

A Measure Oriented Training Scheme for Imbalanced Classification Problems

Bo Yuan and Wenhua Liu

Intelligent Computing Lab, Division of Informatics, Graduate School at Shenzhen,
Tsinghua University, Shenzhen 518055, P.R. China
{yuanb, liuwh}@sz.tsinghua.edu.cn

Abstract. Since the overall prediction error of a classifier on imbalanced problems can be potentially misleading and biased, it is commonly evaluated by measures such as G-mean and ROC (Receiver Operating Characteristic) curves. However, for many classifiers, the learning process is still largely driven by error based objective functions. As a result, there is clearly a gap between the measure according to which the classifier is to be evaluated and how the classifier is trained. This paper investigates the possibility of directly using the measure itself to search the hypothesis space to improve the performance of classifiers. Experimental results on three standard benchmark problems and a real-world problem show that the proposed method is effective in comparison with commonly used sampling techniques.

Keywords: Imbalanced Datasets, Neural Networks, ROC, G-Mean, SMOTE

1 Introduction

The challenging issue of imbalanced datasets is inevitable in many real-world data mining applications, such as network intrusion detection, video surveillance, oil spills detection in satellite radar images, diagnoses of rare medical conditions and text categorization [2, 11]. These applications share a common characteristic: samples from one class are rare (referred to as minority or positive samples), compared to the number of samples in other classes (referred to as majority or negative samples).

For example, in medical diagnosis applications, it is important to build a predictive model that can reliably identify people with high risk of acquiring certain disease in the earliest stage [18]. However, abnormal samples typically only account for a small fraction of all subjects under test, resulting in a highly imbalanced dataset. Note that a naïve model that simply classifies all subjects as being negative can still achieve high overall prediction accuracies but is otherwise useless as it is incapable of identifying positive samples.

The major challenge comes from the fact that the rarely occurring samples are usually overwhelmed by the majority class samples so that they are much harder to be identified. In the meantime, traditional learning algorithms usually aim at achieving the lowest overall misclassification rate (i.e., use an error based objective function to search the hypothesis space), which creates an inherent bias in favor of the majority classes because the rare class has less impact on accuracy.

Strictly speaking, almost all real-world datasets are imbalanced and how to train and evaluate a classifier taking into account all classes is an important research question. In recent years, this topic has attracted more and more attention from the research community, focusing on mainly two aspects: informative performance measures and how to improve the performance of classifiers [2]. Consequently, some more appropriate measures such as G-mean, ROC, Lift analysis and F-measure have been employed. For example, ROC is very flexible as it assumes no fixed threshold values, and can be used to project a complete image of the classifier in face of the tradeoff between true positive and false positive rates.

In the meantime, various sampling techniques have been proposed, aiming at directly manipulating the original dataset to modify the class distributions. The original dataset can be either over-sampled or under-sampled to increase the influence of positive samples or to reduce the dominance of negative samples. Although these methods do alleviate the challenge to some extent, they also bring in new issues. For instance, the under-sampling methods may unintentionally remove important samples and are not economic when samples are expensive to acquire; the over-sampling methods, on the other hand, may introduce samples that are infeasible in the specific domain and/or lead to overfitting. After all, there is still an open question on the optimal class distributions, which are likely to be domain and classifier dependent.

There is another branch of research on applying Ensemble methods [8] to solving imbalanced problems [4, 7, 12]. For example, misclassification costs can be incorporated into the procedure of weight updating or over-sampling techniques can be embedded to increase the sampling weights for minority samples. This topic is beyond the scope of our current study, which focuses on using a single classifier.

Apart from the many successful applications of sampling methods on solving imbalanced classification problems, there is still a gap between the measure according to which the classifier is to be evaluated and how the classifier is trained. Regardless of what sampling methods are in use, in many situations, the search in the hypothesis space is still driven by error based objective functions. For instance, MSE (Mean Square Error) is commonly used in training Neural Networks. Unfortunately, the relationship between the training error and the measures for after-training evaluation is generally non-trivial. In idealized situations where the dataset can be perfectly separated, the classifier will have zero misclassification error with a possibly very small MSE value. In this situation, commonly used measures such as G-mean, Lift analysis, AUC (Area Under Curve, a performance metric for ROC curves) will also reach their maximum values. However, in many cases, such classifiers may not exist or, due to the local optima in the search space, cannot be found in practice.

In order to bridge the gap between the objective function and the performance measure, in this paper, we conduct an investigation on the possibility and effect of directly using performance measures as the objective functions in the training of classifiers. It is expected that the training process will become more targeted and efficient, with the guidance from the more informative objective functions.

Section 2 gives a brief review of the sampling techniques and performance measures for imbalanced classification problems. Section 3 shows how to train a classifier with measure based objective functions. Experimental results are presented in Section 4 and this paper is concluded in Section 5 with some discussions and a number of directions for future work.

2 Techniques for Imbalanced Problems

Existing techniques for imbalanced classification problems can be roughly grouped into two topics: how to train a classifier properly and how to evaluate a classifier in a meaningful way.

2.1 Sampling Methods

Sampling is one of the common data preprocessing techniques. The idea of sampling is to purposefully manipulate the class distributions so that positive samples can be well represented in the training set. Its major advantage is that the classifier and the training algorithm do not need to be changed. The basic version of sampling is to randomly remove some negative samples, called under-sampling, and/or make copies of positive samples, called over-sampling. In under-sampling, some important samples (e.g., samples along the class boundary) may be discarded, resulting in information loss and a less than optimal model. In the meantime, since over-sampling makes exact copies of positive samples, adding no new information to the dataset, it may cause the overfitting problem.

Since the basic version of sampling does not work well in practice, a series of studies have been conducted most of which focused on developing smart heuristic sampling methods [15, 16]. A widely used over-sampling technique is called SMOTE (Synthetic Minority Over-sampling Technique), which creates synthetic samples between each positive sample and one of its neighbors [3]. It can introduce new samples to enrich the dataset and counter the sparsity in the distribution and create larger and more general decision regions compared to over-sampling with replication.

2.2 Performance Measures

For binary classification problems, classifiers are normally evaluated based on the confusion matrix, as shown in Table 1. Given a specific threshold (e.g., 0.5 for continuous outputs within $[0, 1]$), samples are classified as being either positive or negative and the overall prediction accuracy is defined as $(TP+TN)/(TP+FP+TN+FN)$. The major issue is that, for imbalanced problems, a classifier can still achieve a high prediction accuracy by simply marking all samples as being negative. Instead, a good classifier should be able to achieve high accuracies on predicting both positive samples (i.e., high TP values) and negative samples (i.e., high TN values).

Table 1. A confusion matrix for binary classification problems. TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Based on the confusion matrix, two popular measures have been proposed: G-mean and F-measure, defined as below:

$$G\text{-mean} = (Acc^+ \times Acc^-)^{1/2} \quad (1)$$

$$\text{where } Acc^+ = \frac{TP}{TP + FN}; \quad Acc^- = \frac{TN}{TN + FP}$$

$$F\text{-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2)$$

$$\text{where } Precision = \frac{TP}{TP + FP}; \quad Recall = \frac{TP}{TP + FN} = Acc^+$$

In Eq.1, Acc^+ and Acc^- are the true positive rate and true negative rate respectively, and G-mean represents a tradeoff between the accuracies on both classes. In Eq. 2, *Precision* refers to the proportion of true positive samples among all samples that are predicted as being positive while *Recall* is the proportion of true positive samples that are correctly identified by the classifier, which is the same as Acc^+ .

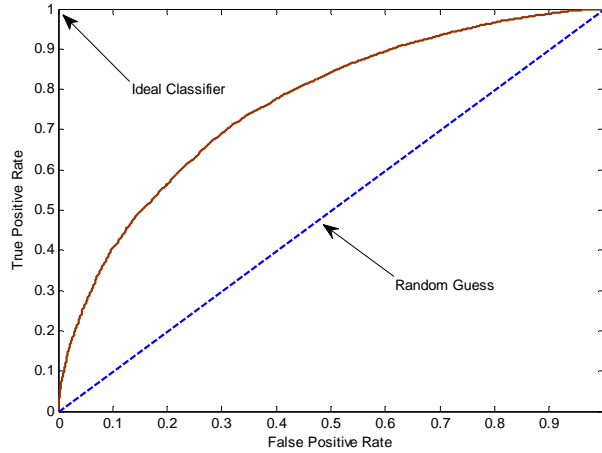


Fig. 1. An illustration of ROC curves. The dashed diagonal line represents the performance of a classifier making random guess. AUC is defined as the area under the curve with a maximum value of 1, corresponding to a classifier that perfectly separates the two classes.

The major advantage of ROC is that it can project a complete image on the behavior of a classifier in face of the tradeoff between true positive rate and true negative rate. The horizontal axis represents the false positive rate ($1 - Acc^-$) while the vertical axis indicates the corresponding true positive rate. With the maximum threshold value, all samples are classified as being negative, corresponding to the origin. By contrast, with the minimum threshold, all samples are classified as being positive, corresponding to the upper right corner.

3 Measure Oriented Training Scheme

Given a hypothesis space containing all candidate classifiers that can be possibly reached, each measure creates a different fitness landscape (i.e., the hypothesis space plus an extra dimension for the measure) with its own structural properties (e.g., the locations of optima). It is unlikely that this fitness landscape is precisely consistent with the one implied by the objective function used in the training of classifiers. As a result, it is conceptually plausible and appealing to directly use the measure of interest as the objective function, in order to bridge the gap between the two landscapes.

In the meantime, many classifiers feature a localized learning pattern in that each time the classifier is updated only a subset of the samples or even a single sample takes effect. For instance, when training a neural network, for each sample, the expected and real outputs are compared and the error information is used to modify the weights and thresholds. When constructing a decision tree, how to split a certain node is only dependent on the subset of samples belonging to that node. In both cases, there is no consideration of the overall performance of the classifier. It is likely that some samples may need to be sacrificed to achieve better global performance.

However, most measures cannot be easily applied in a straightforward manner as it is difficult to derive analytical solutions based on them. Instead, since the measures describe the overall performance of classifiers, it is more appropriate to evaluate and update the classifier as a whole. We must admit that there is no readily available solution to each type of classifiers but for some classifiers, such as neural networks, there is a well developed solution: learning by evolution [20].

The parameters of a neural network are typically encoded into a real-valued vector called chromosome or individual. A population of such individuals represents a set of candidate solutions, which are to be evaluated according to the measure in use and evolved in parallel by evolutionary techniques such as Genetic Algorithms [10]. Each individual is evaluated as a black box and the training process does not require the objective function to have analytical solutions or to be differentiable. Traditionally, the major motivation of using evolutionary techniques over gradient based learning algorithms for training neural networks is to alleviate the curse of local optima. Also, it is possible to evolve the network structure at the same time, solving another well known dilemma. By contrast, the reason that we choose to evolve a neural network is to take its advantage of being flexible with objective functions.

The next question is which measure to choose? Theoretically, all measures can be incorporated into this learning by evolution framework. Since all real-world problems are literally imbalanced to some extent, without loss of generality, we assume that both classes are equally important. For problems where the positive class has significantly higher values, we regard them as falling into another category called minority mining problems, which is beyond the scope of this paper.

Furthermore, some measures such as ROC curves do create another issue: the fitness landscapes are not search friendly. Since ROC only considers the order of samples in terms of the classifier's outputs, there are potentially an infinite number of classifiers with the same AUC value, which creates a search space with many flat areas, a nightmare for all optimization techniques. Also, a classifier with high AUC values may have exceptionally large MSE values (e.g., all samples have output values close to 1 or 0). As a result, G-mean was selected as the measure in our studies.

4 Experiment

To validate the proposed measure oriented training (MOT) scheme, a series of experiments were conducted with standard feedforward neural networks and G-mean as the classifier and the measure respectively. The objective was not to perform a comprehensive and competitive test against existing state-of-the-art techniques. Instead, the major motivation was to demonstrate its general effectiveness and explore its performance with regard to the property of datasets.

4.1 Specification

In experimental studies, there are many factors that can put an impact on the final outcomes. In order to ensure a comparison that is as fair as possible and improve the replicability of the results, in our studies, all parameters were chosen without any specific tuning and were kept unchanged as we were not interested in finding out the best setting for each specific dataset. Also, the standard GA routines implemented in Matlab 2009 were used to reduce any coding related effects.

Table 2 gives a summary of the key experimental settings. Four datasets were used as benchmarks three of which were from the UCI Repository [19] and the last one contained real customer data from a major national bank in China. Note that, in imbalanced datasets, there are often only a handful of positive samples available and running a 10-fold cross validation will result in test sets with few positive samples. As a result, all datasets were randomly divided into the training set and the test set. Some of the properties of the datasets are shown in Table 3.

Table 2. Experimental settings used in the following studies.

Parameters	Values
NN: Number of Input Nodes	The dimension of dataset
NN: Number of Hidden Nodes	5
GA: Population Size	200
GA: Initial Range	[-5, 5]
GA: Other Parameters	As default
SMOTE Sampling Ratio	100% for Cancer; 500% for others

Table 3. The four datasets used in the experimental studies.

Datasets	Number of Attributes	Number of Instances	Proportion of Positive Samples
Cancer	9	683	34.99%
Yeast	8	1484	3.44%
Wine	11	4898	3.67%
Churn	27	1524	4.79%

4.2 Data Preprocessing

All datasets were normalized so that the attribute values were within the range of [0, 1] and all samples with missing values were removed. The positive and negative samples were labeled by “1” and “0” respectively. The Cancer dataset was created from the Wisconsin Breast Cancer Database [17] by removing its 16 samples with missing values (from totally 699 samples). The original Yeast Dataset [13] is a multiclass problem and the class named “ME2” was arbitrarily chosen to be the positive class while all other 9 classes were merged as the negative class, creating a highly imbalanced binary classification problem. The Wine dataset was created from the Wine Quality Dataset (white wine) [5] by marking all samples with scores no less than 8 out of 10 as the positive samples and all other samples as the negative ones. The Churn dataset consisted of various attributes of bank customers such as age, gender, profession, income and so on and was used to predict whether a customer was going to opt out of the service. From Table 3, it is clear that the last three datasets are expected to create much greater challenge for classifiers without appropriate techniques for handling imbalanced class distributions.

4.3 Results

The neural network was evolved by a GA with three training schemes: training based on MSE (**Baseline**), training based on MSE with oversampled training data (**SMOTE**) and training based on G-mean (**MOT**). Each scheme was tested on each dataset for 10 independent trials (the GA is a stochastic optimization algorithm) and the true positive rate, the true negative rate and the G-mean value on the test set were recorded. The average results (with standard deviations for G-mean) are shown in Tables 4-7. It is clear that the Cancer dataset was everyone’s game and it was a two-horse race on the Yeast dataset while MOT stood out of the crowd on the Wine dataset and the Churn dataset. The three schemes were also tested on a few different pairs of training set and test set for each dataset, which produced similar performance patterns.

Table 4. Experimental results on the Cancer dataset.

Methods	True Positive Rate	True Negative Rate	G-mean
Baseline	0.97984	0.96146	0.97058±0.0090
SMOTE	0.98488	0.95688	0.97068±0.0072
MOT	0.98152	0.95778	0.96952±0.0096

Table 5. Experimental results on the Yeast dataset.

Methods	True Positive Rate	True Negative Rate	G-mean
Baseline	0.20716	0.99064	0.39388±0.25
SMOTE	0.62142	0.94086	0.76390±0.036
MOT	0.69288	0.91416	0.79568±0.020

Table 6. Experimental results on the Wine dataset.

Methods	True Positive Rate	True Negative Rate	G-mean
Baseline	0	1	0±0.00
SMOTE	0.19766	0.95628	0.42042±0.12
MOT	0.71164	0.7330	0.72192±0.011

Table 7. Experimental results on the Churn dataset.

Methods	True Positive Rate	True Negative Rate	G-mean
Baseline	0.06154	0.99084	0.18968±0.18
SMOTE	0.44104	0.90214	0.62926±0.046
MOT	0.63076	0.87196	0.74064±0.040

4.4 Analysis

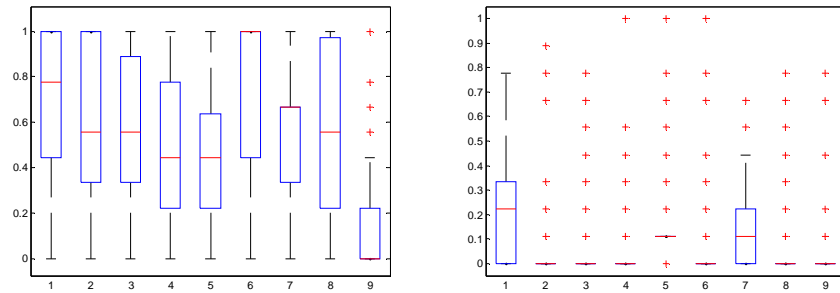


Fig. 2. The box plots of the Cancer dataset: Positive Class (left) and Negative Class (right). The horizontal axis represents the 9 attributes and the vertical axis shows the attribute values.

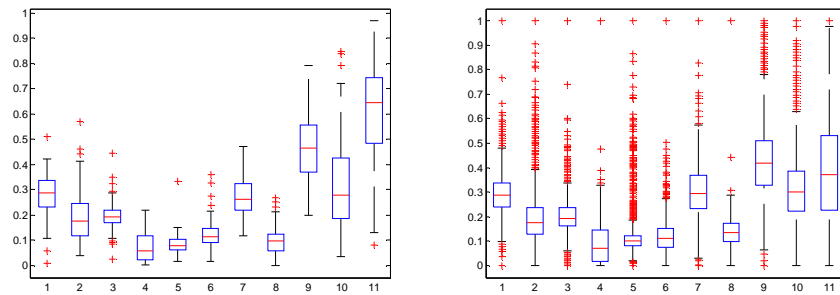


Fig. 3. The box plots of the Wine dataset: Positive Class (left) and Negative Class (right). The horizontal axis represents the 11 attributes and the vertical axis shows the attribute values.

In addition to the quantitative results, it would be interesting to move one step further to address the *why* part of the story. Certainly, for high dimensional datasets, it is difficult to visualize the distributions of data and the decision boundaries. Here, we show the box plots of the Cancer dataset on which Baseline performed very well and the Wine dataset on which Baseline performed extremely badly. Fig. 2 shows that the positive samples and negative samples of the Cancer dataset are reasonably well separated (the horizontal axis shows the attributes). It is reasonable to speculate that the decision boundaries were somewhere between the two classes, as evidenced by the small MSE values (Baseline: 0.02752; SMOTE: 0.03146; MOT: 0.04264).

There is a totally different situation on the Wine dataset where the two classes overlap significantly. In order to achieve a small MSE value, it is tempting to classify all samples as being negative as the positive samples only account for less than 4% of the dataset. On the other hand, in order to achieve a high G-mean value, a large portion of positive samples must be classified correctly, even at the cost of misclassifying some negative samples. In fact, the average MSE value was 0.03310 for Baseline and 0.25648 for MOT. This is a clear example where the MSE based objective function does not agree with the G-mean measure.

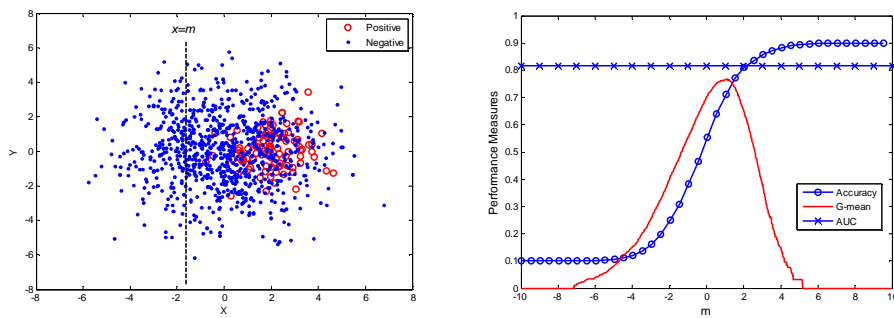


Fig. 4. Illustration of a 2D dataset where the positive class and the negative class overlap significantly (left) and the values of overall accuracy, G-mean and AUC as the decision boundary ($x=m$) moves horizontally (right).

To better demonstrate the property of imbalanced datasets and the relationship among the measures, a 2D dataset was created with 1000 positive samples and 9000 negative samples. The two classes overlapped significantly as shown in Fig. 4 (left). Note that only 10% samples were plotted for better visual effect. For simplicity, the classifier was assumed to be a vertical line (its output was defined as $x-m$) and samples on the right hand side of the line $x=m$ were classified as being positive while samples on the left hand side were classified as being negative.

As the value of m changed from -10 to +10 continuously, the decision boundary moved horizontally from left to right and, at each position, the corresponding values of overall accuracy, G-mean and AUC were recorded as shown in Fig. 4 (right). The patterns of these measures were quite different. The AUC value was constant as the order of samples along the horizontal axis did not change with m . In the meantime, the overall accuracy monotonously increased from 0.1 to 0.9 (the positive samples accounted for 10% of the dataset). By contrast, G-mean reached its peak with $m \approx 1$ when there was a good balance between TP and TN. Note that the value of G-mean reduced to zero when the overall accuracy was at its top.

5 Conclusion

In this paper, we approached the imbalanced classification problems from a new angle. Instead of trying to manipulate the datasets through sampling to change the class distributions or assigning different costs to classes, we proposed to explicitly use the measure itself as the objective function when searching the hypothesis space. This scheme is conceptually plausible as the learning process will become more targeted and efficient by bridging the gap between the traditional error based objective functions and the measures of interest.

The results on three benchmark datasets as well as a real world dataset suggested that, as the challenge of the datasets went up, the advantage of the proposed scheme became more distinctive. Certainly, it is too early to make any conclusive claim on the comparison between this measure oriented training scheme and existing techniques for classifying imbalanced datasets, which requires more extensive and rigorous empirical and theoretical studies. Nevertheless, it offers a new perspective for developing more effective approaches to imbalanced datasets. In fact, since the performance measures in data mining problems are often conflicting with each other, the idea of applying multiobjective evolutionary techniques has become increasingly popular in recent years [14], with some interesting applications in the domain of imbalanced classification problems [1, 6, 9].

As to future work, for classifiers that cannot be evolved in the straightforward manner, other strategies need to be developed to incorporate this measure oriented objective function. In the meantime, since Ensemble methods can adaptively modify the weights of samples, influencing the class distributions in a more informative way, it is also interesting to investigate the possibility of combining measure oriented training with Ensemble methods.

Acknowledgement

This work was supported by the Scientific Research Foundation for Returned Overseas Scholars, Ministry of Education, P.R. China and National Natural Science Foundation of China (No. 60905030 and No. 61003100). The authors are also grateful to the anonymous reviewers for their very helpful comments.

References

1. Bhowan, U., Zhang, M.J., Johnston, M.: Multi-Objective Genetic Programming for Classification with Unbalanced Data. In: Twenty-Second Australasian Conference on Artificial Intelligence, pp. 370--380 (2009)
2. Chawla, N.V.: Data Mining for Imbalanced Datasets: An Overview. In: Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, pp. 853--867. Springer (2005)

3. Chawla, N.V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: SMOTE: Synthetic Minority Oversampling Technique. *Journal of Artificial Intelligence Research* 16, 321--357 (2002)
4. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In: *Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 107--119 (2003)
5. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling Wine Preferences by Data Mining from Physicochemical Properties. *Decision Support Systems* 47(4), 547--553 (2009)
6. Ducange, P., Lazzerini, B., Marcelloni, F.: Multi-Objective Genetic Fuzzy Classifiers for Imbalanced and Cost-Sensitive Datasets. *Soft Computing* 14(7), pp. 713--728 (2010)
7. Fan, W., Stolfo, S.J., Zhang, J., Chan, P.K.: AdaCost: Misclassification Cost-Sensitive Boosting. In: *Sixteenth International Conference on Machine Learning*, pp. 97--105. Morgan Kaufmann (1999)
8. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: *Thirteenth International Conference on Machine Learning*, pp. 148--156 (1996)
9. García, S., Aler, R., Galván, I.M.: Using Evolutionary Multiobjective Techniques for Imbalanced Classification Data. In: *Twentieth International Conference on Artificial Neural Networks*, pp. 422--427 (2010)
10. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley (1989)
11. Han, S.L., Yuan, B., Liu, W.H.: Rare Class Mining: Progress and Prospect. In: *2009 Chinese Conference on Pattern Recognition*, pp. 137--141. IEEE Press (2009)
12. Hoens, T.R., Chawla, N.V.: Generating Diverse Ensembles to Counter the Problem of Class Imbalance. In: *Fourteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 488--499 (2010)
13. Horton, P., Nakai, K.: A Probabilistic Classification System for Predicting the Cellular Localization Sites of Proteins. In: *Fourth International Conference on Intelligent Systems for Molecular Biology*, pp. 109--115 (1996)
14. Jin, Y.C., Sendhoff, B.: Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies. *IEEE Transactions on Systems, Man and Cybernetics — Part C: Applications and Reviews* 38 (3), 397--415 (2008)
15. Kubat, M., Matwin, S.: Addressing the Curse of Imbalanced Training Sets: One Sided Selection. In: *Fourteenth International Conference on Machine Learning*, pp. 179--186. Morgan Kaufmann (1997)
16. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under-Sampling for Class-Imbalance Learning. In: *Sixth International Conference on Data Mining*, pp. 965--969 (2006)
17. Mangasarian, O.L., Setiono, R., Wolberg, W.H.: Pattern Recognition via Linear Programming: Theory and Application to Medical Diagnosis. In: Coleman, T.F., Li, Y. (eds.) *Large-Scale Numerical Optimization*, pp. 22--30. SIAM Publications (1990)
18. Qu, X.Y., Yuan, B., Liu, W.H.: A Predictive Model for Identifying Possible MCI to AD Conversions in the ADNI Database. In: *Second International Symposium on Knowledge Acquisition and Modeling*, vol. 3, pp. 102--105. IEEE Press (2009)
19. UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml>
20. Yao, X.: Evolving Artificial Neural Networks. *Proceedings of the IEEE* 87(9), 1423--1447 (1999)