessarily all relevant and beneficial for classification accuracy. Additionally, a high number of features may slow down the training and/or the classification phase, while giving similar classification results as obtained with a smaller feature subset. Performance of the NN rule can be severely damaged in a high-dimensional input feature space with finite size TS [12]. Feature subset selection is commonly used in many classification tasks. According to John et al. [13], there are two main approaches to feature subset selection: *filtering* and

wrapper. In the filtering approach a feature subset is selected independently of the learning model that will use the selected features. The idea of the wrapper approach is to select a feature subset using the evaluation function based on the same algorithm that will be used for learning. A comparison of several techniques for feature selection is presented in [14]. Genetic algorithms have often been employed for feature selection (e.g., [15]). The convenience of the simultaneous selection of features and prototypes has been studied in [8, 16, 17].

3. Class imbalance in the training sample

As already pointed out, the performance of the NN rule can be also affected when the class distribution of the data in the TS results imbalanced. Most of the attempts for dealing with the class imbalance problem can be sorted into three categories [18]:

a) Over-sampling the minority class to match the size of the other class [19].
b) Downsizing the majority class so as to match the size of the other class [2].
c) Internally biasing the discrimination based process so as to compensate for the class imbalance [5, 20].

The two re-sampling approaches (by over-sampling the minority class or by under-sampling the majority one) cause the class distribution to become more balanced. Nevertheless, both strategies have shown important drawbacks. Under-sampling may throw out potentially useful data, while over-sampling increases the TS size and hence the computational requirements. In the last years, research has focused on improving these basic methods. Kubat and Matwin [2] proposed an under-sampling technique that is aimed at removing those majority patterns that are "redundant" or that "border" the minority instances. They assume that these bordering cases are noisy examples. Their procedure begins with all the examples of the minority class and adds as many examples of the majority class as necessary to complete a consistent subset. Then, all instances of the majority class in that reduced subset that participate in the so-called Tomek's pairs [10] are removed from this reduced subset.

Chawla et al. [19] proposed a technique for over-sampling the minority class. Instead of merely replicating instances of the minority class, they form new minority "synthetic" patterns. Barandela et al. [21] explore the convenience of designing a multiple classification system for working in imbalanced situations. Instead of using a single classifier, an ensemble is implemented. The idea is to train each one of the individual components of the ensemble with a balanced TS. In order to achieve this, each individual component of the ensemble is trained with a subset of the TS. As many subsets of the TS as required to get balanced subsets are generated.

Pazzani et al. [22] take a slightly different approach when learning from an imbalanced TS by assigning different weights to instances of the different classes. On the other hand, Ezawa et al. [20] bias the classifier in favor of certain feature relationships. Kubat et al. [4] use some counter-examples to bias the recognition process. Some authors propose to select or to weight features for tackling the imbalance issue [6, 23].

4. A new constrained genetic optimization approach

Genetic Algorithms (GA) are kinds of search methods based on an analogy to the natural process of evolution. These algorithms start with a population of individuals (chromosomes) and seek to alter and eventually optimize their composition for the solution of a particular problem. Individuals (candidate solutions to a problem) evolve over a series of generations. A genetic algorithm is, basically, a random process guided by a fitness function. In each iteration of the algorithm, a fixed number of chromosomes is generated through the application of several "genetic" operators: selection, crossover and mutation. Selection is based on the principle of survival of the fittest, in which the individuals that are best suited for the environment are the ones that survive. Fitness values are calculated for all chromosomes in the population. Crossover occurs when two individuals chosen randomly from the population are joined or "mated" such that the resulting offspring contain partial replications of the information contained in each of the parents. In nature, evolutionary progress often occurs through sudden mutations in the genetic constitution of individuals. This mutation cause jumps in the rate of evolutionary change. In GA, chromosomes can be mutated by changing one or a few components of a few individuals. A particular GA is identified by a method of coding the solutions, a form of the genetic operators, and a definition of the fitness function.

GAs have been shown to be effective for exploring NP-hard search spaces as efficient optimizers relative to computer-intensive exhaustive search (e.g., [24]). Editing the TS with GAs has been discussed by Chang and Lippman [25].

Kuncheva and Jain [8] proposed a genetic algorithm aimed at simultaneously editing the TS (by removing some instances) and selecting appropriate features, when working with a NN classifier. If the TS embraces N instances and n features, the algorithm looks for a subset S_1 of the patterns and a subset S_2 of the features. In their codification, each chromosome C is a binary string consisting of $N + n$ bits and representing these two subsets. The ith bit has value 1 when the respective element of the original instances (original features) is included in S_1 (S_2), and 0 otherwise. As the fitness function they use:

$$F(S) = P(S) - \alpha \, (\text{card}\,(S_1) + \text{card}\,(S_2))/\,(N + n) \qquad (1)$$

where $P(S)$ is the estimation of the classification accuracy to be obtained when replacing the TS with the subset S (S contains all the original instances and features, whose respective bits in C have values 1), and α is a positive coefficient. Other parameters of their algorithm are:

- population size = 10;
- initialization probability = 0.8 (the number of 1's in the initial population will be around 80% of all the values generated);
- terminal number of generations = 100;
- 5 couples are selected at random (repetitions are permitted) to produce 10 offspring chromosomes (probability of crossover = 1.0);
- probability of mutation = 0.1;
- selection strategy: elitist, i.e., the current population and the offspring are pooled, and the "fittest" survive as the next population.

Since our interest is not only to reduce the computational demands of the NN rule, but also to tackle the imbalanced TS problem, our proposal involves two different modifications of the algorithm of Kuncheva and Jain.

4.1 Direct one-sided selection

In the first one (GA1), the genetic operators are allowed to work only with the bits corresponding to the patterns of the majority class (and with the bits of all the features). In fact, a chromosome C in GA1 is a string of $N_1 + n$ bits, where N_1 is the size of the majority class in the TS and n is, as before, the number of features in TS. That means that the examples of the minority class remain unaltered, except for the elimination of some of the features. Attempting to get a better balance between the two classes, for the bits corresponding to the instances of the majority class the initialization probability is set as the proportion: minority-size/majority-size.

In accordance with the above statement in Section 1, concerning the proper criterion to measure classification performance in imbalanced situations, we have substituted the overall accuracy term, P(S), in the above formula for the fitness function, by the geometric mean value that would result of using the subset S as the new TS: g(S). In our algorithm, S is formed by all the original instances of the minority class and all the original examples of the majority class and the features whose respective bits in C have values 1. Besides, we have used the roulette strategy for selection: survivors are randomly chosen among all the chromosomes in the population, each chromosome with a survival probability proportional to its fitness value. All the other parameters in GA1 are the same as in the algorithm of Kuncheva and Jain.

4.2 Imbalanced two-sided selection

On the other hand, there is no reason to consider that the minority class –in an imbalanced TS- is completely free from atypical cases. These noisy instances certainly must affect the classifier's performance. However, none of the previously published works concerned with the imbalanced problem has reported attempts to eliminate minority noisy examples. Because of the relative small size of the minority class, its examples are considered as very important and elimination of some of them is usually regarded as a very risky undertaking. To explore the convenience of doing also some (small) editing in the minority class, we have designed another idea for a genetic algorithm.

In the second proposal (GA2), each chromosome, as in Kuncheva and Jain, consists of $N + n$ bits. For the bits corresponding to the patterns of the minority class the initialization probability is set as 0.85. In that way, we are looking for a reduction (through elimination of some noisy examples) in the minority class size of about 15%. It is a rather conservative rate, since in several experiments with Wilson's editing algorithm reductions of 20-25% have been observed (e.g., [26]). Then, the initialization probability for the bits associated to the training patterns in the majority class is set in such a way as to balance the sizes of both classes in the resulting subset. That is, this initialization probability is set as the proportion (0.85*minority size)/majority size. All other aspects in GA2 are exactly the same as in GA1.

5. Experiments and results

Our two proposals have been assessed with experiments that were carried out with four real datasets that have been taken from the UCI Machine Learning Database Repository (http://www. ics.uci.edu/~mlearn/). Five-fold cross-validation was employed to obtain averaged results of the *g* criterion. To facilitate comparison with other published results [2], in the Glass dataset the problem was transformed for discriminate class 7 against all the other classes and in the Vehicle dataset the task is to classify class 1 against all the others. Satimage dataset was also mapped to configure a two-class problem: the training patterns of classes 1, 2, 3, 5 and 6 were joined to form a unique class and the original class 4 was left as the minority one. Phoneme is a two-class dataset.

Table 1. Averaged *g* values obtained in the experiments

	Glass	*Phoneme*	*Satimage*	*Vehicle*
Original	86.7	73.8	70.9	55.8
Kuncheva & Jain	91.1	73.7	70.0	55.6
GA1	91.6	76.2	76.8	67.8
GA2	97.7	75.4	73.6	71.6
Kubat & Matwin	79.0	68.3	72.9	65.5

The experimental results are shown in Table 1. Averaged geometric mean values obtained when classifying with the original (not processed) TS, and with this TS after being processed as in [2] and with the algorithm in [8], are also included for comparison purposes. It can be seen that the two proposed algorithms outperform all the other methods on all datasets.

Table 2. Majority/minority ratio

	Glass	*Phoneme*	*Satimage*	*Vehicle*
Original	6.25	2.41	9.28	2.99
Kuncheva & Jain	6.03	2.41	9.15	2.98
GA1	1.39	1.06	1.22	1.94
GA2	1.42	1.06	1.23	1.24
Kubat & Matwin	3.00	6.02	1.10	1.36

The effects of the GA1 and GA2 methods can be better analyzed when considering the resulting balance between both classes after their application. Figures in Table 2 permit this analysis and a comparison with the balance yielded by the other techniques. The two new genetics-based algorithms reach an almost perfect balance in the resulting TSs. This is particularly remarkable in the case of GA2, despite the fact that this algorithm reduces also the size of the minority class.

Table 3. Size-minority-class/dimensionality ratio

	Glass	*Phoneme*	*Satimage*	*Vehicle*
Original	2.7	253.6	13.9	9.4
Kuncheva & Jain	3.5	227.7	13.7	9.4
GA1	6.3	264.2	56.8	18.1
GA2	8.2	216.3	55.7	19.2
Kubat & Matwin	0.9	42.1	12.6	6.9

Another issue of importance for a better classification performance is the requirement for the size/dimensionality ratio of the TS to be high enough. According to some authors, for each feature included in the TS there must be 5-10 training patterns. Moreover, this ratio is to be measured by considering the size of the less represented class in the TS (in our case, the minority class). Any procedure aimed at downsizing the majority class can not be very successful in handling the imbalance situations in cases such as that of the Glass dataset, if reduction in the number of features is not, at the same time, accomplished. That is the reason explaining the bad result produced by the procedure of Kubat and Matwin in the mentioned dataset (see Table 1). Genetic algorithms like that of Kuncheva and Jain and our GA1 and GA2 are more suitable for those situations, as can be seen in Table 3.

When downsizing the majority class, it is possible, in general, to obtain a better balance in the TS and, consequently, to increase the value of the attained geometric mean. But, at the same time, some reduction in the overall accuracy (as it is traditionally measured) is produced. This situation is reported in some papers (e.g., [2]). One of the findings of our experiments is that, by using the genetics-based approaches, the g mean value can be increased while keeping the overall accuracy in levels comparable to that obtained when working with the original (not processed) TS. The details can be examined in Table 4.

Table 4. Overall accuracy

	Glass	*Phoneme*	*Satimage*	*Vehicle*
Original	96.0	76.1	88.1	71.6
Kuncheva & Jain	97.0	76.2	87.6	71.6
GA1	94.5	74.7	80.4	75.3
GA2	96.0	74.8	78.1	73.0
Kubat & Matwin	75.5	63.3	81.2	66.9

6. Concluding remarks

The high computational demand of the NN rule can become a serious drawback in practical applications of this well known classifier. On the other hand, in many real-world domains, supervised pattern recognition methods have to cope with highly imbalanced TSs. In this context, two new genetic algorithms have been proposed in the present paper. The idea of designing a genetic algorithm for downsizing the majority class has accomplished promising results. This algorithm is guided by a fitness function that includes the geometric mean associated to each resulting subset (instead of the overall accuracy), allowing for a proper balance in the TS. Moreover, this algorithm permits also a reduction in the number of features to be considered yielding a more suitable size/dimensionality ratio, even in those cases where the minority class is too small. The results of our genetic algorithms have excelled, in all the experimental datasets, those obtained by the proposal of Kubat and Matwin.

The idea of cleaning also the minority class, through removal of some noisy instances, deserves further attention. At the present, we are doing some research about the proper initialization probability for the bits of the minority class in the implementation of GA2, to improve its performance.

References

1. Hand D, Mannila H, Smyth P, *Principles of Data Mining*, MIT Press, Cambridge, 2001.
2. Kubat M, Matwin S, Addressing the curse of imbalanced training sets: one-sided selection, In: Proc. 14th International Conference on Machine Learning (1997) 179-186.
3. Fawcett T, Provost F, Adaptive fraud detection, *Data Mining & Knowledge Discovery* **1** (1996) 291-316.
4. Kubat M, Holte R, Matwin S, Detection of oil-spills in radar images of sea surface, *Machine Learning* **30** (1998) 195-215.
5. Woods K, Doss C, Bowyer KW, Solka J, Priebe C, Kegelmeyer WP, Comparative evaluation of pattern recognition techniques for detection of micro-calcification in mammography, *International Journal of Pattern Recognition and Artificial Intelligence* **7** (1993) 1417-1436.
6. Mladenic D, Grobelnik M, Feature selection for unbalanced class distribution and naive Bayes, In: Proc. 16th International Conference on Machine Learning (1999) 258-267.
7. Barandela R, Sánchez JS, García V, Rangel E, Strategies for learning in class imbalance problems. *Pattern Recognition* **36** (2003) 849-851.
8. Kuncheva LI, Jain LC, Nearest neighbor classifier: simultaneous editing and feature selection, *Pattern Recognition Letters* **20** (1999) 1149-1156.
9. Hart PE, The condensed nearest neighbor rule, *IEEE Trans. Information Theory* **14** (1968) 515-516.
10. Tomek I, Two modifications of CNN, *IEEE Trans. Systems, Man and Cybernetics* **7** (1976) 769-772.
11. Wilson DR, Martinez TR, Reduction techniques for instance-based learning algorithms, *Machine Learning* **38** (2000) 257-286.
12. Domeniconi C, Peng J, Gunopulos D, Locally adaptive metric nearest-neighbor classification, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24** (2002) 1281-1285.
13. John GH, Kohavi R, Pfleger K, Irrelevant features and the subset selection problem, In: Proc. 11th International Conference on Machine Learning (1994) 121-129.
14. Ferri FJ, Pudil P, Hatel M, Kittler J, Comparative study of techniques for large-scale feature selection, In: Proc. Pattern Recognition in Practice IV (1994) 403-413.
15. Siedlecki W, Sklansky J, A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters* **10** (1989) 335-347.
16. Dasarathy BV, Sánchez JS, Concurrent feature and prototype selection in the nearest neighbor based decision process, In: Proc. 4th World Multi-conference on Systemics, Cybernetics and Informatics, VII (2000) 628-633.
17. Ho SY, Liu CC, Liu S, Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm, *Pattern Recognition Letters* **23** (2002) 1495-1503.
18. Eavis T, Japkowicz N, A recognition-based alternative to discrimination-based multi-layer perceptrons, Workshop on Learning from Imbalanced Data Sets, Technical Report WS-00-05, 2000.
19. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP, SMOTE: synthetic minority over-sampling techniques, *Journal of Artificial Intelligence Research* **16** (2000) 321-357.
20. Ezawa KJ, Singh M, Norton SW, Learning goal oriented Bayesian networks for telecommunications management, In: Proc. 13th International Conference on Machine Learning (1996) 139-147.
21. Barandela R, Sánchez JS, Valdovinos RS, New applications of ensembles of classifiers, *Pattern Analysis and Applications* **6** (2003) 245-256.
22. Pazzani M, Merz C, Murphy P, Ali K, Hume T, Brunk C, Reducing misclassification costs, In: Proc. 11th International Conference on Machine Learning (1994) 217-225.
23. Cardie C, Howe N, Improving minority class prediction using case-specific feature weights, In: Proc. 14th International Conference on Machine Learning (1997) 57-65.
24. Goldberg D, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
25. Chang EI, Lippmann RP, Using genetic algorithms to improve pattern classification performance, In: Advances in Neural Information Processing Systems 3, Morgan Kaufmann, San Mateo, CA (1991) 797-803.
26. Sánchez JS, Barandela R, Ferri FJ (2002) On filtering the training prototypes for nearest neighbor classification, In: Lecture Notes in Artificial Intelligence 2504 (2002) 239-248.