Use of Ensemble Based on GA for Imbalance Problem

Laura Cleofas[1], Rosa Maria Valdovinos[1], Vicente García[2], and Roberto Alejo[3]

[1] Centro Universitario UAEM Valle de Chalco,
56615 Valle de Chalco, México
{laura18cs,li_rmvr}@hotmail.com
[2] Universitat Jaume I, 12071 Castelló de la Plana, Spain
vgarciaj@hotmail.com
[3] Centro Universitario UAEM Atlacomulco,
50450 Atlacomulco, México
ralejoe@uaemex.mx

Abstract. In real-world applications, it has been observed that class imbalance (significant differences in class prior probabilities) may produce an important deterioration of the classifier performance, in particular with patterns belonging to the less represented classes. One method to tackle this problem consists to resample the original training set, either by over-sampling the minority class and/or under-sampling the majority class. In this paper, we propose two ensemble models (using a modular neural network and the nearest neighbor rule) trained on datasets under-sampled with genetic algorithms. Experiments with real datasets demonstrate the effectiveness of the methodology here proposed.

Keywords: Genetic Algorithm, Imbalance, Nearest Neighbor Rule, Modular Neural Network.

1 Introduction

The class imbalance problem has received considerable attention in areas such as Machine Learning and Pattern Recognition. A two-class dataset is said to be imbalanced when one of the classes (the minority one) is heavily under-represented in comparison to the other class (the majority one) [1]. This issue is particularly important in real-world applications where it is costly to misclassify examples from the minority class, such as the diagnosis of rare diseases [2], the detection of fraudulent telephone calls [3], text categorization [4] and credit assessment [5], between others. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases, respectively, they are also known as positive and negative examples.

Basically, the research on this topic can be categorized into three groups:

1. Solutions methods for handling the imbalance problem in two levels: the data level [6], or the algorithmic level [7].
2. Measuring the classifier performance in imbalanced domains [8,9].
3. Analyzing the relationship between class imbalance and other data complexity characteristics [10,11].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 547–554, 2009.

Focusing on the first one, which is the most investigated, the data level methods, balancing the original data set by resampling the data space until the classes are approximately equally represented. On the other hand, the algorithmic level methods, try to adapt existing learning algorithms to deal the imbalance problem, while keeping the original training data sets unchanged.

Nowadays, the best strategy for handling this problem is not defined, however, several studies suggest to combine two or more strategies of the same level as the best option [6,12]. For example, Barandela et al. [6] propose to use SMOTE [13] for over-sampling the minority class, and after that, applying Wilson Editing remove patterns which belong to the majority class.

In this paper, we propose a methodology for handling the imbalance problem using a solution method, which consider two treatment level, the data and algorithmic level. Thus, in a first step (the data level), the original training set is under-sampled by a Genetic Algorithms (GA). Next, (the algorithm level) an ensemble is trained with the solutions given by the GA.

In this way, using a GA we obtain subsamples whose chromosome considers four aspects: size reduction, diversity, good fitness and balance. After that, the GA method finds the best subsamples for train the ensemble.

On the other hand, an ensemble is a set of individual classifiers whose decisions are combined when classifying new patterns [14]. In general, an ensemble is built in two steps, that is, training multiple individual classifiers and then combining their predictions. According to the styles of training the base classifiers, current ensemble algorithms can be roughly categorized into two groups, that is, algorithms where base classifiers must be trained sequentially, and algorithms where base classifiers could be trained in parallel. In this work, we employ two parallel ensembles using a mixture of experts (modular neural network) [15] and the 1-nearest neighbour rule as learning algorithm.

From now, on the rest paper is organized as follows: Section 2 exposes the GA method. Section 3 describes the ME used in this paper. Next the experimental results are discussed in Section 4. Finally, Section 5 gives the main conclusions and points out possible directions for future research.

2 Genetic Algorithms

The most basic structure of the GA proposed by Holland [16], begins with a set of possible solutions (population) codified as a chain of bits (called chromosome), later with the use of a method to evaluate the behavior (fitness) of each chromosome, the parents of the next population are determined.

In this work we modify the GA proposed by [17]. Diaz et al., to reduce the processing time of the GA, in addition to the 0′s, some chromosomes are reduced in 20%, that is to say, during the evolutive process, several genes marked with a different value of 0 or 1 were ignored. The leaving-one-out method was used as fitness method and, an elitist method select the best solutions in each step and uses these chromosomes to apply the genetic operators: crossover and mutation. The former, consists of the uniform crossover and, next, randomly change 10% of the genes in each chromosome.

Here, this algorithm was modified using a threshold h in order to identify the minority classes and for obtain a balanced chromosome. The threshold is obtained according to the following function:

$$h = \frac{t}{c} \tag{1}$$

were t is the number of training samples and c is the number of classes in the problem.

In the GA process, after to obtain de first population, if the number of patterns in any class is higher than h, the genes corresponds to that class which is adjusted to h. With this, we obtain balanced chromosomes, in other words, balanced subsamples. It is right to suppose that in the complete GA process the balance caught change, but on some way, we guaranteed a similar distribution between the classes.

When the evolutionary process was finished, the best five solutions of the all epochs are used for building the ensemble.

3 Mixture of Experts

A Mixture of Experts (ME) or modular network solves a complex computational task by dividing it into a number of simpler subtasks and then combining their individual solutions. Thus, a ME consists of several expert neural networks (modules), where each expert is optimized to perform a particular task of an overall complex operation. An integrating unit, called gating network, is used to select or combine the outputs of the modules (expert networks) in order to form the final output of the modular network. In the more basic implementation of these networks, all the modules are of a same type [18,19], but different schemes could be also used.

There exist several implementations of the modular neural network, although the most important difference among them refers to the nature of the gating network. In some cases, this corresponds to a single neuron evaluating the performance of the other expert modules [21]. Other realizations of the gating network are based on a neural network trained with a data set different from the one used for training the expert networks [22]. In this work, all the modules (the experts and the gating network) will be trained with a unique data set [15,22] (see Fig. 1).

All modules, including the gating network, have n input units, that is, the number of features. The number of output units in the expert networks is equal to the number of classes c, whereas that in the gating network is equal to the number of experts, say r. The learning process is based on the stochastic gradient algorithm, where the objective function is defined as:

$$-\ln\left(\sum_{i=1}^{r} g_i * \exp\left(-\frac{1}{2}\|s - Z_i\|^2\right)\right) \tag{2}$$

where s is the output desired for input x, $z_j = xw_j$ is the output vector of the j'th expert network, g_i is the normalized output of the gating network, u_i is the total weighted input received by output unit j of the gating network, and g_j can be viewed as the probability of selecting expert j for a particular case.

$$g_i = \frac{\exp(u_i)}{\sum_{j=1}^{r} \exp(u_j)} \quad (3)$$

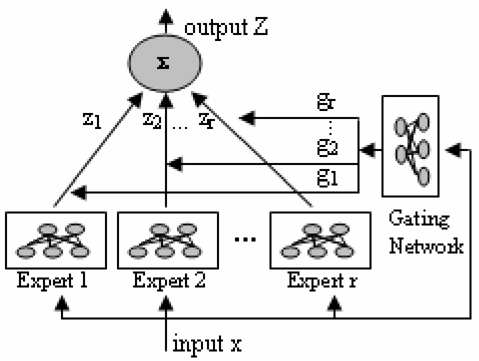

Fig. 1. Graphical representation of the ME architecture. Each module (including the gating network) is a feedforward network and receives the same input vector. The final output of the whole system is the sum of $z_j g_j$.

4 The Nearest Neighbor Rule

The Nearest Neighbor (NN) rule [21] is one of the most celebrated algorithms in machine learning. In recent years, interest in these methods has flourished again in several science fields, due to their conceptual simplicity and to an asymptotic error rate conveniently bounded in terms of the optimal Bayes error, they are revealed as powerful nonparametric classification systems in real-world problems.

In its classical manifestation, given a set of n previously labelled prototypes or training sample (TS), this classifier assigns a given sample to the class indicated by the label of the closest prototype in the TS.

5 Experimental Results

This section expose the experimental results obtained with two ensemble models: using mixture of experts and using the NN rule, both of them trained on undersampled subsamples by a GA. The section was dividing in three parts. The first one, describe the method used for transform the datasets in a problem with two classes. The second part exposes the evaluation criterion for the imbalance problem here used. Finally, the experimental results are shown in the third part.

5.1 Datasets

The results here reported correspond to the experiments over seven real datasets taken from the UCI Machine Learning Database Repository [21]. For each data set, the

5-fold cross-validation error estimate method was employed: 80% of the available patterns were for training purposes and 20% for the test set.

Some datasets were transformed in a problem of two classes. In the Glass dataset the problem was transformed for discriminate class 7 against all the other classes and in the Vehicle dataset the task was 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, Cancer and German are a two-class datasets. Table 1 presents the positive and negative samples in the datasets.

Table 1. Description of the data sets

Dataset	Positive samples	Negative samples	Majority class
Cancer	191	355	1
Pima	268	500	1
Glass	17	197	1,2,3,4,5,6,8,9
German	300	700	1
Phoneme	1586	3818	1
Vehicle	212	634	2,3,4
Satimage	626	5809	1,2,3,5,6

5.2 Performance Evaluation in Class Imbalance Problem

To evaluate the performance of learning systems, a confusion matrix like that in Table 2 (for a two-class problem) is usually employed. The elements in this table characterise the classification behaviour of the given system.

Table 2. Description of the data sets

	Predictive positive	Predictive negative
Positive class	True positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

From this, four simple measures can be directly obtained: TP and TN denote the number of positive and negative cases correctly classified, while FP and FN refer to the number of misclassified positive and negative examples, respectively.

The most widely used metrics for measuring the performance of learning systems are the error rate and the accuracy, which can be computed as:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

Nevertheless, as pointed out by many authors, overall accuracy is not the best criterion to assess the classifier's performance in imbalanced domains. For instance, consider a domain where only 5% of the patterns belong to the minority class. In such a situation, labeling all new patterns as members of the majority class would give an accuracy of 95%. Obviously, this kind of system would be useless. Consequently, other criterion has been proposed. One of the most widely accepted criterion is the geometric mean:

$$g = \sqrt{a^{+} \cdot a^{-}} \tag{5}$$

where a+ is the accuracy on cases from the minority class:

$$a^{+} = \frac{TP}{TP + FN} \tag{6}$$

and a- is the accuracy on cases from the majority one [9].

$$a^{-} = \frac{TN}{TN + FP} \tag{7}$$

This measure tries to maximize the accuracy on each of the two classes while keeping these accuracies balanced.

5.3 Results

The ensemble consists of five members trained on subsamples obtained with two variants of GA: with patterns reduction (GA1) and without pattern reduction (GA2). The experimental results given in Table 3 correspond to the averages of the geometric mean; values in parenthesis indicate the standard deviation. This table has three parts. In the first one, the results when employing the original TS, both with ME and 1-NN classifiers are included for comparison purposes. In the second and the third part, we present the geometric mean values observed when the ME and 1-NN were trained on subsamples under-sampled through GA1 and GA2.

Table 3. Geometric mean values

Dataset	Original TS		ME		1-NN	
	ME	1-NN	GA1	GA2	GA1	GA2
Cancer	86.4(6.8)	94.0(4.1)	87.7(5.8)	85.3(6.9)	95.8(3.3)	96.5(2.4)
Pima	50.4(13.8)	58.4(8.1)	53.9(4.8)	58.5(2.4)	65.1(4.9)	64.9(5.2)
Glass	85.8(9.9)	86.7(12.2)	81.5(6.3)	86.2(9.5)	84.6(16.3)	84.9(16.0)
German	56.3(27.6)	49.8(8.0)	59.3(9.9)	59.8(2.0)	54.1(5.3)	55.8(5.2)
Phoneme	73.9(6.8)	73.8(6.0)	74.8(7.0)	73.6(5.6)	74.1(8.3)	73.6(5.6)
Vehicle	58.2(5.6)	55.8(7.2)	61.2(5.9)	58.0(7.9)	55.5(4.5)	59.4(12.2)
Satimage	69.8(10.6)	70.9(15.1)	71.0(16.4)	65.3(10.5)	68.6(18.4)	66.4(19.4)

From results reported in this section, some preliminary conclusions can be drawn. First, except for Glass dataset, for all data sets there are at least one classifier ensemble whose classification g is higher than the obtained when using the original TS. Second, comparing the two learning algorithms, in general the ME outperforms (five datasets) the 1-NN rule, independent on the GA strategy adopted.

Finally, with respect to differences in g value between the GA1 and GA2, it has to be especially remarked the fact that results of the GA2 strategy are inferior to those of the GA1 approach. As can be seen, although differences are not significant, the GA2 does not seem to present any advantage with respect to the GA1. That can be because in the GA2 approach the subsample obtained caught lost training samples which provide useful information for the classifier algorithm.

6 Concluding Remarks

In many real-world applications, supervised pattern recognition methods have to cope with imbalanced TSs. In the present paper we propose a new methodology focused on the solution methods approach, which combines an under-sampling method using a GA and an ensemble trained with the solutions given by the GA.

The experiments on seven real-problem datasets have been through as a way of demonstrating the behavior and competitively of this methodology. From the experiments carried out, it seems that in general, the ME provide better levels of geometric mean than the NN rule. On the other hand, we also show that the method for reducing the computational cost of the GA (GA2) when some genes are ignored, does not favour substantially the precision of the ensemble.

Future works, pointing to validate the proposal using another neural network model and with ensembles based on resampling methods which including weighting measures in the combining decision schema are in line. More comparisons on others problems form the UCI repository as treatment of the dimensionality and the noisy patterns contained in the database will be developed as soon as possible.

Acknowledgments. This work has been partially supported by grants: GV/2007/105 from the Generlitat Valenciana (Spain), DPI2006-15542-C04-03 from the Spanish CICYT, from the CSD2007-00018 Spanish Ministry of Education and Science, CB-2008-01-107085 from the Mexican CONACyT and PROMEP/103.5/08/3016 from the Mexican SEP.

References

1. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for Learning in Class Imbalance Problems. Pattern Recognition 36, 849–851 (2003)
2. Woods, K., Doss, C., Bowyer, K.W., Solk, 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, 1417–1436 (1993)
3. Fawcett, T., Provost, F.: Adaptive Fraud Detection. Data Mining and Knowledge Discovery 1, 291–316 (1996)
4. Tan, S.: Neighbor-weighted K-Nearest Neighbour for Unbalanced Text Corpus. Expert Systems with Applications 28, 667–671 (2005)
5. Huang, Y., Hung, C., Jiau, H.C.: Evaluation of Neural Networks and Data Mining Methods on a Credit Assessment Task for Class Imbalance Problem. Nonlinear Analysis: Real World Applications 7, 720–747 (2006)
6. Barandela, R., Valdovinos, R.M., Sánchez, J.S., Ferri, F.J.: The Imbalanced Training Sample Problem: Under or Over Sampling? In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 806–814. Springer, Heidelberg (2004)

7. Ezawa, K.J., Singh, M., Norton, S.W.: Learning Goal Oriented Bayesian Networks for Telecommunication Risk Management. In: Proceedings of the 13th International Conference on Machine Learning, pp. 139–147 (1996)
8. Ranawana, R., Palade, V.: Optimized Precision – A New Measure for Classifier Performance Evaluation. In: Proceedings IEEE Congress on Evolutionary Computation, pp. 2254–2261 (2004)
9. Daskalaki, S., Kopanas, I., Avouris, N.: Evaluation of Classifiers for an Uneven Class Distributions Problem. Applied artificial intelligence 20, 381–417 (2006)
10. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Learning with class skews and small disjuncts. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS, vol. 3171, pp. 296–306. Springer, Heidelberg (2004)
11. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Class Imbalance Versus Class Overlapping: An Analysis of a Learning System Behavior. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS, vol. 2972, pp. 312–321. Springer, Heidelberg (2004)
12. Batista, G.E., Pratti, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explorations 6, 20–29 (2004)
13. Chawla, N.V., Bowyer, K.W., Hall, L., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
14. Dietterich, T.G.: Machine Learning Research: Four Current Directions. AI Mag. 68, 97–136 (1997)
15. Jacobs, R., Jordan, M., Hinton, G.: Adaptive Mixture of Local Experts. Neural Computation 3(1), 79–87 (1991)
16. Holland, J.: Adaptation in Natural and Artificial System. The University of Michigan Press (1975)
17. Diaz, R.I., Valdovinos, R.M., Pacheco, J.H.: Comparative Study of Genetic Algorithms and Resampling Methods for Ensemble Constructing. In: Proceedings of IEEE Congress on Evolutionary Computation, Hong Kong, China, pp. 4180–4184 (2008)
18. Bauckhage, C., Thurau, C.: Towards a Fair'n Square Aimbot - Using Mixture of Experts to Learn Context Aware Weapon Handling. In: Proceedings of GAME-ON, Ghent, Belgium, pp. 20–24 (2004)
19. Hartono, P., Hashimoto, S.: Ensemble of Linear Perceptrons with Confidence Level Output. In: Proceedings of the 4th Intl. Conf. on Hybrid Intelligent Systems, Kitakyushu, Japan, pp. 186–191 (2004)
20. Zaman, R., Wunsch III, D.C.: TD Methods Applied to Mixture of Experts for Learning 9x9 Goevaluation Function. In: Proceedings of IEEE/INNS Intl. Joint Conf. on Neural Networks, Washington, DC, pp. 3734–3739 (1999)
21. Dasarathy, V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
22. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, Univ. of California, Irvine, CA (1998)