Front. Comput. Sci.
DOI 10.1007/s11704-012-2943-8

RESEARCH ARTICLE

Measure oriented training: a targeted approach to imbalanced classification problems

Bo YUAN (✉), Wenhuang LIU

Division of Informatics, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China

Abstract Since the overall prediction error of a classifier on imbalanced problems can be potentially misleading and biased, alternative performance measures such as G-mean and F-measure have been widely adopted. Various techniques including sampling and cost sensitive learning are often employed to improve the performance of classifiers in such situations. However, the training process of classifiers is still largely driven by traditional error based objective functions. As a result, there is clearly a gap between the measure according to which the classifier is evaluated and how the classifier is trained. This paper investigates the prospect of explicitly using the appropriate measure itself to search the hypothesis space to bridge this gap. In the case studies, a standard three-layer neural network is used as the classifier, which is evolved by genetic algorithms (GAs) with G-mean as the objective function. Experimental results on eight benchmark problems show that the proposed method can achieve consistently favorable outcomes in comparison with a commonly used sampling technique. The effectiveness of multi-objective optimization in handling imbalanced problems is also demonstrated.

Keywords imbalanced datasets, genetic algorithms (GAs), neural networks, G-mean, synthetic minority over-sampling technique (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 spill detection in satellite radar images, diagnosis of rare medical conditions, and text categorization [1,2]. 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 a high risk of acquiring a certain disease in the earliest stage [3]. However, abnormal samples typically only account for a small fraction of all subjects under test, resulting in a highly imbalanced dataset. Note that a naive model that simply classifies all subjects as being negative can still achieve high overall predication 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 identify. 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 discovering how to train and evaluate a classifier that takes into account all classes is an important research problem. In recent years, this topic has attracted more and more attention from the research community, focusing mainly on two aspects: informative performance measures

Received August 10, 2011; accepted December 11, 2011

E-mail: yuanb@sz.tsinghua.edu.cn

and how to improve the performance of classifiers [1]. Consequently, more appropriate measures such as G-mean, receiver operating characteristic (ROC), lift analysis, and F-measure have been employed, which focus on individual measures instead of the overall performance.

In the meantime, various sampling techniques have been proposed, aimed 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 introduce 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 the open question of the optimal class distributions, which are likely to be domain and classifier dependent.

There is another branch of research into extending ensemble methods [4] to solving imbalanced problems [5–7]. 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.

Despite the many successful applications of sampling methods on solving imbalanced classification problems, there is still a gap between the measure with which the classifier is 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, mean square error (MSE) is commonly used in training neural networks. Unfortunately, the relationship between the training error and the measures for post-training evaluation is generally non-trivial. In idealized situations where the dataset can be perfectly separated, the classifier will have zero misclassification error, possibly with a very small MSE value. In this situation, commonly used measures such as G-mean, lift analysis, and area under curve (AUC) 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.

Our previous work has demonstrated how the gap between the objective function and the performance measure can be bridged by directly using performance measures as the objective functions in the training of classifiers [8]. It is expected that the training process will become more targeted and efficient with the guidance from more informative objective functions. In this paper, we conduct substantially more comprehensive empirical studies with additional benchmark problems, formal statistical tests, different performance measures and multi-objective optimization techniques.

Section 2 gives a brief review of the sampling techniques and performance measures for imbalanced classification problems. Section 3 presents the core principle of training classifiers with measure oriented objective functions. The major experimental results and some analysis are presented in Section 4 while Section 5 contains some further extensions. This paper is concluded in Section 6 with some discussion 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. However, 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 focus on developing smart heuristic sampling methods [9,10]. A widely used over-sampling technique is called the synthetic minority over-sampling technique (SMOTE), which creates synthetic samples between each positive sample and one of its neighbors [11]. 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, the performance of classi-

fiers is normally evaluated based on the confusion matrix, as shown in Table 1.

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

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 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 (high TP values) and negative samples (high TN values).

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

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

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

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{2}$$

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

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 actual positive samples among all samples that are predicted as being positive while *Recall* is the proportion of actual positive samples that are correctly identified by the classifier, which is the same as Acc^{+}.

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.

Figure 1 shows an example where the classifier is a vertical line and the minority and majority instances are represented by white and gray boxes, respectively. The solid line indicates the classifier with the maximum overall accuracy (95%) but misclassifying all minority class instances. By contrast, the dashed line gives a less appealing overall accuracy of 90% but correctly identifies all minority instances. It is easy to see that, in imbalanced classification problems, the accuracy of a classifier on the majority class may need to be compromised in exchange for improved performance on the minority class. Since the majority class dominates the dataset, this will also result in the sacrifice of the accuracy over the entire dataset.

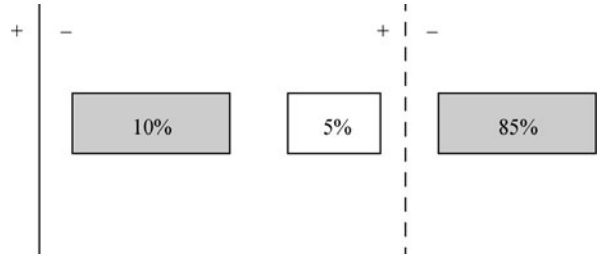

Fig. 1 An imbalanced dataset where the classifier with the best overall performance (solid line, accuracy: 95%) is not consistent with the classifier that can correctly identify the minority class (dashed line, accuracy: 90%)

In the meantime, many classifiers feature a localized learning pattern as 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. Clearly, there is no consideration of the overall performance of the classifier. However, it is likely that some samples may need to be sacrificed to achieve better global performance.

Unfortunately, most measures cannot be easily used in the training process as it is difficult to derive analytical solutions based on them. However, for some classifiers, such as neural networks, there is a well developed solution: learning by evolution [12].

The parameters of a neural network are typically encoded into a real-valued vector called a 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 that are loosely based on the principles of natural selection such as genetic algorithms (GAs) [13]. 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 measure oriented training framework. Since almost all real-world problems are literally imbalanced to some extent, without loss of generality, we assume that both classes are equally important. As a result, we select G-mean as the performance measure in our studies.

4 Experiments

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

4.1 Specification

In experimental studies there are many factors that can impact the final outcome. In order to ensure a comparison that is as fair as possible and maintain the replicability of the results, in our studies, all parameter values were chosen without any specific tuning and were kept unchanged throughout the entire study. Also, the standard GA routines provided by Matlab 2010 were used to reduce implementation related influence.

Table 2 gives a summary of the key experimental settings in which only the number of hidden units and the oversampling rate were manually set.

Table 2 Experimental settings

Parameters	Values
NN: number of input nodes	The dimension of dataset
NN: number of hidden nodes	5
GA parameters	As default[1)]
SMOTE sampling ratio	500%

Eight imbalanced datasets were used as benchmark problems six of which were adopted from the UCI Repository [14]. The Churn dataset consists 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. The PAKDD-09 dataset was first used in the data mining competition in PAKDD 2009[2)]. Attributes on which all samples had identical values and samples that contained outliers were removed.

Note that, in imbalanced datasets, there are often only a handful of positive samples available and running a traditional ten-fold cross validation will result in test sets with few positive samples. As a result, each dataset was randomly divided into training and test sets of approximately equal sizes and this procedure was repeated 30 times.

4.2 Data preprocessing

All datasets were normalized so that attribute values were within the range of [0, 1] and all samples with missing values were removed. The positive and negative samples were labeled “1” and “0”, respectively.

The Abalone dataset was created by merging classes 16–29 as the positive class and all other classes as the negative class. The Cancer dataset was created from the Wisconsin Breast Cancer Database [15] by removing the 16 samples with missing values (from a total of 699 samples). The Car dataset was created by merging the “good” class and the “v-good” class as the positive class and all other classes were treated as the negative class. The Covertype dataset was created by randomly selecting 1% of samples from the original dataset with more than half a million samples and merging classes 4–7 as the positive class. The Wine dataset was created from the Wine Quality Dataset (white wine) [16] by marking all samples with scores no less than eight out of ten as the positive samples and all other samples as negative ones. The original Yeast Dataset [17] is a multiclass problem and the class named “ME2” was arbitrarily chosen as the positive class while the remaining nine classes were merged as the negative class.

From Table 3, it is clear that all except the Cancer dataset are expected to create significant challenge for classifiers without appropriate techniques for handling imbalanced class distributions.

4.3 Results

The neural network was evolved by the GA with three different training schemes: training based on MSE (Baseline), training based on MSE with over-sampled training data (SMOTE), and training based on G-mean (MOT). For each cross-validation, ten independent trials were conducted to

[1)] http://www.mathworks.com/help/toolbox/gads/ga.html
[2)] http://sede.neurotech.com.br/PAKDD2009

Table 3 The eight datasets used in the experiments

Datasets	Number of attributes	Number of instances	Proportion of positive samples/%
Abalone	8	4 177	6.25
Cancer	9	683	34.99
Car	6	1 728	7.76
Churn	27	1 524	4.79
Covertype	54	5 822	8.52
PAKDD-09	20	49 981	19.75
Wine	11	4 898	3.67
Yeast	8	1 484	3.44

account for the randomness of the GA, resulting in a total of 8 (datasets) ×3 (schemes) ×30 (cross-validations) ×10 (GA trials) = 7 200 trials to be conducted.

For each set of the ten trials, corresponding to the same cross-validation, the minimum, average, and maximum G-mean values were recorded. Tables 4–11 shows the results averaged over the 30 independent trials along with the standard deviations. The p-values of the two-sample t-test are given in Table 12.

Table 4 Experimental results on Abalone

Methods	Min	Mean	Max
MOT	0.652±0.022	0.725±0.020	0.795±0.019
SMOTE	0.465±0.127	0.628±0.035	0.711±0.029
Baseline	0.004±0.021	0.180±0.040	0.336±0.046

Table 5 Experimental results on Cancer

Methods	Min	Mean	Max
MOT	0.951±0.012	0.967±0.007	0.977±0.005
SMOTE	0.965±0.009	0.972±0.007	0.977±0.006
Baseline	0.959±0.010	0.968±0.008	0.975±0.007

Table 6 Experimental results on Car

Methods	Min	Mean	Max
MOT	0.879±0.028	0.925±0.012	0.956±0.011
SMOTE	0.908±0.020	0.933±0.017	0.955±0.014
Baseline	0.707±0.069	0.804±0.034	0.863±0.033

Table 7 Experimental results on Churn

Methods	Min	Mean	Max
MOT	0.728±0.042	0.794±0.023	0.835±0.021
SMOTE	0.636±0.055	0.712±0.047	0.781±0.049
Baseline	0.000±0.000	0.074±0.062	0.255±0.103

In Table 12, of the eight benchmark problems, the Cancer dataset produced the best results: all training schemes worked equally well. On the Car and Covertype datasets, MOT and SMOTE worked equally well and in all other cases the superiority of MOT over SMOTE and Baseline was statistically significant. The non-parametric rank sum test was also conducted, which produced as consistent results as the t-test.

Table 8 Experimental results on Covertype

Methods	Min	Mean	Max
MOT	0.699±0.024	0.748±0.011	0.786±0.011
SMOTE	0.690±0.038	0.750±0.010	0.785±0.013
Baseline	0.004±0.021	0.191±0.057	0.398±0.059

Table 9 Experimental results on PAKDD-09

Methods	Min	Mean	Max
MOT	0.574±0.008	0.589±0.004	0.602±0.004
SMOTE	0.378±0.095	0.491±0.019	0.553±0.018
Baseline	0.000±0.000	0.016±0.010	0.082±0.040

Table 10 Experimental results on Wine

Methods	Min	Mean	Max
MOT	0.650±0.036	0.692±0.017	0.722±0.013
SMOTE	0.134±0.114	0.311±0.071	0.437±0.056
Baseline	0.000±0.000	0.001±0.002	0.004±0.020

Table 11 Experimental results on Yeast

Methods	Min	Mean	Max
MOT	0.711±0.059	0.791±0.041	0.844±0.033
SMOTE	0.660±0.077	0.729±0.050	0.781±0.042
Baseline	0.009±0.052	0.184±0.085	0.342±0.108

Table 12 The p-values of the two-sample t-test based on the best results of each training scheme

Datasets	MOT/SMOTE	SMOTE/Baseline	MOT/Baseline
Abalone	0.000	0.000	0.000
Cancer	0.877	0.304	0.232
Car	0.853	0.000	0.000
Churn	0.000	0.000	0.000
Covertype	0.687	0.000	0.000
PAKDD-09	0.000	0.000	0.000
Wine	0.000	0.000	0.000
Yeast	0.000	0.000	0.000

Certainly, the effectiveness of SMOTE is, to some extent, influenced by the sampling ratio (500% in our experiments, a typical value used in the literature [5,11]), which is itself problem dependent. Nevertheless, the comparison above still demonstrates the promising potential of MOT in handling imbalanced datasets.

4.4 Analysis

In addition to the quantitative results, it would be interesting to zoom into the datasets a little further. Certainly, for high dimensional datasets, it is difficult to visualize the distribution of data and the decision boundaries. Here, we show the box plots of the Cancer dataset on which the Baseline performed very well and the Wine dataset on which the Baseline

performed badly. Figure 2 shows that the positive samples and negative samples of the Cancer dataset are reasonably well separated (the horizontal axis shows the attributes and the vertical axis shows the attribute values). Consequently, the decision boundaries could be somewhere between the two classes, resulting in good MSE and G-mean values simultaneously, which was confirmed by the small MSE values of the trained classifiers.

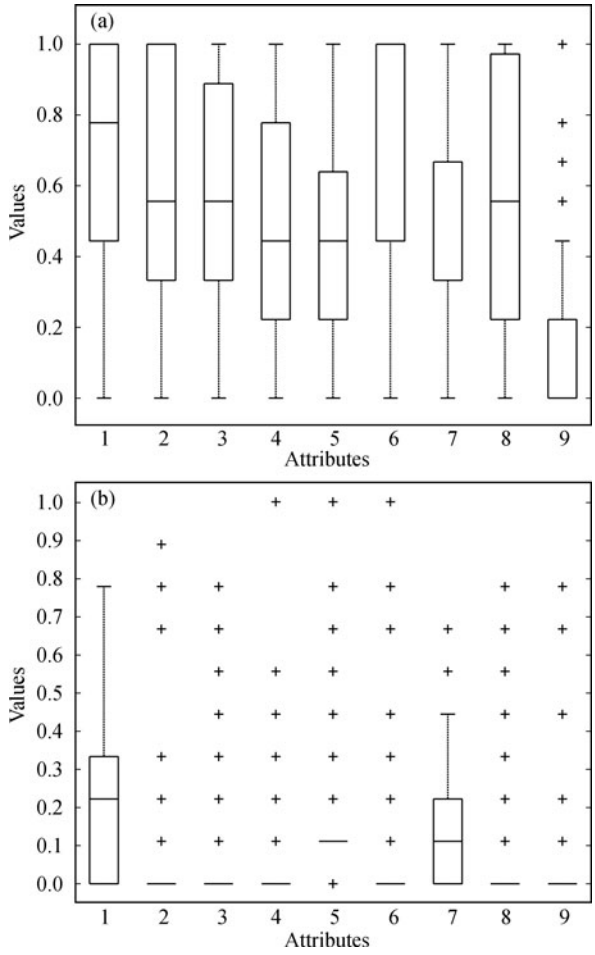

Fig. 2 The box plots of the Cancer dataset: (a) positive class; (b) negative class

Furthermore, suppose that all samples are clustered based on their class labels (i.e., assign all positive samples to one cluster and all negative samples to another cluster). The silhouette plot can provide some visual evidence on the distribution of samples. Figure 3 shows that, in Cancer, the vast majority of negative samples have silhouette values close to 1, indicating a compact cluster. Although the cluster corresponding to the positive class is a bit loose as a few samples have negative silhouette values (i.e., closer to the negative cluster than to the positive cluster), there is still a well shaped clustering pattern in the dataset (i.e., positive and negative classes are reasonably well separated).

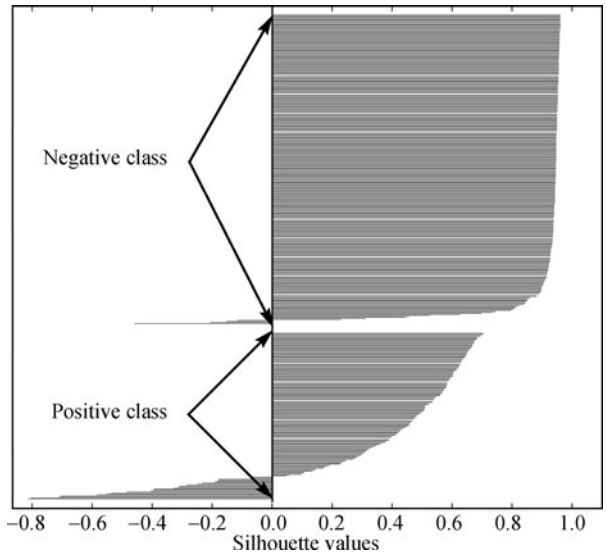

Fig. 3 The silhouette plot of the Cancer dataset

There is a totally different situation on the Wine dataset where the two classes overlap significantly as shown in Fig. 4. 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, which is confirmed by the large MSE values of the trained classifiers. This is a clear example where the MSE based objective function does not agree with the G-mean measure.

To better demonstrate the relationship between the measures, a 2D dataset was created with 1 000 positive samples and 9 000 negative samples. The two classes overlap significantly as shown in Fig. 5(a). 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 the overall accuracy, G-mean and AUC were recorded as shown in Fig. 5(b). The patterns of these measures were quite different. The AUC value was held constant as the order of samples in terms of the outputs of the classifier did not change with m. In the meantime, the overall accuracy monotonously increased from 0.1 to 0.9 (the positive samples account for

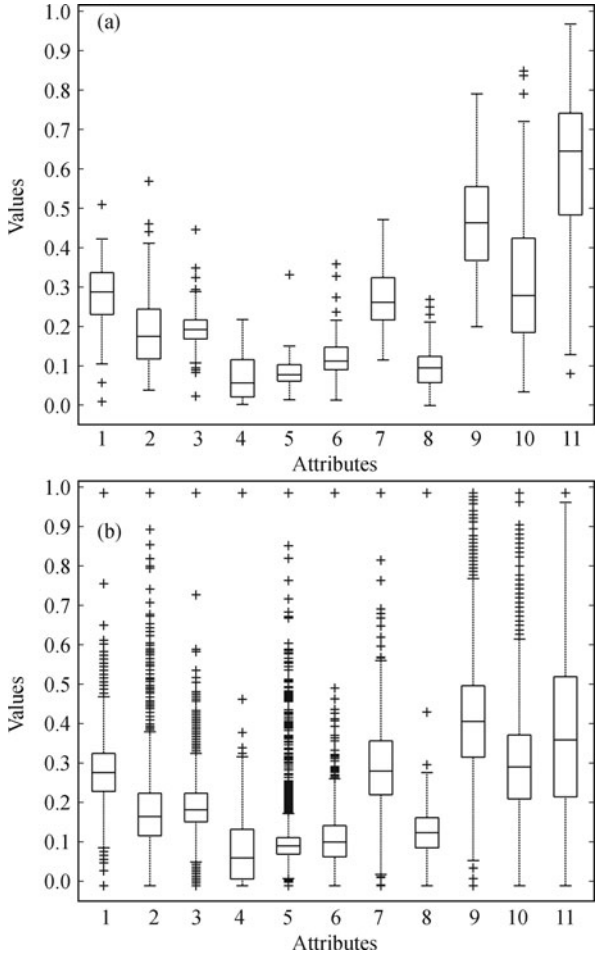

Fig. 4 The box plots of the Wine dataset: (a) positive class; (b) negative class

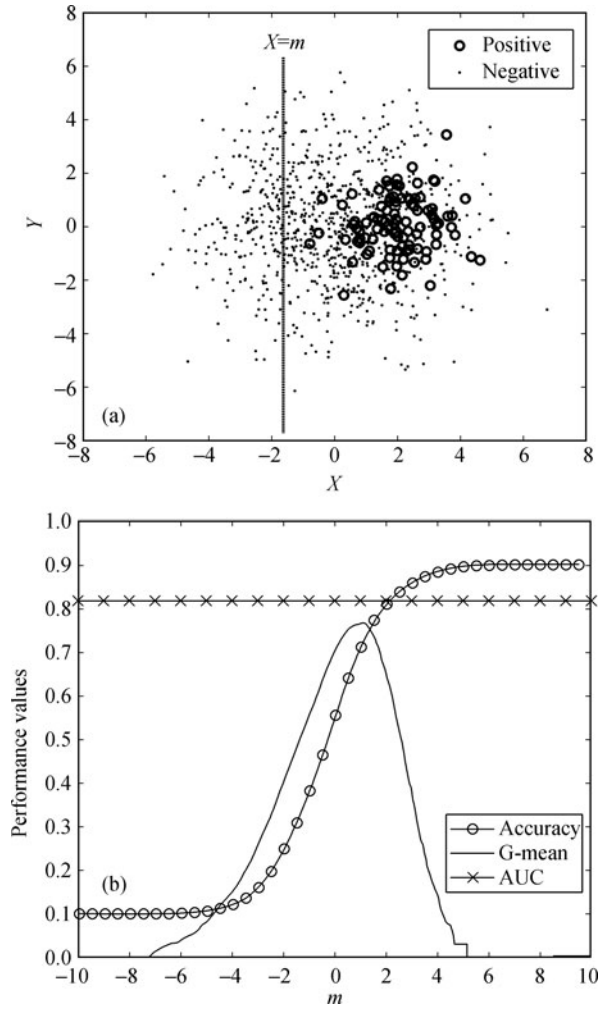

Fig. 5 Illustration of a 2D dataset where (a) the positive class and the negative class overlap significantly; (b) the values of the overall accuracy, G-mean and AUC as the decision boundary ($x = m$) moves horizontally

10% of the dataset). In contrast, G-mean achieved its peak value 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 reached its top value as all positive samples were misclassified.

5 Extensions

The experimental results in Section 4 have demonstrated the effectiveness of MOT using G-mean as the performance measure on a number of benchmark problems. In this section, we will further explore the effectiveness and the properties of MOT.

5.1 The impact of sampling

Since sampling techniques such as SMOTE can help solve the imbalanced problems in general, it is natural to consider whether conducting over-sampling on the original dataset can improve the performance of MOT. Table 13 shows the results of MOT on the eight datasets pre-processed by SMOTE. Compared to Tables 4–11, it can be seen that MOT received little if any benefit from the adoption of SMOTE. Although this finding may be a bit surprising, it shows that MOT itself is sufficient in handling imbalanced datasets and can reliably find good solutions without the need of over-sampling.

Table 13 Experimental results of SMOTE + MOT

Datasets	Min	Mean	Max
Abalone	0.647±0.043	0.736±0.022	0.801±0.019
Cancer	0.956±0.008	0.969±0.005	0.978±0.004
Car	0.893±0.023	0.929±0.014	0.958±0.009
Churn	0.740±0.038	0.798±0.026	0.844±0.020
Covertype	0.695±0.026	0.749±0.011	0.788±0.015
PAKDD-09	0.574±0.007	0.592±0.003	0.604±0.003
Wine	0.671±0.020	0.702±0.014	0.730±0.013
Yeast	0.743±0.055	0.804±0.033	0.846±0.030

Table 14 Experimental results on Wine (F-measure)

Methods	Min	Mean	Max
MOT	0.126±0.033	0.177±0.017	0.218±0.022
SMOTE	0.039±0.044	0.102±0.036	0.172±0.030
Baseline	0.000±0.000	0.000±0.000	0.000±0.000

Table 15 Experimental results on Yeast (F-Measure)

Methods	Min	Mean	Max
MOT	0.201±0.084	0.328±0.056	0.430±0.047
SMOTE	0.301±0.066	0.372±0.057	0.431±0.062
Baseline	0.007±0.020	0.072±0.038	0.189±0.073

5.2 Other performance measures

To demonstrate the versatility of MOT, we also used the F-measure defined in Eq. (2) as the performance measure. Tables 14 and 15 show the results of MOT on Wine and Yeast respectively. On the Wine dataset, MOT outperformed SMOTE and Baseline, while on the Yeast dataset, MOT and SMOTE both achieved significantly improved solutions compared to Baseline.

5.3 Multiple measures

Since the performance measures in data mining problems often conflict with each other, the idea of applying multi-objective evolutionary techniques has become increasingly popular in recent years [18], with some interesting applications in the domain of imbalanced classification problems [19–21].

In multi-objective optimization, each solution is described by multiple objective values and the key principle is to, instead of trying to find the best classifier with regard to a single measure, return a set of classifiers with objective values not dominated by others. For example, suppose that each classifier is evaluated by two measures: Acc^+ and Acc^-. Instead of using G-mean, which is a fixed combination of Acc^+ and Acc^-, it is desirable to find a set of classifiers so that none of them is superior or inferior to others in terms of both measures. In other words, this set of candidates represents the tradeoff between Acc^+ and Acc^-. By doing so, users are given the flexibility to choose from a set of candidates based on their specific preferences, which is convenient when the relative importance of TP and TN may change in different application scenarios.

Figure 6 shows the performance of a set of classifiers found by the multi-objective GA routine in Matlab on the Wine dataset. Note that the performance was based on the test set and a small number of classifiers were actually dominated by others.

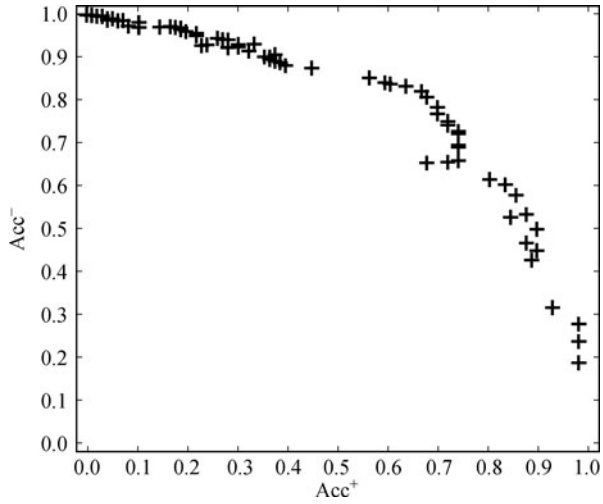

Fig. 6 The performance of a set of tradeoff solutions on the Wine dataset (test set)

6 Conclusions

In this paper, we approached the imbalanced classification problems from a new perspective. Instead of trying to manipulate datasets through sampling to change the class distributions or assigning different costs to classes, we proposed a measure oriented training (MOT) scheme, which explicitly uses the measure itself as the objective function when searching the hypothesis space. Firstly, MOT is conceptually plausible as the learning process will become more targeted by bridging the gap between the traditional error based objective functions and the measures of interest. Secondly, MOT can be easily customized with regard to different measures of interest and with the help of multi-objective optimization techniques MOT can generate a set of candidate classifiers representing tradeoffs between various measures.

Experimental results on eight benchmark datasets suggest that, when coupled with MOT, the neural networks achieved consistently superior performance compared to traditional training schemes based on MSE or SMOTE. Certainly, at this stage, we do not try to make any general claims on MOT, which requires more extensive and rigorous empirical and theoretical studies. Nevertheless, it offers a new perspective for developing more effective approaches to imbalanced datasets.

As to future work, the major challenge remaining is answering the question of how to incorporate the principle of MOT into classifiers such as decision trees and ensemble classifiers, which cannot be conveniently evolved by GAs. Also, it would be interesting to investigate the effectiveness of MOT on multi-class problems and how to customize GAs

or other meta-heuristics to better suit specific scenarios (e.g., different performance measures and classifiers). The application and in-depth analysis of our techniques on some real world problems are also important.

Acknowledgements This work was supported by the Scientific Research Foundation for Returned Overseas Scholars, Ministry of Education, China and the National Natural Science Foundation of China (Grant Nos. 60905030 and 61003100). The authors are also grateful to the anonymous reviewers for their helpful comments.

References

1. Chawla N V. Data mining for imbalanced datasets: an overview. In: Maimon O, Rokach L, eds. Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers. New York: Springer, 2005, 853–867
2. Han S, Yuan B, Liu W. Rare class mining: progress and prospect. In: Proceedings of the 2009 Chinese Conference on Pattern Recognition. 2009, 137–141
3. Qu X, Yuan B, Liu W. A predictive model for identifying possible MCI to AD conversions in the ADNI database. In: Proceeding of the 2nd International Symposium on Knowledge Acquisition and Modeling, Vol 3. 2009, 102–105
4. Freund Y, Schapire R E. Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning. 1996, 148–156
5. Chawla N V, Lazarevic A, Hall L O, Bowyer K W. SMOTEBoost: improving prediction of the minority class in boosting. In: Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. 2003, 107–119
6. Fan W, Stolfo S J, Zhang J, Chan P K. AdaCost: misclassification cost-sensitive boosting. In: Proceedings of the 16th International Conference on Machine Learning. 1999, 97–105
7. Hoens T R, Chawla N V. Generating diverse ensembles to counter the problem of class imbalance. In: Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Part II. 2010, 488–499
8. Yuan B, Liu W. A measure oriented training scheme for imbalanced classification problems. In: Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining Workshop on Biologically Inspired Techniques for Data Mining. 2011, 293–303
9. Kubat M, Matwin S. Addressing the curse of imbalanced training sets: one sided selection. In: Proceedings of the 14th Interactional Conference on Machine Learning. 1997, 179–186
10. Liu X, Wu J, Zhou Z. Exploratory under-sampling for class-imbalance learning. In: Proceedings of the 6th International Conference on Data Mining. 2006, 965–969
11. Chawla N V, Bowyer K W, Hall L O, Kegelmeyer W P. SMOTE: synthetic minority oversampling technique. Journal of Artificial Intelligence Research, 2002, 16(1): 321–357
12. Yao X. Evolving artificial neural networks. Proceedings of the IEEE, 1999, 87(9): 1423–1447
13. Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning. Boston: Addison Wesley, 1989
14. Frank A, Asuncion A. UCI machine learning repository. 2010, http://archive.ics.uci.edu/ml
15. 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. 1990, 22–30
16. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 2009, 47(4): 547–553
17. Horton P, Nakai K. A probabilistic classification system for predicting the cellular localization sites of proteins. In: Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology. 1996, 109–115
18. Jin Y, Sendhoff B. Pareto-based multiobjective machine learning: an overview and case studies. IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews, 2008, 38(3): 397–415
19. Bhowan U, Zhang M, Johnston M. Multi-objective genetic programming for classification with unbalanced data. In: Proceedings of the 22nd Australasian Conference on Artificial Intelligence. 2009, 370–380
20. Ducange P, Lazzerini B, Marcelloni F. Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets. Soft Computing, 2010, 14(7): 713–728
21. García S, Aler R, Galván I. Using evolutionary multiobjective techniques for imbalanced classification data. In: Proceedings of the 20th International Conference on Artificial Neural Networks. 2010, 422–427

Dr. Bo Yuan received his BEng from Nanjing University of Science and Technology, China, in 1998, and his MSc and PhD from the University of Queensland, Australia, in 2002 and 2006, respectively. From 2006 to 2007, he was a research officer on a project funded by the Australian Research Council at the University of Queensland. He is currently an associate professor in the Division of Informatics, Graduate School at Shenzhen, Tsinghua University, China, and a member of the IEEE and the IEEE Computational Intelligence Society. He is mostly interested in data mining, evolutionary computation, and parallel computing.

Prof. Wenhuang Liu received his BEng from Tsinghua University, China, in 1970 and has been a faculty member of Tsinghua University for more than forty years. He was the deputy director of National CIMS Engineering Research Center and the deputy dean of the Graduate School at Shenzhen, Tsinghua University. His research interests include CIMS, operation research, and decision support systems.