Measure optimized wrapper framework for multi-class imbalanced data learning: an empirical study

Peng Cao, Dazhe Zhao and Osmar Zaiane

***Abstract*—Class imbalance is one of the challenging problems for machine learning in many real-world applications. Many methods have been proposed to address and attempt to solve the problem, including re-sampling and cost-sensitive learning. However, the existing methods have room for improvement since the potentially optimal values of the factors associated with best performance are unknown. Moreover most methods only focus on the binary class imbalance problem, thus there is no efficient solution in multi-class imbalanced learning. This paper presents an effective wrapper framework incorporating the evaluation measure into the objective function of cost sensitive learning as well as re-sampling directly, so as to improve the original methods through optimizing factors influencing the performance on the imbalanced data classification. Comprehensive experimental results on various standard benchmark datasets with different ratios of imbalance show that the influence of optimizing parameters on the solutions for learning imbalanced data is critical, and demonstrate the effectiveness of measure-optimized scheme on the imbalanced data learning.**

I. Introduction

Recently, the class imbalance problem has been recognized as a crucial problem in machine learning and data mining [1, 2]. This problem occurs when the training data is not evenly distributed among classes; that is when some classes are significantly larger than others. When data is imbalanced, standard classifiers usually tend to be overwhelmed by the majority class and ignore the minority class examples, resulting in providing unsatisfactory classification performance. This is a consequence of the fact that most traditional classifiers assume an even distribution of examples among classes and assume an equal misclassification cost. Therefore, we need to improve traditional algorithms so as to handle imbalanced data.

These imbalanced data learning methods can be grouped into two categories: the data perspective [3-5] and the algorithm perspective [6-8]. The methods with the data perspective try to balance out the class distribution by re-sampling the data space, either over-sampling instances of the minority class or under-sampling instances of the majority class. The most common method with the algorithm perspective is cost-sensitive learning, which tries to learn more characteristics of samples with the minority class by setting a high cost to the misclassification of a minority class sample. The re-sampling methods or cost sensitive learning indeed improves the classification performance on the imbalanced data to some extent. Nevertheless, there is still room for improvement since the factors affecting the performance are not optimal. In the construction of the processing of re-sampling or cost sensitive learning, the critical parameter (re-sampling level or misclassification cost) plays a crucial role for achieving expected classification results, however they are set to default value without being searched in the parameter space, resulting in suboptimal performance. In addition, the imbalanced data distribution is often accompanied by high dimensionality in real-world data sets such as text classification and bioinformatics. Therefore, high-dimensionality poses additional challenges when dealing with class-imbalanced prediction [9-10]. Thus, it is important to select features that lead to a higher separability among the unequal classes. Furthermore, the feature subset choice influences the appropriate re-sampling ratio or misclassification cost and vice versa, obtaining the optimal critical parameters of imbalanced data learning methods and feature subset must occur simultaneously.

In order to solve the challenges above, we design a novel framework for optimizing cost sensitive learning or re-sampling with the evaluation criteria of the imbalanced distribution, to cope with the multi-class imbalanced data classification. The framework can learn the optimal factors associated with the classification performance of imbalance data learning automatically driven by imbalanced data measures. For a multi-class case, there are two major issues: the search space expands exponentially as the class number increases, and the factors to be searched are mixtures including continuous and discrete variables. Therefore, these two important issues need to be fixed in the training scheme: how to optimize these variables simultaneously and what evaluation criteria to use for guiding their optimization. These two issues are our key step for improving the imbalanced data learning methods.

Our main contributions in this paper are centered around the questions above. To improve the performance of cost-sensitive learning, the factors, including misclassification cost ratio and feature set, are optimized at the same time. Similarly, the re-sampling ratio and feature set are searched in the parameter space simultaneously for achieving the optimal data distribution. We use the targeted performance measure, G-mean [11, 26] and AUC [12] directly to optimize and discover the optimal factors. The purpose is to search for the potentially optimal factors in the parameter space with the highest evaluation score guided by some heuristic optimization function. Our designed

Peng Cao is with the Northeastern University, ShenYang, China (corresponding author e-mail: neusoftcp@gmail.com).

Dazhe Zhao is with the Northeastern University, ShenYang, China

Osmar Zaiane is with the University of Alberta, Edmonton, Canada (e-mail: zaiane@cs.ualberta.ca).

framework can be applied on binary class and multi-class classification. Through extensive experiments on multiple datasets, we demonstrated that this optimization scheme is effective for imbalanced data learning. Although it has been observed that optimizing some parameters associated with the learning performance can improve the traditional methods on class imbalance problems [11, 13-16], up to now there is no thorough investigation about the influence of optimizing parameters with measure oriented on the solutions for learning imbalanced data from two different perspectives.

This paper is organized as follows: Related works are described in Section 2. Our proposed measure optimized framework is presented in Section 3, including MOCS-DT and MOHS-DT. Section 4 demonstrates the experiments and result analysis. Section 5 concludes with general remarks.

II. Related work

A. The common methods for the binary class imbalance

As discussed above, there are two general ways to deal with class imbalance learning: the re-sampling approach independent classifier and the cost sensitive method based on the cost-adaptation. Re-sampling methods only manipulate the original training datasets; therefore it provides a convenient and effective way to deal with imbalanced learning problems using standard classifiers by balancing the instances of the classes. The re-sampling techniques include the under-sampling and over-sampling.

Random under-sampling can cause loss of information so as to affect the performance of a classifier. Some sophisticated under-sampling methods could reduce the influence of important information loss, which only eliminates redundant information or noise, such as Edited Nearest Neighbor Rule under-sampling (ENN) [17] and ACOSampling [18].

Random over-sampling is a non-heuristic method that aims to balance class distribution through the random replication of minority class examples. Many sophisticated deterministic over-sampling methods have been proposed which provide new information avoiding overfitting. A widely used over-sampling technique is called SMOTE, which creates synthetic samples between each positive sample and one of its neighbors [3]. SMOTE is effective to increase the significance of the positive class in the decision region. There exist many methods based on SMOTE for generating more appropriate instances, such as SMOTEBoost [4] and RSM-SMOTE [19].

Cost-sensitive learning is one of the most important topics in machine learning and data mining, and has attracted significant attention in recent years. Cost-sensitive learning methods consider the costs associated with misclassifying examples and treat the different misclassifications differently such as MetaCost [7] and cost-sensitive neural network [6].

Feature selection has recently been extensively studied. In particular, its importance to class imbalance problems was realized and attracted increasing attention from the machine learning and data mining community. A number of researchers have conducted research on using feature selection to combat the class imbalance problem [9-10, 20]. Zheng, et al. [9] suggest that existing measures used for feature selection are not very appropriate for imbalanced data sets. Hulse et al. [20] propose that the wrapper feature selection is a good approach for imbalanced datasets, which can find potentially interesting feature information not captured by other filter techniques.

B. The common methods for multiple class imbalance

Most existing imbalance data learning techniques so far are still limited to the binary class imbalance problem. There are fewer solutions for multi-class imbalance problems, which exist in real-world applications. They have been shown to be less effective or even to cause a negative effect in dealing with multi-class tasks [6]. The experiments in [21] imply that the performance decreases as the number of imbalanced classes increases.

Most existing solutions for multi-class imbalance problems use class decomposition schemes, so as to transform the multi-class classification issue into multiple sub tasks with binary class classification [22]. However, there are some drawbacks: 1) OAO scheme (one-versus-one) may make the training work much more expensive; 2) OAA scheme (one-versus-all) could worsen the imbalanced distribution. Different from the decomposition approaches, Sun et al. develop a cost-sensitive boosting algorithm, named AdaC2 [11]. Wang and Yao proposed an ensemble learning algorithm AdaBoost.NC that combines the strength of negative correlation learning and Boosting ensemble. The AdaBoost.NC working with random over-sampling can deal with the multi-class imbalance data [21].

III. Proposed methods

A. Measure optimized wrapper framework

This paper explicitly treats the measure itself as the objective function when optimizing the approaches to improve the performance of classifiers and discover the best parameters and feature subset. We designed a measure optimized framework for dealing with imbalanced data classification issues.

For the multivariable optimization, especially the hybrid multivariable, the best methods are swarm intelligence techniques. We chose the particle swarm optimization (PSO) [23] as our optimization method because it is very mature and to easy implement. In addition, many experiments claim that PSO has equal effectiveness but superior efficiency over the GA [24]. PSO is a population-based global stochastic search method. PSO optimizes an objective function by a population-based search. The population consists of potential solutions, named particles. These particles are randomly initialized and move across the multi-dimensional search space to find the best position according to an optimization function. During optimization, each particle adjusts its trajectory through the problem space based on the information about its previous best performance (personal best, *pbest*) and the best previous performance of its neighbors (global best, *gbest*). Eventually, all particles will

gather around the point with the highest objective value. The position of individual particles is updated as follows:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (1)$$

With *v*, the velocity calculated as follows:

$$v_{id}^{t+1} = w \times v_{id}^t + c_1 \times r_1 \times (pbest_{id}^t - x_{id}^t) + c_2 \times r_2 \times (gbest^t - x_{id}^t) \quad (2)$$

where v_i^t indicates velocity of particle *i* at iteration *t*, *w* indicates the inertia factor, C_1 and C_2 indicate the cognition and social learning rates, which determine the relative influence of the social and cognition components. r_1 and r_2 are uniformly distributed random numbers between 0 and 1, x_i^t is current position of particle *i* at iteration *t*, $pbest_i^t$ indicates best of particle *i* at iteration *t*, $gbest^t$ indicates the best of the group.

The parameters of imbalanced data learning approaches (misclassification cost or sampling ratio) and feature subset for measure optimized wrapper framework need to be searched at the same time. Thus, the solution in PSO includes two parts: the parameters of imbalanced data learning and the feature subsets. For feature subset, each feature is represented by a 1 or 0 for whether it is selected or not. **Figure 1** illustrates the mixed solution representation in the PSO.

P_1	…	P_N	f_1	…	f_n

Fig. 1 Solution representation

The variables needed to be optimized are enormous and mixed, since the costs or sampling ratios we intend to optimize are continuous while the feature selection is discrete. PSO was originally developed for continuous valued spaces. The discrete PSO [25] can solve the discrete variables. The major difference between the discrete PSO and the original version is that the velocities of the particles are rather defined in terms of probabilities that a bit will change to one. Using this definition a velocity must be restricted within the range [0, 1], to which all continuous values of velocity are mapped by a sigmoid function:

$$v'^t_i = sig(v_i^t) = \frac{1}{1+e^{-v_i^t}} \quad (3)$$

Equation 3 is used to update the velocity vector of the particle while the new position of the particle is obtained using **Equation 4**.

$$x_i^{t+1} = \begin{cases} 1 & if \quad r_i < v'^t_i \\ 0 & otherwise \end{cases} \quad (4)$$

where r_i is a uniform random number in the range [0,1] .

Evaluation measures play a crucial role in both assessing the classification performance and guiding the modeling of the classifier. For imbalanced datasets, the evaluation metric should take into account the imbalance. The average accuracy is not an appropriate evaluation metric. We chose G-mean and AUC as the evaluation metric. Different evaluations reflect different aspect of the classifiers. The AUC concerns the ranking ability more and the G-mean concerns the two accuracies of both classes at the same time.

The detailed algorithm about the measure oriented wrapper framework with PSO is shown in Algorithm 1. It is a wrapper framework for empirically discovering the potential parameters and feature subset. It applies 3-fold cross validation to evaluate classification performance for each potential solution of particles to avoid any estimation biases. The averaged performance measure is calculated as the fitness value of each solution in the particle.

Algorithm 1 *Measure optimized wrapper framework algorithm*

Input: dataset *D*; termination condition *T*; particle update parameters *SN*; metric *E; NumFolds*(3)
Randomly initialize particle population positions and velocities (including critical parameters of imbalanced data learning solutions, and feature subset)
repeat
  foreach particle *i*
    Construct the D_i with the feature selected by the particle *i* (x_i)
    for *k*=1 to *NumFolds*
      Separate D_i into Trt^k_i (80%) for training *and* Trv^k_i (20%) for validation randomly
      Train a classifier with parameters and intrinsic parameters optimized by the particle *i* on the Trt^k_i
      Evaluate the cost sensitive classifier on the Trv^k_i , and obtain the value
      M^k_i based on evaluation metric *E*
    end for
    M_i=average(M^k_i); Assign the fitness of particle *i* (x_i) with M_i ;
    if *fitness* ($pbest_i$) <= *fitness* (x_i) **then** $pbest_i = x_i$
  end foreach
  set *gbest* as best *pbest*
  foreach particle *i*
    update $velocity_i$ and $position_i$ with Eq. 1 - 4.
end foreach
until *T*
output ratio cost and feature subset of *gbest*

In this section, the measure-optimized wrapper framework is described. The purpose is to search for the potentially optimal parameters and features in the parameter space with the highest evaluation score guided by a heuristic optimization function. We choose un-pruned C4.5 decision tree with Laplace smoothing (DT) as our base classifier, since in the literature, it is the most commonly used base classifier in learning models with ensembles on data with class imbalance. It can also be used for multiclass classification without decomposition.

B. Measure optimized cost sensitive DT, MOCS-DT

In this work, we will make use of the cost-sensitive C4.5 decision tree (CS-DT) proposed in [27]. This method changes the class distribution such that the induced tree is in favor of the class with high cost and is less likely to commit errors with high cost. The standard greedy divide-and-conquer procedure for inducing minimum error trees can then be used without modification, except that *w(j)* is used instead of N_j (number of instances of class *j*) in the computation of the test selection criterion in the tree growing process.

$$w(j) = C(j)\frac{N}{\sum_i C(i)N_i} \quad (5)$$

such that the sum of all instance weights is $\sum_j w(j)N_j = N$.

The misclassification cost of class *C(j)* is:

$$C(j) = \sum_i^N cost(i, j) \quad (6)$$

where *cost(i, j)* is the cost of misclassifying a class *j* instance as belonging to class *i*, and *N* is the size of classes.

According to the principle of cost-sensitive learning

algorithm, the parameters of misclassification cost play a crucial role in the construction of a cost sensitive approach. Nevertheless, an important issue of applying the cost-sensitive learning algorithm to the imbalanced data is that the cost matrix is often unavailable for a problem domain. It is often not correct to set the cost ratio to the inverse of the imbalance size ratio (the number of majority instances divided by the number of minority instances). For binary class problems, empirical methods can be used by testing a range of ratios of cost values with a grid search. However, for a multi-class problem, empirical methods are not effective since the search space is expanded exponentially as the class number increases. The measure optimized wrapper framework can automatically determine the optimal cost as well as the feature subset during training of the CS-DT oriented by the imbalanced evaluation criteria

C. Measure optimized hybrid re-sampling DT, MOHS-DT

Weiss and Provost observed that the naturally occurring distribution is not always optimal [28]. Thus, the data distribution needs to be modified before being trained by the future classifier, conditioned on an evaluation function. It is desired to find the optimal class distributions for obtaining best performance. The purpose is to search for the multiple re-sampling levels for each class and feature subset in the parameter space with the highest evaluation score guided.

Many papers showed that a combination of under-sampling and SMOTE over-sampling offers an advantage over either in isolation. Hence, we choose the hybrid re-sampling method integrating the SMOTE and Random under-sampling. Chawla proposed a wrapper approach to determine the re-sampling ratio parameters of SMOTE over-sampling and random under-sampling (RUS) [16]. However, it aimed to optimize the parameters of the re-sampling percentages only without considering the feature selection. Moreover, it is limited on the binary classes.

In the multiple classes case, there are multiple parameters of re-sampling ratio needed to be searched. Moreover, the data distribution is so complex that it is difficult to determine the re-sampling type for all the classes except the largest and smallest classes. MOHS-DT can cope with these issues above. In the binary class case, the minority class is conducted by SMOTE over-sampling, and the majority class is conducted by RUS. In the case of multi-class, the largest class is conducted by RUS, and the smallest class is conducted by SMOTE over-sampling. As for the other classes, the re-sampling type can be also determined according to the value of the re-sampling ratio optimized automatically. If the re-sampling ratio is bigger than 1, the corresponding class will be conduct by SMOTE over-sampling RUS; otherwise it will be manipulated by RUS. The re-sampling amounts can be determined by the specific value in the solution. Hence, the re-sampling type and level of each class can be determined through optimizing of MOHS-DT.

Besides the re-sampling level, the feature set is important to the data distribution. Specially, SMOTE is related with the feature set because the mechanism of over-sampling in SMOTE is based on the feature space. The irrelevant or redundant features can also cause an over-sampling performance degradation. Furthermore, if the feature is too large, the data distribution is so sparse that the new instances are not accurate. Therefore, the re-sampling and the feature selection need to be conducted at the same time. The PSO searches the entire space of hybrid re-sampling level as well as the feature subset on multiple classes. Once the best level for hybrid re-sampling and best feature subset are found, they can be used to build a classifier on the updated training set and applied on the unseen testing set.

IV. EXPERIMENTAL STUDY

A. Dataset description

To evaluate the classification performance of our proposed methods in different classification tasks, and to compare with other methods specifically devised for imbalanced data, we tried several datasets from the UCI database. To evaluate our methods, we chose five binary dataset as well as ten multi-class datasets. The data information is summarized in **Table 1**. The datasets chosen have diversity in the number of attributes and imbalance ratio. Moreover, the datasets used have both continuous and categorical attributes.

Table 1. UCI datasets used.

Dataset	***C***	***Inst.***	***F.***	***Class distribution***
German	2	1000	24	300/700
Pima	2	768	8	268/500
Sick	2	3772	29	231/3541
Spambase	2	4601	57	1813/2788
Breast Cancer	2	699	9	458/241
Cmc	3	1473	9	629/333/511
Annealing	4	898	38	8/99/684/67/40
Balance	3	625	4	49/288/288
Car	4	1728	6	1210/384/69/65
Glass	6	214	9	70/76/17/13/9/29
Page	5	5473	10	4913/329/28/88/115
New-Thyroid	3	215	5	150/35/30
Nursery	4	12958	8	4320/328/4266/4044
Satimage	6	6435	36	1533/703/1358/626/707/1508
Yeast	10	1484	9	463/429/244/163/51/44/35/30/20/5

B. Experiment setup

For the PSO setting of our method, its initial parameter values in our proposed method were set according to the conclusion drawn in [29]. The parameters were used: C_1=2.8, C_2=1.3, w=0.5. For empirically providing good performance while at the same time keeping the time complexity feasible, particle number was set dynamically according to the amount of the variables optimized (=1.5×|variables to be optimized|), and the termination condition could be a certain number of iterations (500 cycles) or other convergence condition (no changes any more within 2× |variables to be optimized| cycles). In all our experiments, instead of the traditional 10-fold cross validation which can result in few instances of minority class, each dataset was randomly separated into training set (70%) for constructing classifiers and test sets (30%) for validating the classifiers. This procedure was

repeated 20 times for obtaining unbiased results. All of the classification algorithms and optimization methods were implemented or directly used in the Weka platform.

C. Experiment 1: how MOCS-DT improves the performance

In this experiment, we compare between the basic DT classifier with and without the feature selection, cost sensitive decision tree (CS-DT) with default setting, our proposed method MOCS-DT with and without the feature selection. For the basic DT with feature selection, it is a common wrapper feature selection method evaluated by classification performance. For the basic CS-DT, the misclassification cost for class C_i is set to $ImbaRatio_i$ which is the size ratio between the largest class and each class C_i. In the MOCS-DT, the range of misclassification cost for class C_i was empirically chosen in [1, 10×*ImbaRatio*]. The misclassification cost of largest class is set to 1.

In this experiment, we evaluate the overall quality and optimize the factors with the G-mean. The G-mean is the geometric mean of specificity and sensitivity, which is commonly utilized when performance of both classes is concerned and expected to be high simultaneously [26]. It is a good indicator of overall performance, and has been used by several researchers for evaluating classifiers on imbalanced datasets [8, 12, 14, 21]. G-mean is typically defined for binary classes but can be expanded to the scenario of multiple classes as the geometric mean of recall values of every class in [12]:

$$G-mean = (\prod_{i}^{N} R_i)^{1/N} \quad (7)$$

The average G-mean scores are shown in the **Table 2**. From the results in **Table 2**, we find that optimizing the cost generally helps the decision tree learn on the different data sets, regardless of feature selection or not. Under the condition where the feature selection is not carried out, we find that MOCS-DT with only optimizing the cost is always better than the basic CS-DT with the default costs. It means that the cost should be searched, so as to obtain an expected performance of cost sensitive learning.

Table 2. Experimental results of the MOCS-DT method with and without feature selection, as well as basic DT and CS-DT

Dataset	Basic DT		CS-DT	MOCS-DT	
	without *FS*	*FS*	without *FS*	without *FS*	*FS*
German	0.603	0.649	0.639	0.722	**0.751**
Pima	0.681	0.689	0.696	0.729	**0.748**
Sick	0.929	0.943	0.929	0.949	**0.961**
Spambase	0.911	0.919	0.924	0.956	**0.964**
Breast Cancer	0.949	0.951	0.933	0.967	**0.972**
Cmc	0.467	0.489	0.521	0.563	**0.569**
Annealing	0.890	0.911	0.895	0.907	**0.930**
Balance	0	0	0.554	0.557	**0.562**
Car	0.837	0.842	0.861	0.909	**0.925**
Glass	0.586	0.596	0.604	0.621	**0.627**
Page	0.836	0.862	0.822	0.897	**0.943**
New-Thyroid	0.874	0.878	0.877	**0.927**	0.924
Nursery	0.913	0.931	0.905	0.955	**0.989**
Satimage	0.756	0.815	0.809	0.864	**0.913**
Yeast	0	0	0.119	0.362	**0.383**

We also find the feature selection step to be important when working on the imbalanced data classification for both the basic DT and the MOCS-DT. With MOCS-DT, the use of feature selection is found to improve the G-mean for each dataset except the New-Thyroid dataset. Therefore, we can draw the conclusion that simultaneously optimizing the misclassification cost and feature subset guided by the imbalanced evaluation measure improves the classification performance of the CS-DT.

D. Experiment 2: how MOHS-DT improves the performance

In this experiment, we conduct the comparison between our MOHS-DT and basic single re-sampling as well as the wrapper-based hybrid re-sampling method (WBHS) proposed in [9]. For the WBHS method, we extend it to multi-class imbalanced data. The average number of instances of all the classes, N_{ave} is calculated as a base level. The re-sampling type of each class can be fixed according to the difference between the N_{ave} and its size. For those classes, of which the size is larger than the N_{ave}, we under-sample; and the remaining classes we use SMOTE over-sampling. The process of the WBHS proceeds in sequence from the largest class to the smallest class. For the WBHS and MOHS, the re-sampling increment step is set to 10% and 50% for RUS and SMOTE over-sampling, respectively. In the MOHS, to avoid missing the potential solution, we set the range of re-sampling ratio larger. The range of re-sampling parameter of class C_i are set to [0.1×$Rsize^S{}_i$×100%, 10×$Rsize^L{}_i$×100%] where the $Rsize^L{}_i$ is the ratio between the largest class and the class C_i, and $Rsize^S{}_i$ is the ratio between the smallest class and the class C_i. For SMOTE, the nearest neighbor parameter K is set to 5, and the classes which are smaller than N_{ave} need to be over-sampled until the average level N_{ave}. For the RUS, the classes which are larger than N_{ave} need to be under-sampled until the average level N_{ave}

Table 3. Experimental results of the MOHS-DT method with and without feature selection, as well as single basic re-sampling method and WBHS

Dataset	Basic re-sampling		WBHS	MOHS-DT	
	SMOTE	RUS	SMOTE & RUS	*Without FS*	*FS*
German	0.629	0.611	0.673	0.682	0.734
Pima	0.733	0.697	0.753	0.745	0.759
Sick	0.947	0.902	0.902	0.938	0.952
Spambase	0.937	0.929	0.942	0.944	0.957
Breast Cancer	0.969	0.943	0.971	0.962	0.976
Cmc	0.453	0.442	0.498	0.507	0.532
Annealing	0.895	0.869	0.902	0.911	0.919
Balance	0	0	0.528	0.574	0.574
Car	0.871	0.733	0.876	0.901	0.927
Glass	0.603	0.555	0.604	0.602	0.608
Page	0.843	0.624	0.871	0.913	0.927
New-Thyroid	0.917	0.896	0.933	0.921	0.928
Nursery	0.968	0.852	0.977	0.963	0.966
Satimage	0.844	0.622	0.861	0.869	0.918
Yeast	0.221	0.103	0.259	0.312	0.341

, We find in **Table 3** that both WBHS and MOHS improve the common re-sampling performance. MOHS without feature selection beats WBHS on 10 of the 15 datasets.

WBHS lacks many potential parameters pairs from the parameter space, while MOHS can find better potential re-sampling ratio pairs. In addition, the feature selection is as important as the re-sampling; it cannot be ignored when re-balancing the data distribution.

E. Experiment 3: MOCS-DT and MOHS-DT vs. the state-of the-art methods

Through both experiments above, the proposed framework improves the method with default setup, although it cannot ensure the solution is the best one in the multi-variable space due to the nature of PSO.

In this experiment, our methods are compared with the other state-of-the-art imbalanced data methods from the data level (algorithms that change data distributions) and classifier level (algorithms that address data imbalance with the classifier). The algorithms from the data level in the comparison include: Edited Nearest Neighbor (ENN) Rule under-sampling [17], and SMOTEBoosting over-sampling [4]. ENN under-sampling removes the largest class examples whose class label differ from the class of at least two of its three nearest neighbors. SMOTEBoost is an over-sampling technique based on a combination of the SMOTE algorithm and the boosting procedure. For SMOTEBoost algorithm, the nearest neighbor parameter K is set to 5, and the classes which are smaller than N_{ave} are over-sampled until the average level N_{ave}. The methods from the classifier level are Metacost [7], Ada.NC combined with random over-sampling [21] and AdaC.M1 [12]. For Metacost, the misclassification cost of class C_i is set to $ImbaRatio_i$. In the setup of AdaC.M1, the chosen fitness function is G-mean. For Ada.NC, the penalty strength parameter is set 9. The sizes of components are 50 in the all three ensemble classifiers.

For assessing the overall performance evaluating these methods with more information, we also use AUC (area under the curve) to test the performance. It is found that AUC is statistically consistent with accuracy and statistically more discriminant than accuracy [30]. Moreover, it is used as evaluation metric on the imbalanced data classification [5, 21]. Hand and Till [13] proposed an AUC for multi-classification problems (MAUC) defined as:

$$MAUC = \frac{2}{M \times (M-1)} \sum_{i<j} \frac{A_{ij} + A_{ji}}{2} \tag{8}$$

where M is the number of classes. A_{ij} is the AUC value calculated from the i-th column of the output matrix considering instances from class i and j. The output matrix is the output of classifier for data set can be arranged as a $N \times M$ matrix. Each row of the matrix is a vector indicating the confidence that instance x_i, $i \in \{1, \ldots, N\}$ belongs to the corresponding class. The experiment results are shown in **Table 4**. As shown in bold in **Table 4**, our MOCS-DT and MOHS-DT outperform all the other approaches on the great majority of datasets. From the average rank on the 15 dataset, MOCS-DT and MOHS-DT are first two best classifiers. Our methods conduct the feature selection in the wrapper paradigm, hence they improve the classification performance remarkably on the data sets which have higher dimensionality, such as Satimage, Spambase and Annealing.

From the results, we find that the methods are also effective in improving MAUC, regardless of binary classes or multiple classes. To better understand the results of our techniques when compared to the other classification approaches, we perform a statistical analysis of our results [31]. Firstly, the Friedman test is used to determine that there is a statistically significant difference between the rankings of the classifiers in terms of G-mean and MAUC. Thus we reject the null-hypothesis. Next, the Bonferroni-Dunn test is applied to compare each classifier against the control classifier. The results of the Bonferroni-Dunn can be seen in **Table 5**.

From the results of this test, it can be concluded that MOCS-DT obtains a significantly better result than ENN, Metacost, Ada.NC and SMOTEBoost at the 95% confidence level in terms of G-mean and MAUC. While MOHS-DT can performs statistically significantly better than ENN and Metacost at the 95% confidence level in terms of G-mean and MAUC. It can also perform statistically significantly better than Ada.NC at the 95% confidence level in terms of MAUC. When the confidence level is 90% MOHS-DT can perform statistically significantly better than SMOTEBoost in terms of MAUC. Although there is no sufficient evidence to prove the statistical significance between the two measure optimized methods and AdaC.M1, our methods can obtain fewer and more effective feature set for predicting future instances.

F. Experiment 4: how the performance of the measure optimized wrapper with MAUC used as the objective function

In all the experiments above, the measure optimized methods only utilize G-mean as objective function. As we known, the AUC has been used to enhance the quality of binary classifiers [12, 16]. Maximizing the volume under the ROC surface can improve the performance in multi-class classification [32]. In this experiment, we analyze the behavior of both measure-optimized methods with MAUC as objective function.

We compared the performances based on these different evaluation objective functions in **Table 6**. In the majority of cases, the G-mean value from the G-mean wrapper is higher than the one of the AUC wrapper, but in some cases, the average G-mean from AUC wrapper is better than the one from G-mean wrapper, such as German, Cmc and Glass datasets. From this, we believe it results in more generalized performances when using AUC as the wrapper evaluation function, which is a similar conclusion as the one drawn in paper [16], where the result of F-measure on some data sets when using AUC as the wrapper evaluation metric is better than the one with F-measure as the metric of the wrapper. We believe that employing the AUC evaluation measure as optimization objective could lead to more generalized performances. Moreover, the two evaluation metrics wrapper optimizations for the same classifier result in different misclassification cost or re-sampling ratio and feature subset, since they optimize different properties of the classifier.

Table 4. Average values of G-mean and MAUC for multiple methods on the data sets

Dataset	Metric	ENN	MC	SMB	Ada.NC	AdaC2.M1	MOCS-DT	MOHS-DT
German	G-mean	0.641	0.635	0.652	0.695	0.732	**0.751**	0.734
	MAUC	0.689	0.655	0.693	0.695	0.702	**0.744**	0.728
Pima	G-mean	0.741	0.738	0.744	0.741	0.752	0.748	**0.759**
	MAUC	0.821	0.846	0.867	0.863	0.884	**0.887**	**0.887**
Sick	G-mean	0.934	0.940	**0.972**	0.951	0.954	0.961	0.952
	MAUC	0.947	0.966	**0.982**	0.975	0.969	0.969	0.974
Spambase	G-mean	0.925	0.929	0.931	0.939	0.939	**0.964**	0.957
	MAUC	0.922	0.948	0.964	0.965	0.977	0.989	**0.992**
Breast Cancer	G-mean	0.942	0.966	0.970	0.972	0.964	0.972	**0.976**
	MAUC	0.951	0.951	0.969	0.962	0.976	**0.985**	0.976
Cmc	G-mean	0.417	0.478	0.489	0.496	0.517	**0.569**	0.532
	MAUC	0.717	0.748	0.746	0.739	0.752	**0.776**	0.755
Annealing	G-mean	0.878	0.885	0.914	0.898	**0.939**	0.930	0.919
	MAUC	0.946	0.967	0.988	0.987	**0.993**	0.988	0.990
Balance	G-mean	0	0.559	0	0.370	0.566	0.562	**0.574**
	MAUC	0.616	0.689	0.711	0.707	**0.744**	0.722	0.739
Car	G-mean	0.879	0.921	0.914	0.928	**0.944**	0.925	0.927
	MAUC	0.964	0.983	0.992	0.987	0.995	**0.996**	**0.996**
Glass	G-mean	0.595	0.609	0.611	0.597	0.619	**0.627**	0.608
	MAUC	0.891	0.889	0.919	0.876	0.922	**0.931**	0.927
Page	G-mean	0.684	0.817	0.845	0.926	0.938	**0.943**	0.927
	MAUC	0.928	0.955	0.987	0.988	0.988	0.989	**0.995**
New-Thyroid	G-mean	0.884	0.905	**0.955**	0.919	0.921	0.924	0.928
	MAUC	0.972	0.986	0.987	0.982	0.987	0.987	**0.989**
Nursery	G-mean	0.898	0.962	0.952	0.963	0.952	0.989	0.966
	MAUC	0.979	0.996	0.995	0.996	**0.999**	**0.999**	0.998
Satimage	G-mean	0.845	0.842	0.891	0.885	0.882	0.913	**0.918**
	MAUC	0.924	0.974	0.990	0.989	0.989	**0.995**	0.992
Yeast	G-mean	0	0.161	0.254	0.242	0.354	**0.383**	0.341
	MAUC	0.788	0.829	0.832	0.835	**0.852**	0.839	0.835

Table 5. The differences of rankings of the Bonferroni-Dunn test at various confidence levels

	ENN	MC	SMB	Ada.NC	AdaC2.M1	MOHS-DT	control
G-mean	4.9*	3.833*	2.367*	2.4*	0.933	0.667	MOCS-DT
	4.23*	3.167*	1.7	1.733	0.267	-	MOHS-DT
	ENN	MC	SMB	Ada.NC	AdaC2.M1	MOHS-DT	control
MAUC	4.83*	3.867*	2.167*	2.533*	0.7	0.167	MOCS-DT
	4.667*	3.7*	$2^{\#}$	2.367*	0.533	-	MOHS-DT

Bonferroni-Dunn test: $CD_{(\alpha=0.05)}$=2.08, $CD_{(\alpha=0.1)}$=1.884 *,#: Statistically difference with α =0.05(*) and α= 0.1(#)

Table 6. The Experimental results (G-mean, MAUC and feature size) of MOCS-DT and MOHS-DT driven by G-mean and MAUC separately

Dataset	MOCSL						MOHS-DT					
	MOCS-DT_{GM}			MOCS-DT_{MAUC}			MOHS-DT_{GM}			MOHS-DT_{MAUC}		
	G-mean	MAUC	Fea.	G-mean	MAUC	Fea.	G-mean	MAUC	Fea.	G-mean	MAUC	Fea.
German	0.751	0.744	16	0.772	0.753	15	0.734	0.728	14	0.738	0.749	15
Pima	0.748	0.887	6	0.742	0.891	4	0.759	0.887	5	0.739	0.884	5
Sick	0.961	0.969	19	0.949	0.975	17	0.952	0.974	16	0.946	0.974	16
Spambase	0.964	0.989	23	0.950	0.993	21	0.957	0.992	21	0.932	0.994	22
Breast Cancer	0.972	0.985	6	0.975	0.988	6	0.976	0.976	6	0.961	0.989	6
Cmc	0.569	0.776	8	0.558	0.777	7	0.532	0.755	8	0.539	0.776	5
Annealing	0.930	0.988	24	0.918	0.994	26	0.919	0.990	19	0.911	0.992	21
Balance	0.562	0.722	3	0.557	0.761	3	0.574	0.739	3	0.468	0.744	3
Car	0.925	0.996	4	0.923	0.996	5	0.927	0.996	4	0.952	0.997	4
Glass	0.627	0.931	6	0.633	0.938	7	0.608	0.927	8	0.611	0.945	7
Page	0.943	0.989	5	0.932	0.996	6	0.927	0.995	7	0.951	0.996	7
New-Thyroid	0.924	0.987	4	0.927	0.991	4	0.928	0.989	4	0.927	0.996	3
Nursery	0.989	0.999	5	0.988	0.999	5	0.966	0.998	5	0.962	0.998	5
Satimage	0.913	0.995	18	0.917	0.997	19	0.918	0.992	20	0.899	0.996	19
Yeast	0.383	0.839	5	0.356	0.844	6	0.341	0.835	7	0.329	0.848	6

I. Conclusion

Learning with class imbalance is a challenging task, and the effects of imbalance on the multiple classes are even more problematic. The paper studies empirically the effect of the measure optimized framework to deal with the multiclass imbalanced data learning. We investigate the framework in two different perspectives: cost sensitive learning and re-sampling technique. The framework can discover the optimal factors based on objective functions like the G-mean and MAUC, so as to improve the performance. The experimental results presented in this study confirm the advantages of our approaches over state-of-the-art methods designed for addressing the imbalance datasets, showing the promising perspective and new understanding of cost sensitive learning and re-sampling methods. Therefore, the important finding in this study suggests that we need to optimize the parameters of the method when confronted with an imbalanced dataset.

The setup of optimized parameters is specific not only to the given data, but also to the learning objective and the base classifier. The kind of objective function can be chosen based on the training objective of the given problem; the alternative performance measures such as F-measure can also be incorporated. In this study, we only demonstrated its applicability with decision tree which is commonly used in the imbalanced data learning. However, our measure optimized framework can be applied on other classifiers. In future research, we will extend and investigate the performance of the framework based on other classifiers..

Acknowledgment

This work is supported by the Alberta Innovates Centre for Machine Learning and one author was supported by the China Scholarship Council for two years at the University of Alberta.

References

[1] H. He, E. Garcia, Learning from imbalanced data, Knowledge and Data Engineering, IEEE Transactions on 21,2009, pp.1263-1284.

[2] N.V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets 2004, 6 (1):1-6

[3] N.V. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, 2002, pp. 341-378.

[4] N.V. Chawla, A. Lazarevic, L. Hall, K. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, Knowledge Discovery in Databases: PKDD 2003, 2003, pp. 107-119.

[5] Chen, H. He, E. Garcia, RAMOBoost: Ranked minority oversampling in boosting, Neural Networks, IEEE Transactions on 21, 2010, pp. 1624-1642.

[6] Z. Zhou, X. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, Knowledge and Data Engineering, IEEE Transactions on 18, 2006, pp.63-77.

[7] P. Domingos, Metacost: a general method for making classifiers cost-sensitive, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 155-164, 1999.

[8] B. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, Foundations of Intelligent Systems, 2008, pp. 38-47.

[9] Z. Zheng, X. Wu, R. Srihari, Feature selection for text categorization on imbalanced data, ACM SIGKDD Explorations 6, 2004, pp.80-89.

[10] L. Lusa, et al., Class prediction for high-dimensional class-imbalanced data, BMC bioinformatics 11, 2010.

[11] Y. Sun, M. Kamel, Y. Wang, Boosting for learning multiple classes with imbalanced class distribution, in ICDM'06, 2006, pp. 592-602.

[12] D. J. Hand & R.J. Till. A simple generalization of the area under the ROC curve for multiple class classification problems. Machine Learning, 2001, 45(2), 171-186.

[13] Y. Tang, Y.Q. Zhang, N. V., Chawla, & S. Krasser. SVMs modeling for highly imbalanced classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39(1), 281-288."SVMs modeling for highly imbalanced classification. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39.1, 2009, pp. 281-288.

[14] N. Thai-Nghe, Z. Gantner, & L. Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2010.

[15] B. Yuan, and X.L. Ma. Sampling + Reweighting: Boosting the Performance of AdaBoost on Imbalanced Datasets. In Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), pp. 2680-2685, 2012.

[16] N.V. Chawla, D. Cieslak, L. Hall, A. Joshi, Automatically countering imbalance and its empirical relationship to cost, Data Mining and Knowledge Discovery 17, 2008, pp. 225-252.

[17] D.L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics, 1972, (3): 408-421.

[18] H. Yu, J. Ni, J. Zhao, Acosampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, Neurocomputing 101, pp.309-318, 2013.

[19] T. Hoens, N. Chawla, Generating diverse ensembles to counter the problem of class imbalance, Advances in Knowledge Discovery and Data Mining, pp. 488-499, 2010.

[20] J. Van Hulse, T. Khoshgoftaar, A. Napolitano, R. Wald, Feature selection with high-dimensional imbalanced data, in: Data Mining Workshops, 2009. ICDMW'09, pp.507-514.

[21] S. Wang, X. Yao, Multiclass imbalance problems: Analysis and potential solutions, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42, 2012, pp. 1119-1130.

[22] G. Ou, Y. Murphey, Multi-class pattern classification using neural networks, Pattern Recognition 40, 2007, pp. 4-18.

[23] J. Kennedy, R. Eberhart, Particle swarm optimization, .IEEE International Conference on Neural Networks, 1995, volume 4, pp. 1942-1948.

[24] D. Martens, B. Baesens, T. Fawcett, Editorial survey: swarm intelligence for data mining, Machine Learning 82, 2011, pp. 1-42.

[25] M. Khanesar, M. Teshnehlab, M. Shoorehdeli, A novel binary parti-cle swarm optimization, in: Control & Automation, 2007. MED'07.

[26] M. Kubat, S. Matwin, et al., Addressing the curse of imbalanced training sets: one-sided selection, Proc. Int'l Conf. Machine Learning, 1997, pp. 179-186.

[27] K.M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Transactions on Knowledge and Data Engineering, vol.14, no.3, pp.659–665, 2002.

[28] G. Weiss, F. Provost, Learning when training data are costly: The effect of class distribution on tree induction, J. Artif. Intell. Res. (JAIR) 19, 2003, pp. 315-354.

[29] A. Carlisle and G. Dozier, An Off-The-Shelf PSO, Proceedings of the Workshop on Particle Swarm Optimization, 2001, pp. 1-6.

[30] C. X. Ling, J. Huang & H. Zhang, AUC: a statistically consistent and more discriminating measure than accuracy. In International Joint Conference on Artificial Intelligence, 2003, Vol. 18, pp. 519-526.

[31] J. Demsar. Statistical comparisons of classifiers over multiple datasets. Journal of Machine Learning Research, 2006, 7 (1):1-30.

[32] K. Tang, R. Wang and T. Chen, Towards maximizing the area under the ROC Curve for multi-class classification problems, *In proceedings of the 25th AAAI Conference on Artificial Intelligence,* 2011, pp. 483-488.