Fourth International Conference on Natural Computation

On the Class Imbalance Problem[*]

Xinjian Guo, Yilong Yin[1], Cailing Dong, Gongping Yang, Guangtong Zhou
School of Computer Science and Technology, Shandong University, Jinan, 250101, China
E-mail: xinjianguo@mail.sdu.edu.cn, ylyin@sdu.edu.cn

Abstract

The class imbalance problem has been recognized in many practical domains and a hot topic of machine learning in recent years. In such a problem, almost all the examples are labeled as one class, while far fewer examples are labeled as the other class, usually the more important class. In this case, standard machine learning algorithms tend to be overwhelmed by the majority class and ignore the minority class since traditional classifiers seeking an accurate performance over a full range of instances. This paper reviewed academic activities special for the class imbalance problem firstly. Then investigated various remedies in four different levels according to learning phases. Following surveying evaluation metrics and some other related factors, this paper showed some future directions at last.

1. Introduction

Many traditional algorithms to machine learning and data mining problems assume that the target classes share similar prior probabilities. However, in many real world applications, such as oil-spill detection, network intrusion detection, fraud detection, this assumption is grossly violated. In such problems, almost all the examples are labeled as one class, while far fewer examples are labeled as the other class, usually the more important class. This situation is known as the problem of class imbalance. In this case, standard classifiers tend to be overwhelmed by the majority class and ignore the minority class. Its importance grew as more and more researchers realized that this imbalance causes suboptimal classification performance, and that most algorithms behave badly when the data sets are highly imbalanced. The class imbalance problem has been a hot topic in machine learning in recent years.

Class imbalance problem has been recognized to be existing in lots of application domains, such as spotting unreliable telecommunication customers, detection of oil spills in satellite radar images, learning word pronunciations, text classification, risk management, information retrieval and filtering tasks, medical diagnosis (e.g. rare disease and rare genes mutations), network monitoring and intrusion detection, fraud detection, shuttle system failure, earthquakes and nuclear explosions and helicopter gear-box fault monitoring. From the view of applications, the nature of the imbalance falls in two cases: The data are naturally imbalanced (e.g. credit card frauds and rare disease) or, the data are not naturally imbalanced but it is too expensive to obtain data of the minority class (e.g. shuttle failure) for learning.

There have been lots of researches on class imbalance problem. Paper [1] reviewed various techniques for handling imbalance dataset problems. Paper [2] traced some of the recent progress in the field of learning from imbalanced data sets, in which Sofia Visa et. al. argued that the poor performance of the classifiers produced by the standard machine learning algorithms on imbalanced data sets is mainly due to the following three factors: accuracy, class distribution and error costs, since they are rarely well satisfied in real world applications. Paper [3] discussed several issues related to learning with skewed class distributions, such as the relationship between cost-sensitive learning and class distributions, and the limitations of accuracy and error rate to measure the performance of classifiers. Weiss [4] presented an overview of the field of learning from imbalanced data. He pays particular attention to differences and similarities between the problems of rare classes and rare cases. He then discussed some of the common issues and their range of solutions in mining imbalanced data sets.

The reminder of the paper is organized as follows. Section 2 reviewed academic activities including two

[*] This paper concentrates on binary classification problem, and by convention, the class label of the minority class is positive, and that of the majority class is negative.

[1] Corresponding author Yilong Yin is with School of Computer Science and Technology, Shandong University. E-mail: ylyin@sdu.edu.cn

978-0-7695-3304-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNC.2008.871

IEEE computer society

workshops and one special issue on the problem of class imbalance. Sections 3 surveyed the remedies to the class imbalance problem from four different levels. Popular evaluation metrics for imbalanced data sets were summarized in section 4. Section 5 briefly analyzed some other factors related to the class imbalance problem, and section 6 concluded the paper and showed some future directions.

2. Academic activities on the class imbalance problem

As described above, recognizing class imbalance problem exists in extensive application domains gave rise to two workshops held at the top conferences in AI, and one special issue on dealing with the class imbalance problem.

The first workshop dedicated to the class imbalance problem was held in conjunction with the American association for artificial intelligence conference 2000 (AAAI'2000). Its main contribution includes observation of many application domains dealing with imbalanced data sets, and several important issues, such as how to evaluate learning algorithms, what evaluation measures should be used, one class learning versus discriminating methods, discussions over various re-sampling methods, discussion of the relation between class imbalance problem and cost-sensitive learning, the goal of creating classifiers that performs well across a range of costs and so on.

The second workshop special for the class imbalance problem is part of the international conference on machine learning 2003 (ICML'2003), in which most research on the problem was guided by the first workshop. For example, ROC or cost curves were used as evaluation metrics, rather than accuracy. The workshop was followed by an interesting and vivid panel discussion. Two major directions presented in the research papers of the workshop: Many papers still reported various tuning methods applied to decision trees in order to perform better on imbalanced data sets, even though presentations in the previous workshop showed their shortcomings, and it was commonly agreed that new classifiers are needed for imbalanced data sets; Besides, re-sampling, under various aspects, was present in half of the papers and was the most debated issue, even though [5] shows that sampling has the same result as moving the decision threshold or adjusting the cost matrix (a result known since 1984 [6]). In addition, N. Japkowicz questioned the fact that the within class imbalance is responsible for the problem [7]. The idea is that within class imbalance leads to a severe lack of representation of some important aspects of the minority class.

The sixth issue of SIGKDD Exploration was dedicated entirely to the class imbalance problem, in which Weiss [4] presented a very good review of the current research on learning from imbalanced data sets, and the other papers in the volume address mainly issues of sampling, feature selection and one-class learning, for example, [9] investigated a boosting method combined with various over-sampling techniques of the hard to classify examples. The method improves the prediction accuracy for both the classes and does not sacrifice one class for the other with experiments on 17 data sets.

3. Remedies for the class imbalance problem

The remedies to deal with the problem of class imbalance are of four different levels according to the phases in learning, i.e. changing class distributions mainly by re-sampling techniques, features selection in the feature level, classifiers level by manipulating classifiers internally and ensemble learning for final classification.

3.1. Changing class distributions

Since examples belong to the minority class are far fewer than those belong to the majority class in situations of the class imbalance problem, one direct way to counter the problem is to change class distributions. Balanced distributions can be obtained by under-sampling the majority class, over-sampling minority class, combining the both and some other advanced sampling ways. There are numerous researches on changing class distributions [10-16]. Weiss investigated the effect of class distributions on decision tree by altering class distributions in several ratios with accuracy and AUC as metrics in his doctoral dissertation [10]. All the methods falls into three basic techniques: heuristic and non-heuristic under-sampling, heuristic and non-heuristic over-sampling and advanced sampling. In- [11]- various- strategies- for- learning- from- imbalanced- data- sets- were- compared,- and- it- concluded- that- under-sampling- and- over-sampling- are- very- effective- methods- for- dealing- with- the- class- imbalance- problem.-

3.1.1. Under-sampling. The most naivest under-sampling method is random under-sampling [12], a non-heuristic method trying to balance class distributions through the random elimination of majority class examples. This leads to discarding potentially useful data that could be important for classifiers.

There have been several heuristic under-sampling methods proposed or introduced from data cleaning in recent years. They are based on either of two different noise model hypotheses. One thinks examples that are near to the classification boundary of the two classes are noise, while the other considers examples with more neighbors of different labels are noise.

Condensed Nearest Neighbor Rule (CNN) [13] bases on the notion of a consistent subset of a sample set, which is a subset who can correctly classifies all of the remaining examples in the training set when used as a stored reference set for the NN rule. If the Bayesian risk is small, i.e., if the underlying densities of the various classes have small overlapping, then the algorithm will tend to pick out examples near the (perhaps fuzzy) boundary between the classes. Typically, points deeply imbedded within a class will not be transferred to "STORE", since they will be correctly classified. If the Bayesian risk is high, then "STORE" will contain essentially all the examples in the original training set, and no important reduction in training size will have been achieved. So CNN is effective only binary classes are of small overlapping.

OSS [14] randomly draws one majority class example and all examples from the minority class and then puts these examples in E′. Afterwards, use a 1-NN over the examples in E′ to classify the examples in E. Every misclassified example from E is moved to E′. The idea behind this implementation of a consistent subset is to eliminate the examples from the majority class that are distant from the decision border, since these examples might be considered less relevant for learning.

Wilson's Edited Nearest Neighbor Rule (ENN) [26] removes any example whose class label differs from the class of at least two of its three nearest neighbors.

Different from ENN, Neighborhood Cleaning Rule (NCL) [8] deals with majority and minority samples separately when cleaning the data sets. NCL uses ENN to remove majority examples, for each example E_i in the training set, its three nearest neighbors are found. If E_i belongs to the majority class and the classification given by its three nearest neighbors contradicts the original class of E_i, then E_i is removed. If E_i belongs to the minority class and its three nearest neighbors misclassify E_i, then the nearest neighbors that belong to the majority class are removed.

Compared with above four under-sampling methods, Tomek links [41] consider samples near the borderline should be paid more attention. Given two examples E_i and E_j belonging to different classes, and $d(E_i, E_j)$ is the distance between E_i and E_j; a (E_i, E_j) pair is called a Tomek link if there is not an example E_l, such that $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j , E_l) < d(E_i, E_j)$. If two examples form a Tomek link, then either one of these examples is noise or both examples form borderline. So, Tomek link can be viewed as an under-sampling method when examples of both classes are removed.

It should be noted that Tomek link, ENN and NCL are highly time-consuming, since for any example in the data sets, nearest neighbors of the sample must be found, so it is impossible for large datasets.

3.1.2. Over-sampling. Random over-sampling is a non-heuristic method that aims to balance class distributions through the random replication of minority class examples. Random over-sampling has two shortcomings. First, it will increase the likelihood of occurring over-fitting, since it makes exact copies of the minority class examples [14, 15]. Second, over-sampling makes learning process more time-consuming if the original data set is already fairly large but imbalanced.

There are several heuristic over-sampling methods mainly based on SMOTE. SMOTE generates synthetic minority examples to over-sample the minority class [15]. Its main idea is to form new minority class examples by interpolating between several minority class examples that lie together. By interpolating instead of replication, SMOTE avoids the over-fitting problem and causes the decision boundaries for the minority class to spread further into the majority class space.

Recognizing examples near the borderline of the classes are more important and more easily misclassified than those far from the borderline, Borderline_SMOTE [16] was proposed. It only over-sample the borderline examples of the minority class, while SMOTE and random over-sampling augment the minority class through all the examples from the minority class or a random subset of the minority class. For the minority class, experiments show that their approaches achieve better TP rate and F-value than SMOTE and random over-sampling methods.

3.1.3. Advanced sampling. Different from various under-sampling and over-sampling methods above, the following advanced sampling methods do re-sampling based on the results of preliminary classifications.

Boosting is an iterative algorithm that place different weights on the training distributions each iteration. After each iteration boosting increases the weights associated with the incorrectly classified examples and decreases the weights associated with the correctly classified examples separately. This forces the learner to focus more on the incorrectly classified examples in the next iteration. Note that boosting effectively alters the distributions of the training data, it can be considered to be a type of advanced sampling technique.

Weiss proposed a heuristic, budget-sensitive, progressive-sampling algorithm [10, 17] for selecting training data that approximates optimum. The budget-sensitive sampling strategy makes two additional assumptions. First, it assumes that the number of potentially available training examples from each class is sufficiently large so that a training set with *n* examples can be formed with any desired marginal class distributions. The second assumption is that the cost of executing the learning algorithm is negligible compared to the cost of procuring examples. This assumption permits the learning algorithm to be run multiple times, in order to provide guidance about which examples to select. He argued that though the heuristically determined class distributions associated with the final training set is not guaranteed to yield the best-performing classifier, the classifier induced using this class distributions performs well in practice.

Han et. al. proposed an over-sampling algorithm based on preliminary classification (OSPC) [18]. Firstly, preliminary classification was made on the test data in order to save the useful information of the majority class as much as possible. Then the test data that were predicted to belong to minority class were reclassified to improve the classification performance of the minority class. OSPC was argued to perform better than under-sampling methods and SMOTE in terms of the classification performance of the minority class and majority class.

It should be noted that all the methods of changing class distributions above are trying to deal with the problem of between-class imbalance. A cluster-based over-sampling [19] proposed to improve accuracy of minority class by dealing with the problems of between and within class imbalance simultaneously.

When the data sets are severely skewed, under-sampling and over-sampling methods are often combined to improve generalization of the learner [8, 12,16,20]. Batista et. al. [20] presented a comparison (and combination) of various sampling strategies. They noted that combining focused over-sampling and under-sampling, such as SMOTE combining with Tomek link or SMOTE combining with ENN is applicable when the data sets are highly imbalanced or there are very few examples of the minority class.

3.2. Feature selection

The majority of work in feature selection for imbalanced data sets has focused on text classification or Web categorization domain [21][22]. A couple of papers in this issue look at feature selection in situations of imbalanced data sets, albeit in text classification or Web categorization.

Zheng et. al. [23] suggested that existing measures used for feature selection are not very appropriate for imbalanced data sets. They proposed a feature selection framework, which selects features for positive and negative classes separately and then explicitly combines them. The authors showed simple ways of converting existing measures so that they separately consider features for negative and positive classes. Castillo and Serrano [24] did not particularly focus on feature selection, but made it a part of their complete framework.

Putten and Someren [25] analyzed the COIL 2000 data sets using the bias-variance decomposition and they reported that the key issue for this particular data set was avoiding over-fitting. They concluded that feature selection in such domains is even more important than the choice of the learning method.

3.3. Classifiers level

3.3.1. Manipulating classifiers internally. Drummond and Holte [27] reported that when using C4.5's default settings, over-sampling is surprisingly ineffective, often producing little or no change in performance in response to modifications of misclassification costs and class distributions. Moreover, they noted that over-sampling prunes less and therefore generalizes less than under-sampling, and that a modification of the C4.5's parameter settings to increase the influence of pruning and other over-fitting avoidance factors can reestablish the performance of over-sampling.

Some classifiers, such as the Naive Bayes classifier or some Neural Networks, yield a score that represents the degree to which an example is a member of a class. Such ranking can be used to produce several classifiers, by varying the threshold of an example pertaining to a class [4].

For internally biasing the discrimination procedure, a weighted distance function was proposed in [28] to be used in the classification phase of kNN. The basic idea behind this weighted distance is to compensate for the imbalance in the training sample without actually altering the class distributions. Thus, weights are assigned, unlike in the usual weighted k-NN rule, to the respective classes and not to the individual prototypes. In this way, since the weighting factor is greater for the majority class than for the minority one, the distance to positive minority class prototypes becomes much lower than the distance to prototypes of the majority class. This produces a tendency for the new patterns to find their nearest neighbor among the prototypes of the minority class.

Another approach to dealing with imbalanced datasets using SVM biases the algorithm so that the academic hyper-plane is further away from the positive class. This is done in order to compensate for the skew associated with imbalanced datasets which pushes the hyper-plane closer to the positive class. This biasing can be accomplished in various ways. In [29] an algorithm is proposed by changing the kernel function to develop this bias. Veropoulos et. al. [30] suggested using different penalty constants for different classes of data, making errors on positive examples costlier than errors on negative examples.

Kaizhu Huang et al. [31] presented Biased Minimax Probability Machine (BMPM) to resolve the imbalance problem. Given the reliable mean and covariance matrices of the majority and minority classes, BMPM can derive the decision hyper-plane by adjusting the lower bound of the real accuracy of the testing set.

3.3.2. Cost-sensitive learning. Besides changing the class distributions, incorporating costs in decision-making is another way to improve classifier's performance when learning from imbalanced datasets. Cost model takes the form of a cost matrix, as shown in Fig.1, where the cost of classifying a sample from a true class j to class i corresponds to the matrix entry *λij*. This matrix is usually expressed in terms of average misclassification costs for the problem. The diagonal elements are usually set to zero, meaning correct classification has no cost. The goal in cost-sensitive classification is to minimize the cost of misclassification, which can be realized by choosing the class with the minimum conditional risk.

		Prediction	
		Class i	Class j
True	Class i	0	λ_{ij}
	Class j	λ_{ji}	0

Fig. 1 Cost matrix

MetaCost [32] is another method to make a classifier cost-sensitive. The procedure begins to learn an internal cost-sensitive model by applying a cost-sensitive procedure, which employs a base learning algorithm. Then, MetaCost procedure estimates class probabilities using bagging and then re-labels the training examples with their minimum expected cost classes, and finally relearns a model using the modified training set.

AdaBoost's weight-update rule has been made cost-sensitive, so that examples belonging to rare class that are misclassified are assigned higher weights than those belonging to common class. The resulting system, Adacost has been empirically shown to produce lower cumulative misclassification costs than AdaBoost [33].

3.3.3. One-class learning. One-class learning is a recognition-based approach, which provides an alternative to discrimination where the model can be created based on the examples of the target class alone. Here, classification is accomplished by imposing a threshold on the similarity value [34] between a query object and the target class. Mainly, two classes of learners were previously studied in the context of the recognition-based one-class approach: SVMs [35][37] and auto-encoders [34][37], and they were found to be competitive [37].

Besides, systems that learn only the minority class may still train using examples belonging to all classes. Brute [38], Shrink [39] and Ripper [40] are three such data mining systems. Brute has been used to look for flaws in the Boeing manufacturing process [38]. Shrink uses a similar approach to detect rare oil spills from satellite radar images [39]. Based on the assumption that there will be many more negative examples than positive examples, Shrink labels mixed regions (i.e., regions with positive and negative examples) with the positive class. Ripper [40] is a rule induction system that utilizes a separate-and-conquer approach to iteratively build rules to cover previously uncovered training examples. Each rule is grown by adding conditions until no negative examples are covered. It normally generates rules for each class from the most rare class to the most common class. So, in this view, Ripper can be view as a one-class learner.

An interesting aspect of one-class (recognition-based) learning is that, under certain conditions such as multi-modality of the domain space, one class approaches to solving the classification problem may in fact be superior to discriminative (two-class) approaches (such as decision trees or Neural Networks) [34].

Raskutti and Kowalczyk demonstrated the optimality of one-class SVMs over two-class ones in certain important imbalanced-data domains, including genomic data [42]. In particular, they showed that one-class learning is particularly useful when used on extremely unbalanced data sets composed of a high dimensional noisy feature space. They argued that the one-class approach is related to aggressive feature selection methods, but is more practical since feature selection can often be too expensive to apply.

3.4. Ensemble learning methods

Ensemble learning has established its superiority in machine learning in recent years, of which Boosting and Bagging are the most successful approaches. Ensemble learning methods have been extensively used to handle class imbalance problems. These methods combine the results of many classifiers. Their successes attribute to the fact that their base learners usually are of diversity in principle or induced with various class distributions.

AdaBoost, introduced by Freund and Schapire [43], solved many of the practical difficulties of the earlier boosting algorithms. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set.

As stated in cost-sensitive learning, by making AdaBoost's weight-update rule cost-sensitive, the resulting system Adacost [33] has been empirically shown to produce lower cumulative misclassification costs than AdaBoost. Thus, it can be used to address class imbalance problem.

Rare-Boost [44] scales false-positive examples in proportion to how well they are distinguished from true-positive examples and scales false-positive examples in proportion to how well they are distinguished from true-negative examples.

Another algorithm that uses boosting to address the class imbalance problem is SMOTEBoost [45]. This algorithm recognizes that boosting may suffer from the same problems as over-sampling (e.g., overfitting). Instead of changing the distributions of training data by updating the weights associated with each example, SMOTEBoost alters the distributions by adding new examples of minority class using the SMOTE algorithm.

Experiment results indicated that the mixture-of-experts approach performs well, generally outperforming AdaBoost with respect to precision and recall on text classification problems, and doing especially well at covering the minority examples. More detailed experiments are presented in [46].

MetaCost [32] is another ensemble method. It begins to learn an internal cost-sensitive model. Then, estimates class probabilities using bagging and then re-labels the training examples with their minimum expected cost classes, and finally relearns a model using the modified training set.

Chan and Stolfo [47] run a set of preliminary experiments to identify a good class distributions and then do resampling to generate multiple training sets with the desired class distributions. Each training set typically includes all minority-class examples and a subset of the majority-class examples; however, each majority-class example is guaranteed to occur in at least one training set, so no data is wasted. The learning algorithm is applied to each training set and meta-learning is used to form a composite learner from the resulting classifiers. Since it is a wrapper method, it can be used with any learning method internally. The same basic approach for partitioning the data and learning multiple classifiers has been used with support vector machines. The resulting SVM ensembles [48] was shown to outperform both under-sampling and over-sampling. While these ensemble approaches are effective for dealing with the class imbalance problem, they assume that a good class distributions is known. This can be estimated using some preliminary runs, but it is time consuming. From the style constructing the training data sets, they can be viewed as a variant of bagging.

Phua et. al. [49] combined bagging and stacking to identify the best mix of classifiers. In their insurance fraud detection domain, they noted that stacking-bagging achieves the best cost-savings.

Besides ensemble learning algorithms of boosting and bagging style, Kotsiantis and Pintelas [12] used three agents (the first learns using Naive Bayes, the second using C4.5 and the third using 5NN) on a filtered version of training data and combined their predictions according to a voting scheme. This technique attempts to achieve diversity in the errors of the academic models by using different learning algorithms. The intuition is that the models generated using different learning biases are more likely to make errors in different ways. They also used feature selection of the training data because in small data sets the amount of class imbalance affects more the induction and thus feature selection makes the problem less difficult.

Motivated Zheng and Srihari's work [23], Castillo and Serrano [24] do not particularly focus on feature selection, but make it a part of their complete framework. They use a multi-strategy classifier system to construct multiple learners, each doing its own feature selection based on genetic algorithm. Their proposed system also combines the predictions of each learner using genetic algorithms.

3. Evaluation metrics

Accuracy is the most common evaluation metric for most traditional application. But accuracy is not suitable to evaluate imbalanced data sets, since many practitioners have observed that for extremely skewed class distributions the recall of the minority class is often 0, which means that there are no classification rules generated for the minority class. Using terminology from information retrieval, the minority class has much lower precision and recall than the majority class.

Accuracy places more weight on the majority class than on minority class, which makes it difficult for a classifier to perform well on the minority class. For this reason, additional metrics are coming into widespread use.

In recent years, several new metrics have been proposed or introduced from other domains for imbalanced data sets. They are precision and recall from information retrieval domain, ROC and AUC (Area Under the roc Curve) from medical domain, F-value, maximum geometry mean (MGM) of the accuracy on the majority class and the minority class, maximum sum (MS) of the accuracy. All the metrics can be classified into two categories: metrics based on confusion matrix directly and that based on accuracy of binary classes or precision and recall directly. Accuracy, precision and recall, FP rate, TP rate, ROC and AUC fall into the first, while F-values and other more complex metrics, such as MGM of the accuracy on the majority class and the minority class, MS, fall into the other.

Table 1 shows the confusion matrix, and a good understanding to confusion matrix will be helpful.

Table 1. Confusion matrix

		Prediction	
		positive	negative
Real	positive	**TP**(True Positive)	**FN**(False Negative)
	negative	**FP**(False Positive)	**TN**(True Negative)

As promised at the beginning of the paper, the class label of the minority class is positive, and the class label of the majority class is negative. Fig. 1 presents the most well known evaluation metrics. As shown in Table 1, TP and TN denote the number of positive and negative examples that are classified correctly, while FN and FP denote the number of misclassified positive and negative examples respectively. By definition, Accuracy, Precision$_+$, Recall$_+$, FP rate, TP rate and F-value can be represented by Equations from Eq.1 to Eq.6 as shown in Fig.2, where Precision$_+$ and Recall$_+$ are precision and recall of the minority class.

FP rate denotes the percentage of the misclassified negative examples, and TP rate is the percentage of the correctly classified positive examples. The point (0, 1) is the ideal point of the learners. That is there is no positive examples were misclassified to negative class, and vice versa.

F-value (or F-measure) is a popular evaluation metric for imbalance problem [50]. It is a kind of combination of recall and precision, which are effective metrics for information retrieval community where the imbalance problem exists. F-value is high when both recall and precision are high, and can be adjusted through changing the value of β, where β corresponds to relative importance of precision vs. recall, for example, F-1 counts both equally, while F-2 counts recall twice as much.

Perhaps the most common metric to assess overall classification performance is ROC analysis and the associated use of the area under the ROC curve (AUC) [51]. In detail, ROC curve is a two-dimensional graph in which TP rate is plotted on the y-axis and FP rate is plotted on the x-axis. ROC curves, like precision-recall curves, can also be used to assess different trade-offs ROC curve depicts relative trade-offs between benefits (TP rate) and costs (FP rate), that is the number of positive examples correctly classified can be increased at the expense of introducing additional false positives. A major disadvantage of ROC analysis is that it does not deliver a single, easy to use performance measure like accuracy directly. AUC does not place more emphasis on one class over the other, so it is not biased against the minority class.

$$Accuracy = (TP + TN)/(TP + FN + FP + TN) \qquad \text{Eq.1}$$

$$\mathrm{Pr}ecison_{+} = TP/(TP + FP) \qquad \text{Eq.2}$$

$$\mathrm{Re}call_{+} = TP/(TP + FN) \qquad \text{Eq.3}$$

$$FP\ rate = FP/(FP + TN) \qquad \text{Eq.4}$$

$$TP\ rate = TP/(TP + FN) \qquad \text{Eq.5}$$

$$F_{-value} = \frac{(1+\beta^2)\mathrm{Re}call * \mathrm{Pr}ecision}{\beta^2 * \mathrm{Re}call + \mathrm{Pr}ecision} \qquad \text{Eq.6}$$

$$MGM = \sqrt{Accuracy_{+} * Accuracy_{-}} \qquad \text{Eq.7}$$

$$MS = Accuracy_{+} + Accuracy_{-} \qquad \text{Eq.8}$$

Fig.2. Evaluation metrics based on confusion matrix

Besides, minimum cost criterion, is also used to evaluate the performance of classifiers in learning from imbalanced data sets when performing cost-sensitive learning.

When applying machine learning algorithms to real world applications, rarely would one or more of these assumptions hold, but to select a classifier, certain conditions must exist, and we may need more information. If one ROC curve dominates all others, then the best method is the one that produced the dominant curve, which is also the curve with the largest area (with maximum AUC). To select a classifier from the dominant curve, we need additional information, such as a target FP rate. On the other hand, if multiple curves dominate in different parts of the ROC space, then we can use the ROC Convex Hull method to select the optimal classifier [52].

4. Relations to other problems

However, it has also been observed that in some domains, for example the Sick data set, standard machine learning algorithms are capable of inducing good classifiers, even using highly imbalanced training sets. This shows that class imbalance is not the only problem responsible for the decrease in performance of learning algorithms. Class imbalance is not the only problem to contend with. Besides, the distributions within each class of the data, i.e. within class imbalance, are also relevant [53].

It was found that in certain cases, addressing the small disjuncts problem with regardless of the class imbalance problem was sufficient to increase performance. Experiments by Jo and Japkowicz suggested that the problem is not directly caused by class imbalances, but rather, that class imbalances may yield small disjuncts which, in turn, will cause degradation [19]. A cluster-based over-sampling approach was proposed, whose idea is to consider not only the between-class imbalance but also the within-class imbalance and to over-sample the dataset by rectifying these two types of imbalances simultaneously.

The experiments results of Prati et. al. [54] , using a discrimination-based inductive scheme, suggested that the problem is not solely caused by class imbalance, but is also related to the degree of data overlapping among the classes.

It was also found that data duplication is generally harmful, although for classifiers such as Naive Bayes and Perceptrons with Margins, high degrees of duplication are necessary to harm classification [55]. It was argued that the reason why class imbalances and overlapping classes are related is that misclassification often occurs near class boundaries where overlap usually occurs as well.

Weiss [10] investigated the relation between class imbalance and training set size. Experiments showed that while the position of the best class distributions varies somewhat with training-set size, in many cases, especially with error rate, the variation is small which gives support to the notion that there is a "best" marginal class distribution for a learning task. The results also indicated that, for any fixed class distribution, increasing the size of the training set always leads to improved classifier performance.

5. Conclusion

Learning from imbalanced data sets is an important issue in machine learning. A direct method to solve the imbalance problem is artificially balancing the class distributions, and its effectiveness has been empirically analyzed in [11]. However, there is some evidence that re-balancing the class distributions artificially does not have much effect on the performance of the induced classifier, since some learning systems are not sensitive to differences in class distributions. It seems that we still need a clearer and deeper understanding of how class distribution affects each phase of the learning process for more learners except decision trees. A deeper understanding of the basics will help us to design better methods for dealing with the problem of learning with skewed class distributions.

As is stated in section 5, some data sets are immune to class imbalance problem. It was argued that the class imbalance problem is not directly caused by class imbalance, but rather, that class imbalance may yield small disjuncts which, in turn, will cause degradation. Though maximum specification bias in induction processes and dealing with the problems of within class imbalance and between class imbalance have present their effectiveness according to minority class, more effective methods are needed. Current researches on small disjuncts are ad hoc, so standard metrics for the degree of small disjuncts are deadly in need.

Since machine learning is an application-driven science and the class imbalance problem and some other related ones are of domain-specific nature, realizing to explore idiographic solutions for specific applications is very important and valuable for practitioners, and a better data understanding and more knowledge on the domain will be helpful in the process.

6. References

[1] S. Kotsiantis, D. Kanellopoulos and P. Pintelas, "Handling imbalanced datasets: A review", *GESTS International Transactions on Computer Science and Engineering 30 (1)* ，2006， pp. 25-36

[2] S. Visa and A. Ralescu, "Issues in Mining Imbalanced Data Sets-A Review Paper" , in *Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, MAICS-2005, Dayton, 2005, pp. 67-73.

[3] M. C. Monard and G. E. A. P. A. Batista, "Learning with Skewed Class Distribution", in *Advances in Logic, Artificial Intelligence and Robotics*, Sao Paulo, SP, IOS Press, 2002, pp. 173-180.

[4] G. Weiss, "Mining with rarity: A unifying framework", *SIGKDD Explorations 6(1)*, 2004, pp. 7-19.

[5] M. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown", in *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets II*, 2003, pp. 73-80.

[6] L.Breiman, J. R.Friedeman, C. Stone, and R.A. Olshen, "Classification and Regression Trees", Chapman and Hall/CRC Press, 1984.

[7] N. Japkowicz, "Class imbalances: Are we focusing on the right issue?", *Proceedings of the ICML-2003 Workshop: Learning with Imbalanced Data Sets II*, 2003, pp. 17-23.

[8] J. Laurikkala, “Improving Identification of Difficult Small Classes by Balancing Class Distribution”. *Technical Report, A-2001-2*, University of Tampere, 2001.

[9] H. Guo and V. Herna, “Learning from imbalanced data sets with boosting and data generation: The data boosting approach”, *SIGKDD Explorations 6(1)*, 2004, pp. 30-39.

[10] G.M. Weiss, “The Effect of Small Disjuncts and Class Distribution on Decision Tree Learning”, *Ph.D. Dissertation, Department of Computer Science, Rutgers University*, New Brunswick, New Jersey, May 2003.

[11] N. Japkowicz, “Learning from Imbalanced Data Sets: a Comparison of Various Strategies”, *AAAI Workshop on Learning from Imbalanced Data Sets*, Menlo Park, CA, AAAI Press, 2000.

[12] S. Kotsiantis and P. Pintelas, “Mixture of Expert Agents for Handling Imbalanced Data Sets”, *Annals of Mathematics, Computing & TeleInformatics, Vol 1,* 2003, pp. 46-55.

[13] P.E. Hart, “The condensed nearest neighbor rule”, *IEEE Transactions on Information Theory*, *IT-14,* 1968, pp. 515-516.

[14] M. Kubat and S. Matwin, “Addressing the curse of imbalanced training sets: One sided Selection”, In *Proceedings of the Fourteenth International Conference on Machine Learning*, Nashville, Tennesse, Morgan Kaufmann, 1997, pp. 179-186.

[15] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Oversampling Technique”, *Journal of Artificial Intelligence Research, 16*, 2002, pp. 321-357.

[16] H. Han, W.Y. Wang, and B.H. Mao, “Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning”, in *Proceedings of the International Conference on Intelligent Computing 2005*, *Part I,* LNCS 3644, 2005, pp. 878–887.

[17] G. M. Weiss and F. Provost, “Learning when training data are costly: the effect of class distribution on tree induction”, *Journal of Artificial Intelligence Research, 19,* 2003, pp. 315-354.

[18] H. Han, L. Wang, M. Wen, and W. Y. Wang, “Over-sampling Algorithm Based on Preliminary Classification in Imbalanced Data Sets Learning”, *Journal of computer allocations (in Chinese), 2006 Vol.26 No.8*, pp.1894-1897.

[19] Taeho Jo and N. Japkowicz, “Class Imbalances versus Small Disjuncts”, *SIGKDD Explorations*. *Volume 6, Issue 1*, 2004, pp. 40-49.

[20] G. Batista, M. Prati, and M. Monard, “A study of the behavior of several methods for balancing machine learning training data”, *SIGKDD Explorations 6(1)*, 2004, pp. 20-29.

[21] G. Forman, “An extensive empirical study of feature selection metrics for text classification”. *Journal of Machine Learning Research*, *3*, 2003, pp. 1289-1305.

[22] D. Mladenic and M. Grobelnik, “Feature selection for unbalanced class distribution and naive Bayes”, in *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999, pp. 258-267.

[23] Z. Zheng, X. Wu, and R. Srihari, “Feature selection for text categorization on imbalanced data”, *SIGKDD Explorations, 6(1)*, 2004, pp. 80-89.

[24] M. Castillo and J. Serrano, “A multistrategy approach for digital text categorization from imbalanced documents”, *SIGKDD Explorations, 6(1)*, 2004, pp. 70-79.

[25] P. Van Der Putten and M. Van Someren, “A bias-variance analysis of a real world learning problem: the coil challenge 2000.” *Machine Learning 57(1-2)*, 2004, pp. 177-195.

[26] D.L. Wilson, "*Asymptotic Properties of Nearest Neighbor Rules using Edited Data*", *IEEE Trans. on Systems, Man and Cybernetics, Vol. 2*, 1972, pp. 408-420.

[27] C. Drummond and R. C. Holte, “C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling”, in *ICML 2003 Workshop on Learning from Imbalanced Data Sets II* , Washington, DC, 2003.

[28] R. Barandela, J.S. Sánchez, V. García, and E. Rangel, “Strategies for learning in class imbalance problems”, *Pattern Recognition 36(3)*, 2003, pp. 849-851.

[29] G. Wu and E. Chang, “Class-Boundary Alignment for Imbalanced Dataset Learning”, in *ICML 2003 Workshop on Learning from Imbalanced Data Sets II*, Washington, DC, 2003.

[30] K. Veropoulos, C. Campbell, and N. Cristianini, “Controlling the sensitivity of support vector machines”, in *Proceedings of the International Joint Conference on AI*, 1999, pp. 55–60.

[31] K.Z. Huang, H.Q. Yang, I. King, and M.R. Lyu, “Learning Classifiers from Imbalanced Data Based on Biased Minimax Probability Machine”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , 2004.

[32] P. Domingos, “MetaCost: A general method for making classifiers cost-sensitive”, in *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, ACM Press, 1999, pp. 155-164.

[33] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, “AdaCost: misclassification cost-sensitive boosting”, in *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999, pp. 99-105.

[34] N. Japkowicz, “Supervised versus unsupervised binary learning by feed forward neural networks”, *Machine Learning, 42(1/2)*, 2001, pp. 97-122.

[35] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “ Estimating the support of a high-dimensional distribution”, *Neural Computation, 13(7)*, 2001, pp. 1443-1472.

[36] D. Tax, “One-class classification”, *Ph.D. dissertation*, Delft University of Technology, 2001.

[37] L. M. Manevitz and M. Yousef, “One-class SVMs for document classification”, *Journal of Machine Learning Research, 2*, 2001, pp. 139-154.

[38] P. Riddle, R. Segal, and O. Etzioni, “Representation design and brute-force induction in a Boeing manufacturing design”, *Applied Artificial Intelligence, 8*, 1994, pp. 125-147.

[39] M. Kubat, R. Holte, and S. Matwin, “Learning when negative examples abound”, in *Proceedings of the Ninth European Conference on Machine Learning, LNAI 1224*, 1997, Springer, pp. 146-153.

[40] W. W. Cohen, “Fast effective rule induction”, in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 115-123.

[41] I. Tomek, “Two Modifications of CNN”, *IEEE Transactions on Systems Man and Communications SMC-6*, 1976, pp. 769-772.

[42] B. Raskutti and A. Kowalczyk, “Extreme rebalancing

for SVMs: a case study", *SIGKDD Explorations, 6(1)*, 2004, pp. 60-69.

[43] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, *55(1)*, 1997, pp. 119–139.

[44] M. V. Joshi, V. Kumar, and R. C. Agarwal, "Evaluating boosting algorithms to classify rare cases: comparison and improvements", in *Proceedings of the First IEEE International Conference on Data Mining*, 2001, pp. 257-264.

[45] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting", in *Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, Dubrovnik, Croatia, 2003, pp. 107-119.

[46] A. Estabrooks, Taeho Jo, and N. Japkowicz, "A Multiple Resampling Method for Learning from Imbalanced Data Sets", *Computational Intelligence 20 (1),* 2004, pp. 18-36.

[47] P. K. Chan and S. J. Stolfo, "Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection", in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 164-168.

[48] R. Yan, Y. Liu, R. Jin, and A. Hauptmann, "On predicting rare classes with SVM ensembles in scene classification". in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2003.

[49] C. Phua and D. Alahakoon, "Minority report in fraud detection: Classification of skewed data", *SIGKDD Explorations, 6(1)*, 2004, pp. 50-59.

[50] Estabrooks and N. Japkowicz, "A mixture-of-experts framework for learning from unbalanced data sets", in *Proceedings of the 2001 Intelligent Data Analysis Conference*, 2001, pp. 34-43.

[51] Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms", *Pattern Recognition, 30(7),* 1997, pp. 1145-1159.

[52] F. Provost and T. Fawcett, "Robust classification for imprecise environments", *Machine Learning, 42*, 2001, pp. 203-231.

[53] N. Japkowicz, "Concept-learning in the presence of between-class and within-class imbalances", in *Proceedings of the Fourteenth Conference of the Canadian Society for Computational Studies of Intelligence*, 2001, pp. 67-77.

[54] R. C. Prati, G. E. Batista, A. P. A. and M. C. Monard, "Class Imbalances versus Class Overlapping: an Analysis of a Learning System Behavior", in *MICAI 2004*, LNAI 2972, 2004, pp. 312–321.

[55] Kolez, A. Chowdhury, and J. Alspector, "Data duplication: An imbalance problem?", in *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data sets II*, 2003.