Learning from Imbalanced Data

Haibo He, *Member*, *IEEE*, and Edwardo A. Garcia

Abstract—With the continuous expansion of data availability in many large-scale, complex, and networked systems, such as surveillance, security, Internet, and finance, it becomes critical to advance the fundamental understanding of knowledge discovery and analysis from raw data to support decision-making processes. Although existing knowledge discovery and data engineering techniques have shown great success in many real-world applications, the problem of learning from imbalanced data (the imbalanced learning problem) is a relatively new challenge that has attracted growing attention from both academia and industry. The imbalanced learning problem is concerned with the performance of learning algorithms in the presence of underrepresented data and severe class distribution skews. Due to the inherent complex characteristics of imbalanced data sets, learning from such data requires new understandings, principles, algorithms, and tools to transform vast amounts of raw data efficiently into information and knowledge representation. In this paper, we provide a comprehensive review of the development of research in learning from imbalanced data. Our focus is to provide a critical review of the nature of the problem, the state-of-the-art technologies, and the current assessment metrics used to evaluate learning performance under the imbalanced learning scenario. Furthermore, in order to stimulate future research in this field, we also highlight the major opportunities and challenges, as well as potential important research directions for learning from imbalanced data.

Index Terms—Imbalanced learning, classification, sampling methods, cost-sensitive learning, kernel-based learning, active learning, assessment metrics.

1 Introduction

Recent developments in science and technology have enabled the growth and availability of raw data to occur at an explosive rate. This has created an immense opportunity for knowledge discovery and data engineering research to play an essential role in a wide range of applications from daily civilian life to national security, from enterprise information processing to governmental decision-making support systems, from microscale data analysis to macroscale knowledge discovery. In recent years, the imbalanced learning problem has drawn a significant amount of interest from academia, industry, and government funding agencies. The fundamental issue with the imbalanced learning problem is the ability of imbalanced data to significantly compromise the performance of most standard learning algorithms. Most standard algorithms assume or expect balanced class distributions or equal misclassification costs. Therefore, when presented with complex imbalanced data sets, these algorithms fail to properly represent the distributive characteristics of the data and resultantly provide unfavorable accuracies across the classes of the data. When translated to real-world domains, the imbalanced learning problem represents a recurring problem of high importance with wide-ranging implications, warranting increasing exploration. This increased interest is reflected in the recent installment of several major workshops, conferences, and special issues including the American Association for Artificial Intelligence (now the Association for the Advancement of Artificial Intelligence) workshop on Learning from Imbalanced Data Sets (AAAI '00) [1], the International Conference on Machine Learning workshop on Learning from Imbalanced Data Sets (ICML'03) [2], and the Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining Explorations (ACM SIGKDD Explorations '04) [3].

With the great influx of attention devoted to the imbalanced learning problem and the high activity of advancement in this field, remaining knowledgeable of all current developments can be an overwhelming task. Fig. 1 shows an estimation of the number of publications on the imbalanced learning problem over the past decade based on the Institute of Electrical and Electronics Engineers (IEEE) and Association for Computing Machinery (ACM) databases. As can be seen, the activity of publications in this field is growing at an explosive rate. Due to the relatively young age of this field and because of its rapid expansion, consistent assessments of past and current works in the field in addition to projections for future research are essential for long-term development. In this paper, we seek to provide a survey of the current understanding of the imbalanced learning problem and the state-of-the-art solutions created to address this problem. Furthermore, in order to stimulate future research in this field, we also highlight the major opportunities and challenges for learning from imbalanced data.

In particular, we first describe the nature of the imbalanced learning problem in Section 2, which provides the foundation for our review of imbalanced learning solutions. In Section 3, we provide a critical review of the innovative research developments targeting the imbalanced learning

• *The authors are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030. E-mail: hhe@stevens.edu, edwardo.garcia@nyu.edu.*

Manuscript received 1 May 2008; revised 6 Oct. 2008; accepted 1 Dec. 2008; published online 19 Dec. 2008.
Recommended for acceptance by C. Clifton.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-05-0233.
Digital Object Identifier no. 10.1109/TKDE.2008.239.

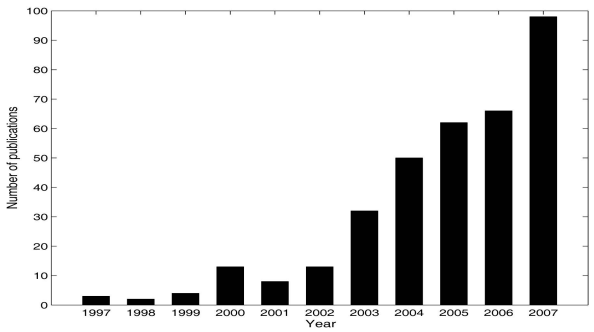


Fig. 1. Number of publications on imbalanced learning.

problem, including sampling methods, cost-sensitive learning methods, kernel-based learning methods, and active learning methods. Assessment metrics for imbalanced learning are reviewed in Section 4, which provides various suggested methods that are used to compare and evaluate the performance of different imbalanced learning algorithms. Considering how learning from imbalanced data is a relatively new topic in the research community, in Section 5, we present a detailed discussion on the opportunities and challenges for research development in this field. We hope that this section will provide some useful suggestions to promote and guide the long-term advancement of research in this area. Finally, a conclusion is provided in Section 6.

2 Nature of the Problem

Technically speaking, any data set that exhibits an unequal distribution between its classes can be considered imbalanced. However, the common understanding in the community is that imbalanced data correspond to data sets exhibiting significant, and in some cases extreme, imbalances. Specifically, this form of imbalance is referred to as a *between-class imbalance*; not uncommon are between-class imbalances on the order of 100:1, 1,000:1, and 10,000:1, where in each case, one class severely outrepresents another [4], [5], [6]. Although this description would seem to imply that all between-class imbalances are innately binary (or two-class), we note that there are multiclass data in which imbalances exist between the various classes [7], [8], [9], [10], [11], [12]. In this paper, we only briefly touch upon the multiclass imbalanced learning problem, focusing instead on the two-class imbalanced learning problem for space considerations.

In order to highlight the implications of the imbalanced learning problem in the real world, we present an example from biomedical applications. Consider the "Mammography Data Set," a collection of images acquired from a series of mammography exams performed on a set of distinct patients, which has been widely used in the analysis of algorithms addressing the imbalanced learning problem [13], [14], [15]. Analyzing the images in a binary sense, the natural classes (labels) that arise are "Positive" or "Negative" for an image representative of a "cancerous" or "healthy" patient, respectively. From experience, one would expect the number of noncancerous patients to exceed greatly the number of cancerous patients; indeed, this data set contains 10,923 "Negative" (majority class) samples and 260 "Positive" (minority class) samples. Preferably, we require a classifier that provides a balanced degree of predictive accuracy (ideally 100 percent) for both the minority and majority classes on the data set. In reality, we find that classifiers tend to provide a severely imbalanced degree of accuracy, with the majority class having close to 100 percent accuracy and the minority class having accuracies of 0-10 percent, for instance [13], [15]. Suppose a classifier achieves 10 percent accuracy on the minority class of the mammography data set. Analytically, this would suggest that 234 minority samples are misclassified as majority samples. The consequence of this is equivalent to 234 cancerous patients classified (diagnosed) as noncancerous. In the medical industry, the ramifications of such a consequence can be overwhelmingly costly, more so than classifying a noncancerous patient as cancerous [16]. Therefore, it is evident that for this domain, we require a classifier that will provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class. Furthermore, this also suggests that the conventional evaluation practice of using singular assessment criteria, such as the overall accuracy or error rate, does not provide adequate information in the case of imbalanced learning. Therefore, more informative assessment metrics, such as the receiver operating characteristics curves, precision-recall curves, and cost curves, are necessary for conclusive evaluations of performance in the presence of imbalanced data. These topics will be discussed in detail in Section 4 of this paper. In addition to biomedical applications, further speculation will yield similar consequences for domains such as fraud detection, network intrusion, and oil-spill detection, to name a few [5], [16], [17], [18], [19].

Imbalances of this form are commonly referred to as *intrinsic*, i.e., the imbalance is a direct result of the nature of the dataspace. However, imbalanced data are not solely restricted to the intrinsic variety. Variable factors such as time and storage also give rise to data sets that are imbalanced. Imbalances of this type are considered *extrinsic*, i.e., the imbalance is not directly related to the nature of the dataspace. Extrinsic imbalances are equally as interesting as their intrinsic counterparts since it may very well occur that the dataspace from which an extrinsic imbalanced data set is attained may not be imbalanced at all. For instance, suppose a data set is procured from a continuous data stream of balanced data over a specific interval of time, and if during this interval, the transmission has sporadic interruptions where data are not transmitted, then it is possible that the acquired data set can be imbalanced in which case the data set would be an extrinsic imbalanced data set attained from a balanced dataspace.

In addition to *intrinsic* and *extrinsic* imbalance, it is important to understand the difference between *relative imbalance* and *imbalance due to rare instances* (or "*absolute rarity*") [20], [21]. Consider a mammography data set with 100,000 examples and a 100:1 between-class imbalance. We would expect this data set to contain 1,000 minority class examples; clearly, the majority class dominates the minority class. Suppose we then double the sample space by testing more patients, and suppose further that the distribution

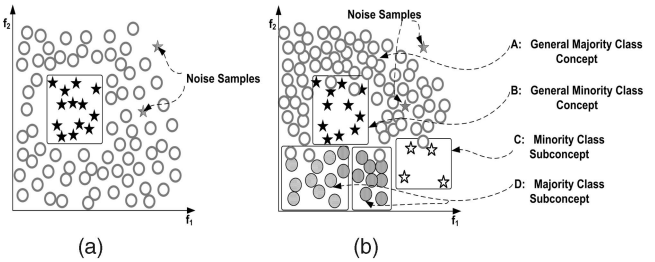


Fig. 2. (a) A data set with a between-class imbalance. (b) A high-complexity data set with both between-class and within-class imbalances, multiple concepts, overlapping, noise, and lack of representative data.

does not change, i.e., the minority class now contains 2,000 examples. Clearly, the minority class is still outnumbered; however, with 2,000 examples, the minority class is not necessarily rare in its own right but rather relative to the majority class. This example is representative of a relative imbalance. Relative imbalances arise frequently in real-world applications and are often the focus of many knowledge discovery and data engineering research efforts. Some studies have shown that for certain relative imbalanced data sets, the minority concept is accurately learned with little disturbance from the imbalance [22], [23], [24]. These results are particularly suggestive because they show that the degree of imbalance is not the only factor that hinders learning. As it turns out, data set complexity is the primary determining factor of classification deterioration, which, in turn, is amplified by the addition of a relative imbalance.

Data complexity is a broad term that comprises issues such as overlapping, lack of representative data, small disjuncts, and others. In a simple example, consider the depicted distributions in Fig. 2. In this figure, the stars and circles represent the minority and majority classes, respectively. By inspection, we see that both distributions in Figs. 2a and 2b exhibit relative imbalances. However, notice how Fig. 2a has no overlapping examples between its classes and has only one concept pertaining to each class, whereas Fig. 2b has both multiple concepts and severe overlapping. Also of interest is subconcept C in the distribution of Fig. 2b. This concept might go unlearned by some inducers due to its lack of representative data; this issue embodies imbalances due to rare instances, which we proceed to explore.

Imbalance due to *rare instances* is representative of domains where minority class examples are very limited, i.e., where the target concept is rare. In this situation, the lack of representative data will make learning difficult regardless of the between-class imbalance [20]. Furthermore, the minority concept may additionally contain a subconcept with limited instances, amounting to diverging degrees of classification difficulty [25], [26]. This, in fact, is the result of another form of imbalance, a *within-class imbalance*, which concerns itself with the distribution of representative data for subconcepts within a class [27], [28], [29]. These ideas are again highlighted in our simplified example in Fig. 2. In Fig. 2b, cluster B represents the dominant minority class concept and cluster C represents a subconcept of the minority class. Cluster D represents two subconcepts of the majority class and cluster A (anything not enclosed) represents the dominant majority class concept. For both classes, the number of examples in the dominant clusters significantly outnumber the examples in their respective subconcept clusters, so that this dataspace exhibits both within-class and between-class imbalances. Moreover, if we completely remove the examples in cluster B, the dataspace would then have a homogeneous minority class concept that is easily identified (cluster C), but can go unlearned due to its severe underrepresentation.

The existence of within-class imbalances is closely intertwined with the problem of *small disjuncts*, which has been shown to greatly depreciate classification performance [23], [27], [28], [29]. Briefly, the problem of small disjuncts can be understood as follows: A classifier will attempt to learn a concept by creating multiple disjunct rules that describe the main concept [20], [25], [26]. In the case of homogeneous concepts, the classifier will generally create large disjuncts, i.e., rules that cover a large portion (cluster) of examples pertaining to the main concept. However, in the case of heterogeneous concepts, small disjuncts, i.e., rules that cover a small cluster of examples pertaining to the main concept, arise as a direct result of underrepresented subconcepts [20], [25], [26]. Moreover, since classifiers attempt to learn both majority and minority concepts, the problem of small disjuncts is not only restricted to the minority concept. On the contrary, small disjuncts of the majority class can arise from noisy misclassified minority class examples or underrepresented subconcepts. However, because of the vast representation of majority class data, this occurrence is infrequent. A more common scenario is that noise may influence disjuncts in the minority class. In this case, the validity of the clusters corresponding to the small disjuncts becomes an important issue, i.e., whether these examples represent an actual subconcept or are merely attributed to noise. For example, in Fig. 2b, suppose a classifier generates disjuncts for each of the two noisy minority samples in cluster A, then these would be illegitimate disjuncts attributed to noise compared to cluster C, for example, which is a legitimate cluster formed from a severely underrepresented subconcept.

The last issue we would like to discuss is the *combination of imbalanced data and the small sample size problem* [30], [31]. In many of today's data analysis and knowledge discovery applications, it is often unavoidable to have data with high dimensionality and small sample size; some specific examples include face recognition and gene expression data analysis, among others. Traditionally, the small sample size problem has been studied extensively in the pattern recognition community [30]. Dimensionality reduction methods have been widely adopted to handle this issue, e.g., principal component analysis (PCA) and various extension methods [32]. However, when the representative data sets' concepts exhibit imbalances of the forms described earlier, the combination of imbalanced data and small sample size presents a new challenge to the community [31]. In this situation, there are two critical issues that arise simultaneously [31]. First, since the sample size is small, all of the issues related to absolute rarity and within-class imbalances are applicable. Second and more importantly, learning algorithms often fail to

generalize inductive rules over the sample space when presented with this form of imbalance. In this case, the combination of small sample size and high dimensionality hinders learning because of difficultly involved in forming conjunctions over the high degree of features with limited samples. If the sample space is sufficiently large enough, a set of general (albeit complex) inductive rules can be defined for the dataspace. However, when samples are limited, the rules formed can become too specific, leading to overfitting. In regards to learning from such data sets, this is a relatively new research topic that requires much needed attention in the community. As a result, we will touch upon this topic again later in our discussions.

3 The State-of-the-Art Solutions for Imbalanced Learning

The topics discussed in Section 2 provide the foundation for most of the current research activities on imbalanced learning. In particular, the immense hindering effects that these problems have on standard learning algorithms are the focus of most of the existing solutions. When standard learning algorithms are applied to imbalanced data, the induction rules that describe the minority concepts are often fewer and weaker than those of majority concepts, since the minority class is often both outnumbered and underrepresented. To provide a concrete understanding of the direct effects of the imbalanced learning problem on standard learning algorithms, we observe a case study of the popular decision tree learning algorithm.

In this case, imbalanced data sets exploit inadequacies in the splitting criterion at each node of the decision tree [23], [24], [33]. Decision trees use a recursive, top-down greedy search algorithm that uses a feature selection scheme (e.g., information gain) to select the best feature as the split criterion at each node of the tree; a successor (leaf) is then created for each of the possible values corresponding to the split feature [26], [34]. As a result, the training set is successively partitioned into smaller subsets that are ultimately used to form disjoint rules pertaining to class concepts. These rules are finally combined so that the final hypothesis minimizes the total error rate across each class. The problem with this procedure in the presence of imbalanced data is two-fold. First, successive partitioning of the dataspace results in fewer and fewer observations of minority class examples resulting in fewer leaves describing minority concepts and successively weaker confidences estimates. Second, concepts that have dependencies on different feature space conjunctions can go unlearned by the sparseness introduced through partitioning. Here, the first issue correlates with the problems of relative and absolute imbalances, while the second issue best correlates with the between-class imbalance and the problem of high dimensionality. In both cases, the effects of imbalanced data on decision tree classification performance are detrimental. In the following sections, we evaluate the solutions proposed to overcome the effects of imbalanced data.

For clear presentation, we establish here some of the notations used in this section. Considering a given training data set S with m examples (i.e., $|S| = m$), we define: $S = \{(\boldsymbol{x_i}, y_i)\},\ i = 1, \ldots, m$, where $\boldsymbol{x_i} \in X$ is an instance in the n-dimensional feature space $X = \{f_1, f_2, \ldots, f_n\}$, and $y_i \in Y = \{1, \ldots, C\}$ is a class identity label associated with instance $\boldsymbol{x_i}$. In particular, $C = 2$ represents the two-class classification problem. Furthermore, we define subsets $S_{min} \subset S$ and $S_{maj} \subset S$, where S_{min} is the set of minority class examples in S, and S_{maj} is the set of majority class examples in S, so that $S_{min} \cap S_{maj} = \{\Phi\}$ and $S_{min} \cup S_{maj} = \{S\}$. Lastly, any sets generated from sampling procedures on S are labeled E, with disjoint subsets E_{min} and E_{maj} representing the minority and majority samples of E, respectively, whenever they apply.

3.1 Sampling Methods for Imbalanced Learning

Typically, the use of sampling methods in imbalanced learning applications consists of the modification of an imbalanced data set by some mechanisms in order to provide a balanced distribution. Studies have shown that for several base classifiers, a balanced data set provides improved overall classification performance compared to an imbalanced data set [35], [36], [37]. These results justify the use of sampling methods for imbalanced learning. However, they do not imply that classifiers cannot learn from imbalanced data sets; on the contrary, studies have also shown that classifiers induced from certain imbalanced data sets are comparable to classifiers induced from the same data set balanced by sampling techniques [22], [23]. This phenomenon has been directly linked to the problem of rare cases and its corresponding consequences, as described in Section 2. Nevertheless, for most imbalanced data sets, the application of sampling techniques does indeed aid in improved classifier accuracy.

3.1.1 Random Oversampling and Undersampling

The mechanics of *random oversampling* follow naturally from its description by adding a set E sampled from the minority class: for a set of randomly selected minority examples in S_{min}, augment the original set S by replicating the selected examples and adding them to S. In this way, the number of total examples in S_{min} is increased by $|E|$ and the class distribution balance of S is adjusted accordingly. This provides a mechanism for varying the degree of class distribution balance to any desired level. The oversampling method is simple to both understand and visualize, thus we refrain from providing any specific examples of its functionality.

While oversampling appends data to the original data set, *random undersampling* removes data from the original data set. In particular, we randomly select a set of majority class examples in S_{maj} and remove these samples from S so that $|S| = |S_{min}| + |S_{maj}| - |E|$. Consequently, undersampling readily gives us a simple method for adjusting the balance of the original data set S.

At first glance, the oversampling and undersampling methods appear to be functionally equivalent since they both alter the size of the original data set and can actually provide the same proportion of balance. However, this commonality is only superficial, each method introduces its own set of problematic consequences that can potentially hinder learning [25], [38], [39]. In the case of undersampling, the problem is relatively obvious: removing

examples from the majority class may cause the classifier to miss important concepts pertaining to the majority class. In regards to oversampling, the problem is a little more opaque: since oversampling simply appends replicated data to the original data set, multiple instances of certain examples become "tied," leading to overfitting [38]. In particular, overfitting in oversampling occurs when classifiers produce multiple clauses in a rule for multiple copies of the same example which causes the rule to become too specific; although the training accuracy will be high in this scenario, the classification performance on the unseen testing data is generally far worse [25].

3.1.2 Informed Undersampling

Two examples of informed undersampling that have shown good results are presented in [40], the *EasyEnsemble* and *BalanceCascade* algorithms. The objective of these two methods is to overcome the deficiency of information loss introduced in the traditional random undersampling method. The implementation of EasyEnsemble is very straightforward: it develops an ensemble learning system by independently sampling several subsets from the majority class and developing multiple classifiers based on the combination of each subset with the minority class data. In this way, EasyEnsemble can be considered as an unsupervised learning algorithm that explores the majority class data by using independent random sampling with replacement. On the other hand, the BalanceCascade algorithm takes a supervised learning approach that develops an ensemble of classifiers to systematically select which majority class examples to undersample. Specifically, for the first hypothesis of the ensemble, $H(1)$, consider a sampled set of majority class examples, E, such that $|E| = |S_{min}|$ and subject the ensemble to set $N = \{E \cup S_{min}\}$ to induce $H(1)$. Observing the results of $H(1)$, identify all $\boldsymbol{x_i} \in N$ that are correctly classified as belonging to S_{maj}, call this collection N^*_{maj}. Then, since we already have $H(1)$, it is reasonable to assume that N^*_{maj} is somewhat redundant in S_{maj} given that $H(1)$ is already trained. Based on this, we remove set N^*_{maj} from S_{maj} and generate a new sampled set of majority class samples, E, with $|E| = |S_{min}|$ and again subject the ensemble to set $N = \{E \cup S_{min}\}$ to derive $H(2)$. This procedure is iterated to a stopping criteria at which point a cascading combination scheme is used to form a final hypothesis [40].

Another example of informed undersampling uses the K-nearest neighbor (KNN) classifier to achieve undersampling. Based on the characteristics of the given data distribution, four KNN undersampling methods were proposed in [41], namely, NearMiss-1, NearMiss-2, NearMiss-3, and the "most distant" method. The NearMiss-1 method selects those majority examples whose average distance to the three closest minority class examples is the smallest, while the NearMiss-2 method selects the majority class examples whose average distance to the three farthest minority class examples is the smallest. NearMiss-3 selects a given number of the closest majority examples for each minority example to guarantee that every minority example is surrounded by some majority examples. Finally, the "most distance" method selects the majority class examples whose average distance to the three closest minority class examples is the largest. Experimental results suggested that the NearMiss-2 method can provide competitive results for imbalanced learning.

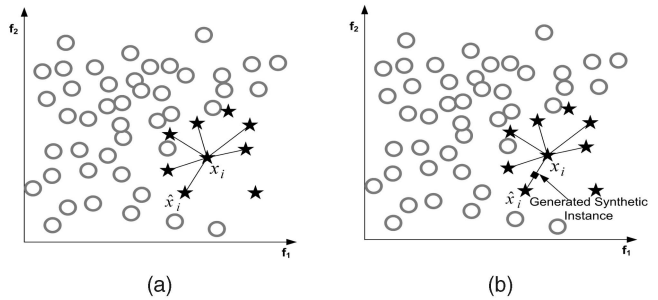


Fig. 3. (a) Example of the K-nearest neighbors for the $\boldsymbol{x_i}$ example under consideration ($K = 6$). (b) Data creation based on euclidian distance.

There are also other types of informed undersampling methods. For instance, the one-sided selection (OSS) method [42] selects a representative subset of the majority class E and combines it with the set of all minority examples S_{min} to form a preliminary set $N, N = \{E \cup S_{min}\}$. This set N is further refined by using a data cleaning technique. We will return to the discussion of this method in Section 3.1.5, now turning our attention to synthetic sampling methods.

3.1.3 Synthetic Sampling with Data Generation

In regards to synthetic sampling, the synthetic minority oversampling technique (SMOTE) is a powerful method that has shown a great deal of success in various applications [13]. The SMOTE algorithm creates artificial data based on the feature space similarities between existing minority examples. Specifically, for subset $S_{min} \in S$, consider the K-nearest neighbors for each example $\boldsymbol{x_i} \in S_{min}$, for some specified integer K; the K-nearest neighbors are defined as the K elements of S_{min} whose euclidian distance between itself and $\boldsymbol{x_i}$ under consideration exhibits the smallest magnitude along the n-dimensions of feature space X. To create a synthetic sample, randomly select one of the K-nearest neighbors, then multiply the corresponding feature vector difference with a random number between $[0, 1]$, and finally, add this vector to $\boldsymbol{x_i}$

$$\boldsymbol{x_{new}} = \boldsymbol{x_i} + (\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_i}) \times \delta, \tag{1}$$

where $\boldsymbol{x_i} \in S_{min}$ is the minority instance under consideration, $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ is one of the K-nearest neighbors for $\boldsymbol{x_i}$: $\hat{\boldsymbol{x}}_{\boldsymbol{i}} \in S_{min}$, and $\delta \in [0, 1]$ is a random number. Therefore, the resulting synthetic instance according to (1) is a point along the line segment joining $\boldsymbol{x_i}$ under consideration and the randomly selected K-nearest neighbor $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$.

Fig. 3 shows an example of the SMOTE procedure. Fig. 3a shows a typical imbalanced data distribution, where the stars and circles represent examples of the minority and majority class, respectively. The number of K-nearest neighbors is set to $K = 6$. Fig. 3b shows a created sample along the line between $\boldsymbol{x_i}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$, highlighted by the diamond shape. These synthetic samples help break the ties introduced by simple oversampling, and furthermore, augment the original data set in a manner that generally significantly improves learning. Though it has shown many promising benefits,

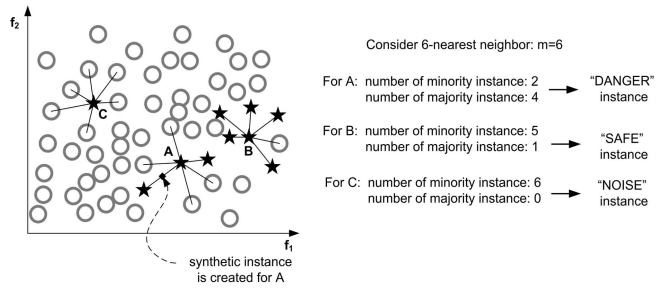


Fig. 4. Data creation based on Borderline instance.

the SMOTE algorithm also has its drawbacks, including over generalization and variance [43]. We will further analyze these limitations in the following discussion.

3.1.4 Adaptive Synthetic Sampling

In the SMOTE algorithm, the problem of over generalization is largely attributed to the way in which it creates synthetic samples. Specifically, SMOTE generates the same number of synthetic data samples for each original minority example and does so without consideration to neighboring examples, which increases the occurrence of overlapping between classes [43]. To this end, various adaptive sampling methods have been proposed to overcome this limitation; some representative work includes the Borderline-SMOTE [44] and Adaptive Synthetic Sampling (ADASYN) [45] algorithms.

Of particular interest with these adaptive algorithms are the techniques used to identify minority seed samples. In the case of Borderline-SMOTE, this is achieved as follows: First, determine the set of nearest neighbors for each $\boldsymbol{x_i} \in S_{min}$; call this set $S_{i:m-NN}, S_{i:m-NN} \subset S$. Next, for each $\boldsymbol{x_i}$, identify the number of nearest neighbors that belongs to the majority class, i.e., $|S_{i:m-NN} \cap S_{maj}|$. Finally, select those $\boldsymbol{x_i}$ that satisfy:

$$\frac{m}{2} \leq |S_{i:m-NN} \cap S_{maj}| < m. \tag{2}$$

Equation (2) suggests that only those $\boldsymbol{x_i}$ that have more majority class neighbors than minority class neighbors are selected to form the set "DANGER" [44]. Therefore, the examples in DANGER represent the borderline minority class examples (the examples that are most likely to be misclassified). The DANGER set is then fed to the SMOTE algorithm to generate synthetic minority samples in the neighborhood of the borders. One should note that if $|S_{i:m-NN} \cap S_{maj}| = m$, i.e., if all of the m nearest neighbors of $\boldsymbol{x_i}$ are majority examples, such as the instance C in Fig. 4, then this $\boldsymbol{x_i}$ is considered as noise and no synthetic examples are generated for it. Fig. 4 illustrates an example of the Borderline-SMOTE procedure. Comparing Fig. 4 and Fig. 3, we see that the major difference between Borderline-SMOTE and SMOTE is that SMOTE generates synthetic instances for each minority instance, while Borderline-SMOTE only generates synthetic instances for those minority examples "closer" to the border.

ADASYN, on the other hand, uses a systematic method to adaptively create different amounts of synthetic data according to their distributions [45]. This is achieved as follows: First, calculate the number of synthetic data examples that need to be generated for the entire minority class:

$$G = \big(|S_{maj}| - |S_{min}|\big) \times \beta, \tag{3}$$

where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after the synthetic data generation process. Next, for each example $\boldsymbol{x_i} \in S_{min}$, find the K-nearest neighbors according to the euclidean distance and calculate the ratio Γ_i defined as:

$$\Gamma_i = \frac{\Delta_i / K}{Z}, \quad i = 1, \ldots, |S_{min}|, \tag{4}$$

where Δ_i is the number of examples in the K-nearest neighbors of $\boldsymbol{x_i}$ that belong to S_{maj}, and Z is a normalization constant so that Γ_i is a distribution function ($\sum \Gamma_i = 1$). Then, determine the number of synthetic data samples that need to be generated for each $\boldsymbol{x_i} \in S_{min}$:

$$g_i = \Gamma_i \times G. \tag{5}$$

Finally, for each $\boldsymbol{x_i} \in S_{min}$, generate g_i synthetic data samples according to (1). The key idea of the ADASYN algorithm is to use a density distribution Γ as a criterion to automatically decide the number of synthetic samples that need to be generated for each minority example by adaptively changing the weights of different minority examples to compensate for the skewed distributions.

3.1.5 Sampling with Data Cleaning Techniques

Data cleaning techniques, such as Tomek links, have been effectively applied to remove the overlapping that is introduced from sampling methods. Generally speaking, Tomek links [46] can be defined as a pair of minimally distanced nearest neighbors of opposite classes. Given an instance pair: $(\boldsymbol{x_i}, \boldsymbol{x_j})$, where $\boldsymbol{x_i} \in S_{min}, \boldsymbol{x_j} \in S_{maj}$, and $d(\boldsymbol{x_i}, \boldsymbol{x_j})$ is the distance between $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, then the $(\boldsymbol{x_i}, \boldsymbol{x_j})$ pair is called a Tomek link if there is no instance $\boldsymbol{x_k}$, such that $d(\boldsymbol{x_i}, \boldsymbol{x_k}) < d(\boldsymbol{x_i}, \boldsymbol{x_j})$ or $d(\boldsymbol{x_j}, \boldsymbol{x_k}) < d(\boldsymbol{x_i}, \boldsymbol{x_j})$. In this way, if two instances form a Tomek link then either one of these instances is noise or both are near a border. Therefore, one can use Tomek links to "cleanup" unwanted overlapping between classes after synthetic sampling where all Tomek links are removed until all minimally distanced nearest neighbor pairs are of the same class. By removing overlapping examples, one can establish well-defined class clusters in the training set, which can, in turn, lead to well-defined classification rules for improved classification performance. Some representative work in this area includes the OSS method [42], the condensed nearest neighbor rule and Tomek Links (CNN+Tomek Links) integration method [22], the neighborhood cleaning rule (NCL) [36] based on the edited nearest neighbor (ENN) rule—which removes examples that differ from two of its three nearest neighbors, and the integrations of SMOTE with ENN (SMOTE+ENN) and SMOTE with Tomek links (SMOTE+Tomek) [22].

Fig. 5 shows a typical procedure of using SMOTE and Tomek to clean the overlapping data points.

Fig. 5a shows the original data set distribution for an artificial imbalanced data set; note the inherent overlapping that exists between the minority and majority examples.

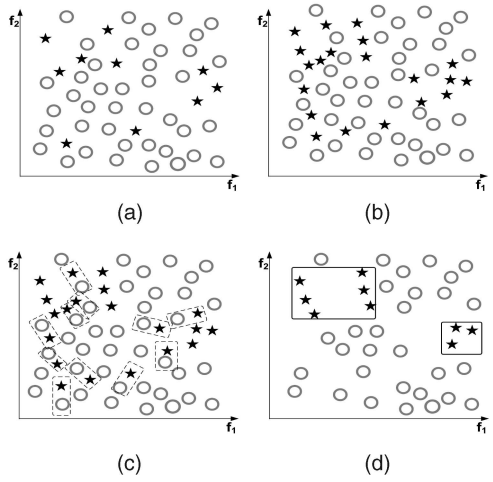


Fig. 5. (a) Original data set distribution. (b) Post-SMOTE data set. (c) The identified Tomek Links. (d) The data set after removing Tomek links.

Fig. 5b shows the data set distribution after synthetic sampling by SMOTE. As can be seen, there is an increased amount of overlapping introduced by SMOTE. In Fig. 5c, the Tomek links are identified, which are represented by the dashed boxes. Last, Fig. 5d shows the data set after cleanup is performed. We can see that the algorithm produces more well-defined class clusters, which potentially contributes to improved classification performance. Furthermore, the idea illustrated in Fig. 5 is important since it introduces a consideration for class clusters; we further investigate class clusters in the following discussion of the cluster-based sampling algorithm.

3.1.6 Cluster-Based Sampling Method

Cluster-based sampling algorithms are particularly interesting because they provide an added element of flexibility that is not available in most simple and synthetic sampling algorithms, and accordingly can be tailored to target very specific problems. In [27], the cluster-based oversampling (CBO) algorithm is proposed to effectively deal with the within-class imbalance problem in tandem with the between-class imbalance problem.

The CBO algorithm makes use of the K-means clustering technique. This procedure takes a random set of K examples from each cluster (for both classes) and computes the mean feature vector of these examples, which is designated as the cluster center. Next, the remaining training examples are presented one at a time and for each example, the euclidean distance vector between it and each cluster center is computed. Each training example is then assigned to the cluster that exhibits the smallest distance vector magnitude. Lastly, all cluster means are updated and the process is repeated until all examples are exhausted (i.e., only one cluster mean is essentially updated for each example).

Fig. 6 illustrates these steps. Fig. 6a shows the original distribution. Here, the majority class has three clusters A, B, and C ($m_{maj} = 3$), and each of the clusters has 20, 10, and 8 examples, respectively. The minority class has two clusters, D and E ($m_{min} = 2$), each with eight and five examples, respectively. Fig. 6b shows the cluster means

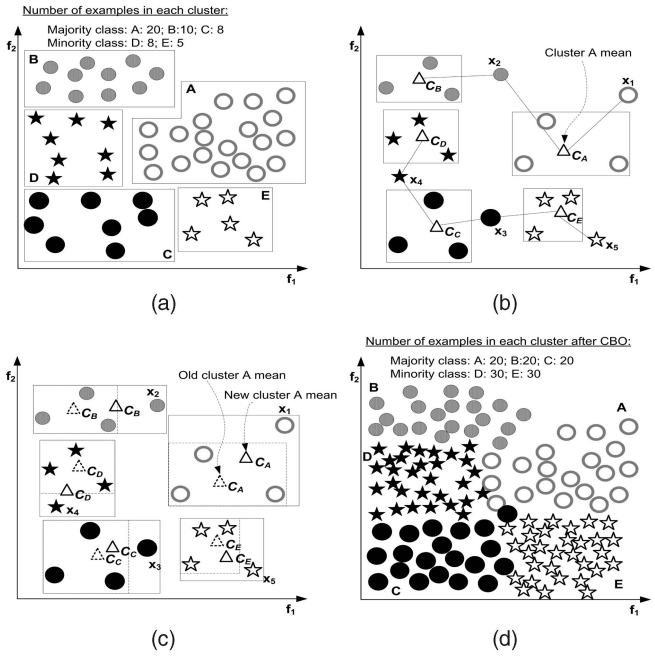


Fig. 6. (a) Original data set distribution. (b) Distance vectors of examples and cluster means. (c) Newly defined cluster means and cluster borders. (d) The data set after cluster-based oversampling method.

(represented by triangles) for three random examples of each cluster, i.e., $k = 3$. Fig. 6b also shows the distance vectors for the five individually introduced examples $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4$, and $\boldsymbol{x}_5$. Fig. 6c shows the updated cluster means and cluster borders as a result of the five introduced examples. Once all examples are exhausted, the CBO algorithm inflates all majority class clusters other than the largest by oversampling so that all majority class clusters are of the same size as the largest (i.e., clusters B and C will each have 20 examples). We denote the total number of majority class examples after the oversampling process as $N_{CBO}, N_{CBO} = |S_{maj}| + |E_{maj}|$ (e.g., $N_{CBO} = 60$ in our example). Then, we oversample the minority clusters so that each cluster contains N_{CBO}/m_{min} total examples (i.e., each minority clusters D and E will have a total number of $60/2 = 30$ examples after the oversampling procedure). Fig. 6d shows the final data set after applying the CBO method. Compared to Fig. 6a, we can see that the final data set has a stronger representation of rare concepts. We would also like to note that different oversampling methods can be integrated into the CBO algorithm. For instance, Jo and Japkowicz [27] used the random oversampling method discussed in Section 3.1.1, while our example in Fig. 6 uses synthetic sampling. Empirical results of CBO are very suggestive into the nature of the imbalanced learning problem; namely, that targeting *within-class imbalance* in tandem with the *between-class imbalance* is an effective strategy for imbalanced data sets.

3.1.7 Integration of Sampling and Boosting

The integration of sampling strategies with ensemble learning techniques has also been studied in the community. For instance, the SMOTEBoost [47] algorithm is based

on the idea of integrating SMOTE with Adaboost.M2. Specifically, SMOTEBoost introduces synthetic sampling at each boosting iteration. In this way, each successive classifier ensemble focuses more on the minority class. Since each classifier ensemble is built on a different sampling of data, the final voted classifier is expected to have a broadened and well-defined decision region for the minority class.

Another integrated approach, the DataBoost-IM [14] method, combines the data generation techniques introduced in [48] with AdaBoost.M1 to achieve high predictive accuracy for the minority class without sacrificing accuracy on the majority class. Briefly, DataBoost-IM generates synthetic samples according to the ratio of difficult-to-learn samples between classes. Concretely, for a data set S with corresponding subsets $S_{min} \subset S$ and $S_{maj} \subset S$, and weighted distribution D_t representing the relative difficulty of learning for each example $\boldsymbol{x_i} \in S$, we rank $\boldsymbol{x_i}$ in descending order according to their respective weight. We then select the top $|S| \times error(t)$ examples to populate set $E, E \subset S$, where $error(t)$ is the error rate of the current learned classifier. Thus, E is a collection of the hard-to-learn (hard) samples from both classes and has subsets $E_{min} \subset E$ and $E_{maj} \subset E$. Moreover, since minority class samples are generally more difficult to learn than majority class samples, it is expected that $|E_{maj}| \leq |E_{min}|$.

Once the difficult examples are identified, DataBoost-IM proceeds to create synthetic samples according to a two-tier process: first, identify the "seeds" of E from which synthetic samples are formed, and then, generate synthetic data based on these samples. The seed identification procedure is based on the ratio of class representation in E and S. The number of majority class seeds M_L is defined as $M_L = min(\frac{|S_{maj}|}{|S_{min}|}, |E_{maj}|)$, and the number of minority seeds M_S is defined as $M_S = min(\frac{|S_{maj}| \times M_L}{|S_{min}|}, |E_{min}|)$. We then proceed to generate synthetic set E_{syn}, with subsets $E_{smin} \subset E_{syn}$ and $E_{smaj} \subset E_{syn}$, such that $|E_{smin}| = M_S \times |S_{min}|$ and $|E_{smaj}| = M_L \times |S_{maj}|$. Set S is then augmented by E_{syn} to provide a more balanced class distribution with more new instances of the minority class. Lastly, the weighted distribution D_t is updated with consideration to the newly added synthetic samples.

Evidence that synthetic sampling methods are effective in dealing with learning from imbalanced data is quite strong. However, the data generation methods discussed thus far are complex and computationally expensive. Noting the essential problem of "ties" in random oversampling as discussed in Section 3.1.1, Mease et al. [38] propose a much simpler technique for breaking these ties: instead of generating new data from computational methods, use the duplicate data obtained from random oversampling and introduce perturbations ("jittering") to this data to break ties. The resulting algorithm, over/undersampling with jittering (JOUS-Boost), introduces independently and identically distributed (iid) noise at each iteration of boosting to minority examples for which oversampling creates replicates [38]. This idea is relatively simple compared to its synthetic sampling counterparts and also incorporates the benefits of boosted ensembles to improve performance. It was shown to provide very efficient results in empirical studies, which suggests that synthetic procedures can be successful without jeopardizing runtime costs.

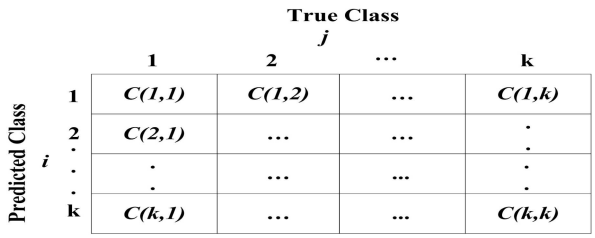


Fig. 7. Multiclass cost matrix.

3.2 Cost-Sensitive Methods\for Imbalanced Learning

While sampling methods attempt to balance distributions by considering the representative proportions of class examples in the distribution, cost-sensitive learning methods consider the costs associated with misclassifying examples [49], [50]. Instead of creating balanced data distributions through different sampling strategies, cost-sensitive learning targets the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying any particular data example. Recent research indicates that there is a strong connection between cost-sensitive learning and learning from imbalanced data; therefore, the theoretical foundations and algorithms of cost-sensitive methods can be naturally applied to imbalanced learning problems [3], [20], [51]. Moreover, various empirical studies have shown that in some application domains, including certain specific imbalanced learning domains [11], [52], [53], cost-sensitive learning is superior to sampling methods. Therefore, cost-sensitive techniques provide a viable alternative to sampling methods for imbalanced learning domains.

3.2.1 Cost-Sensitive Learning Framework

Fundamental to the cost-sensitive learning methodology is the concept of the cost matrix. The cost matrix can be considered as a numerical representation of the penalty of classifying examples from one class to another. For example, in a binary classification scenario, we define $C(Min, Maj)$ as the cost of misclassifying a majority class example as a minority class example and let $C(Maj, Min)$ represents the cost of the contrary case. Typically, there is no cost for correct classification of either class and the cost of misclassifying minority examples is higher than the contrary case, i.e., $C(Maj, Min) > C(Min, Maj)$. The objective of cost-sensitive learning then is to develop a hypothesis that minimizes the overall cost on the training data set, which is usually the *Bayes conditional risk*. These concepts are easily extended to multiclass data by considering $C(i, j)$ which represents the cost of predicting class i when the true class is j, where $i, j \in Y = \{1, \ldots, C\}$. Fig. 7 shows a typical cost matrix for a multiclass problem. In this case, the conditional risk is defined as $R(i|\boldsymbol{x}) = \sum_j P(j|\boldsymbol{x})C(i, j)$,

where $P(j|\boldsymbol{x})$ represents the probability of each class j for a given example $\boldsymbol{x}$ [49], [54].

There are many different ways of implementing cost-sensitive learning, but, in general, the majority of techniques fall under three categories. The first class of techniques apply misclassification costs to the data set as a form of dataspace weighting; these techniques are essentially cost-sensitive bootstrap sampling approaches where misclassification costs are used to select the best training distribution for induction. The second class applies cost-minimizing techniques to the combination schemes of ensemble methods; this class consists of various Metatechniques where standard learning algorithms are integrated with ensemble methods to develop cost-sensitive classifiers. Both of these classes have rich theoretical foundations that justify their approaches, with cost-sensitive dataspace weighting methods building on the *translation theorem* [55], and cost-sensitive Metatechniques building on the *Metacost framework* [54]. In fact, many of the existing research works often integrate the Metacost framework with dataspace weighting and adaptive boosting to achieve stronger classification results. To this end, we consider both of these classes of algorithms as one in the following section. The last class of techniques incorporates cost-sensitive functions or features directly into classification paradigms to essentially "fit" the cost-sensitive framework into these classifiers. Because many of these techniques are specific to a particular paradigm, there is no unifying framework for this class of cost-sensitive learning, but in many cases, solutions that work for one paradigm can often be abstracted to work for others. As such, in our discussion of these types of techniques, we consider a few methods on a case-specific basis.

3.2.2 Cost-Sensitive Dataspace Weighting with Adaptive Boosting

Motivated by the pioneering work of the AdaBoost algorithms [56], [57], several cost-sensitive boosting methods for imbalanced learning have been proposed. Three cost-sensitive boosting methods, AdaC1, AdaC2, and AdaC3, were proposed in [58] which introduce cost items into the weight updating strategy of AdaBoost. The key idea of the AdaBoost.M1 method is to iteratively update the distribution function over the training data. In this way, on each iteration $t := 1, \ldots, T$, where T is a preset number of the total number of iterations, the distribution function D_t is updated sequentially and used to train a new hypothesis:

$$D_{t+1}(i) = D_t(i) \exp(-\alpha_t h_t(\boldsymbol{x_i}) y_i)/Z_t, \tag{6}$$

where $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$ is the weight updating parameter, $h_t(\boldsymbol{x_i})$ is the prediction output of hypothesis h_t on the instance $\boldsymbol{x_i}$, ε_t is the error of hypothesis h_t over the training data $\varepsilon_t = \sum_{i:h_t(\boldsymbol{x_i}) \neq y_i} D_t(i)$, and Z_t is a normalization factor so that D_{t+1} is a distribution function, i.e., $\sum_{i=1}^{m} D_{t+1}(i) = 1$. With this description in mind, a cost factor can be applied in three ways, inside of the exponential, outside of the exponential, and both inside and outside the exponential. Analytically, this translates to

$$D_{t+1}(i) = D_t(i) \exp(-\alpha_t C_i h_t(\boldsymbol{x_i}) y_i)/Z_t, \tag{7}$$

$$D_{t+1}(i) = C_i D_t(i) \exp(-\alpha_t h_t(\boldsymbol{x_i}) y_i)/Z_t, \tag{8}$$

$$D_{t+1}(i) = C_i D_t(i) \exp(-\alpha_t C_i h_t(\boldsymbol{x_i}) y_i)/Z_t. \tag{9}$$

Equations (7), (8), and (9) corresponds to the AdaC1, AdaC2, and AdaC3 methods, respectively. Here, the cost item C_i is the associated cost for each $\boldsymbol{x_i}$ and C_is of higher value correspond to examples with higher misclassification costs. In essence, these algorithms increase the probability of sampling a costly example at each iteration, giving the classifier more instances of costly examples for a more targeted approach of induction. In general, it was observed that the inclusion of cost factors into the weighting scheme of Adaboost imposes a bias toward the minority concepts and also increases the use of more relevant data samples in each hypothesis, providing for a more robust form of classification.

Another cost-sensitive boosting algorithm that follows a similar methodology is AdaCost [59]. AdaCost, like AdaC1, introduces cost sensitivity inside the exponent of the weight updating formula of Adaboost. However, instead of applying the cost items directly, AdaCost uses a cost-adjustment function that aggressively increases the weights of costly misclassifications and conservatively decreases the weights of high-cost examples that are correctly classified. This modification becomes:

$$D_{t+1}(i) = D_t(i) \exp(-\alpha_t h_t(\boldsymbol{x_i}) y_i \beta_i)/Z_t, \tag{10}$$

with the cost-adjustment function β_i, defined as $\beta_i = \beta(sign(y_i, h_t(\boldsymbol{x_i})), C_i)$, where $sign(y_i, h_t(\boldsymbol{x_i}))$ is positive for correct classification and negative for misclassification. For clear presentation, one can use β_+ when $sign(y_i, h_t(\boldsymbol{x_i})) = 1$ and β_- when $sign(y_i, h_t(\boldsymbol{x_i})) = -1$. This method also allows some flexibility in the amount of emphasis given to the importance of an example. For instance, Fan et al. [59] suggest $\beta_+ = -0.5C_i + 0.5$ and $\beta_- = 0.5C_i + 0.5$ for good results in most applications, but these coefficients can be adjusted according to specific needs. An empirical comparison over four imbalanced data sets of AdaC1, AdaC2, AdaC3, and AdaCost and two other similar algorithms, CSB1 and CSB2 [60], was performed in [58] using decision trees and a rule association system as the base classifiers. It was noted that in all cases, a boosted ensemble performed better than the stand-alone base classifiers using F-measure (see Section 4.1) as the evaluation metric, and in nearly all cases, the cost-sensitive boosted ensembles performed better than plain boosting.

Though these cost-sensitive algorithms can significantly improve classification performance, they take for granted the availability of a cost matrix and its associated cost items. In many situations, an explicit description of misclassification costs is unknown, i.e., only an informal assertion is known, such as *misclassifications on the positive class are more expensive than the negative class* [51]. Moreover, determining a cost representation of a given domain can be particularly challenging and in some cases impossible [61]. As a result, the techniques discussed in this section are not applicable in these situations and other solutions must be established. This is the prime motivation for the cost-sensitive fitting techniques mentioned earlier. In the following sections, we provide an overview of these methods for two popular learning paradigms, namely, decision trees and neural networks.

3.2.3 Cost-Sensitive Decision Trees

In regards to decision trees, cost-sensitive fitting can take three forms: first, cost-sensitive adjustments can be applied to the decision threshold; second, cost-sensitive considerations can be given to the split criteria at each node; and lastly, cost-sensitive pruning schemes can be applied to the tree.

A decision tree threshold moving scheme for imbalanced data with unknown misclassification costs was observed in [51]. The relationships between the misclassification costs of each class, the distribution of training examples, and the placement of the decision threshold have been established in [62]. However, Maloof [51] notes that the precise definition of these relationships can be task-specific, rendering a systematic approach for threshold selection based on these relationships unfeasible. Therefore, instead of relying on the training distribution or exact misclassification costs, the proposed technique uses the ROC evaluation procedure (see Section 4.2) to plot the range of performance values as the decision threshold is moved from the point where the total misclassifications on the positive class are maximally costly to the point where total misclassifications on the negative class are maximally costly. The decision threshold that yields the most dominant point on the ROC curve is then used as the final decision threshold.

When considering cost sensitivity in the split criterion, the task at hand is to fit an impurity function that is insensitive to unequal costs. For instance, traditionally accuracy is used as the impurity function for decision trees, which chooses the split with minimal error at each node. However, this metric is sensitive to changes in sample distributions (see Section 4.1), and thus, inherently sensitive to unequal costs. In [63], three specific impurity functions, Gini, Entropy, and DKM, were shown to have improved cost insensitivity compared with the accuracy/error rate baseline. Moreover, these empirical experiments also showed that using the DKM function generally produced smaller unpruned decision trees that at worse provided accuracies comparable to Gini and Entropy. A detailed theoretical basis explaining the conclusions of these empirical results was later established in [49], which generalizes the effects of decision tree growth for any choice of spit criteria.

The final case of cost-sensitive decision tree fitting applies to pruning. Pruning is beneficial for decision trees because it improves generalization by removing leaves with class probability estimates below a specified threshold. However, in the presence of imbalanced data, pruning procedures tend to remove leaves describing the minority concept. It has been shown that though pruning trees induced from imbalanced data can hinder performance, using unpruned trees in such cases does not improve performance [23]. As a result, attention has been given to improving the class probability estimate at each node to develop more representative decision tree structures such that pruning can be applied with positive effects. Some representative works include the Laplace smoothing method of the probability estimate and the Laplace pruning technique [49].

3.2.4 Cost-Sensitive Neural Networks

Cost-sensitive neural networks have also been widely studied in the community for imbalanced learning. The neural network is generally represented by a densely interconnected set of simple neurons. Most practical applications of the neural network classifier involve a multilayer structure, such as the popular multilayer perceptron (MLP) model [64], and learning is facilitated by using the back propagation algorithm in tandem with the gradient descent rule. Concretely, assume that one defines an error function as

$$E(\omega) = \frac{1}{2}\sum (t_k - o_k)^2, \tag{11}$$

where ω is a set of weights that require training, and t_k and o_k are the target value and network output value for a neuron k, respectively. The gradient descent rule aims to find the steepest descent to modify the weights at each iteration:

$$\Delta\omega_n = -\eta\nabla_\omega E(\omega_n), \tag{12}$$

where η is the specified neural network learning rate and ∇_ω represents the gradient operator with respect to weights ω. Moreover, a probabilistic estimate for the output can be defined by normalizing the output values of all output neurons.

With this framework at hand, cost sensitivity can be introduced to neural networks in four ways [65]: first, cost-sensitive modifications can be applied to the probabilistic estimate; second, the neural network outputs (i,e., each o_k) can be made cost-sensitive; third, cost-sensitive modifications can be applied to the learning rate η; and fourth, the error minimization function can be adapted to account for expected costs.

In regards to the probability estimate, Kukar and Kononenko [65] integrate cost factors into the testing stage of classification to adaptively modify the probability estimate of the neural network output. This has the benefit of maintaining the original structure (and outputs) of the neural network while strengthening the original estimates on the more expensive class through cost consideration. Empirical results in [65] showed that this technique improves the performance over the original neural network, although the improvement is not drastic. However, we note that a more significant performance increase can be achieved by applying this estimate to ensemble methods by using cross-validation techniques on a given set; a similar approach is considered in [11], however using a slightly different estimate.

The second class of neural network cost-sensitive fitting techniques directly changes the outputs of the neural network. In [65], the outputs of the neural network are altered during training to bias the neural network to focus more on the expensive class. Empirical results on this method showed an improvement in classification performance on average, but also showed a high degree of variance in the performance measures compared to the least expected cost over the evaluated data sets. We speculate that ensemble methods can be applied to alleviate this problem, but to our knowledge, such experiments have not been performed to date.

The learning rate η can also influence the weight adjustment (see (12)). As a result, cost-sensitive factors can be applied to the learning rate to change the impact that the modification procedure has on the weights, where costly examples will have a greater impact on weight changes. The key idea of this approach is to put more attention on costly examples during learning by effectively decreasing the learning rate for each corresponding costly example. This also suggests that low-cost examples will train at a faster rate than costly examples, so this method also strikes a balance in training time. Experiments on this technique have shown it to be very effective for training neural networks with significant improvements over the base classifier [65].

The final adaptation of cost-sensitive neural networks replaces the error-minimizing function shown in (11) by an expected cost minimization function. This form of cost-sensitive fitting was shown to be the most dominant of the methods discussed in this section [65]. It also is in line with the back propagation methodology and theoretic foundations established on the transitivity between error-minimizing and cost-minimizing classifiers.

Though we only provide a treatment for decision trees and neural networks, many cost-sensitive fitting techniques exist for other types of learning paradigms as well. For instance, a great deal of works have focused on cost-sensitive Bayesian Classifiers [66], [67], [68], [69] and some works exist which integrate cost functions with support vector machines [70], [71], [72]. Interested readers can refer to these works for a broader overview.

3.3 Kernel-Based Methods and Active Learning Methods for Imbalanced Learning

Although sampling methods and cost-sensitive learning methods seem to dominate the current research efforts in imbalanced learning, numerous other approaches have also been pursued in the community. In this section, we briefly review kernel-based learning methods and active learning methods for imbalanced learning. Since kernel-based learning methods provide state-of-the-art techniques for many of today's data engineering applications, the use of kernel-based methods to understand imbalanced learning has naturally attracted growing attention recently.

3.3.1 Kernel-Based Learning Framework

The principles of kernel-based learning are centered on the theories of statistical learning and Vapnik-Chervonenkis (VC) dimensions [73]. The representative kernel-based learning paradigm, support vector machines (SVMs), can provide relatively robust classification results when applied to imbalanced data sets [23]. SVMs facilitate learning by using specific examples near concept boundaries (support vectors) to maximize the separation margin (soft-margin maximization) between the support vectors and the hypothesized concept boundary (hyperplane), meanwhile minimizing the total classification error [73].

The effects of imbalanced data on SVMs exploit inadequacies of the soft-margin maximization paradigm [74], [75]. Since SVMs try to minimize total error, they are inherently biased toward the majority concept. In the simplest case, a two-class space is linearly separated by an "ideal" separation line in the neighborhood of the majority concept. In this case, it might occur that the support vectors representing the minority concept are "far away" from this "ideal" line, and as a result, will contribute less to the final hypothesis [74], [75], [76]. Moreover, if there is a lack of data representing the minority concept, there could be an imbalance of representative support vectors that can also degrade performance. These same characteristics are also readily evident in linear nonseparable spaces. In this case, a kernel function is used to map the linear nonseparable space into a higher dimensional space where separation is achievable. However, in this case, the optimal hyperplane separating the classes will be biased toward the majority class in order to minimize the high error rates of misclassifying the more prevalent majority class. In the worst case, SVMs will learn to classify all examples as pertaining to the majority class—a tactic that, if the imbalance is severe, can provide the minimal error rate across the dataspace.

3.3.2 Integration of Kernel Methods with Sampling Methods

There have been many works in the community that apply general sampling and ensemble techniques to the SVM framework. Some examples include the SMOTE with Different Costs (SDCs) method [75] and the ensembles of over/undersampled SVMs [77], [78], [79], [80]. For example, the SDC algorithm uses different error costs [75] for different classes to bias the SVM in order to shift the decision boundary away from positive instances and make positive instances more densely distributed in an attempt to guarantee a more well-defined boundary. Meanwhile, the methods proposed in [78], [79] develop ensemble systems by modifying the data distributions without modifying the underlying SVM classifier. Lastly, Wang and Japkowicz [80] proposed to modify the SVMs with asymmetric misclassification costs in order to boost performance. This idea is similar to the AdaBoost.M1 [56], [57] algorithm in that it uses an iterative procedure to effectively modify the weights of the training observations. In this way, one can build a modified version of the training data based on such sequential learning procedures to improve classification performance.

The Granular Support Vector Machines—Repetitive Undersampling algorithm (GSVM-RU) was proposed in [81] to integrate SVM learning with undersampling methods. This method is based on granular support vector machines (GSVMs) which were developed in a series of papers according to the principles from statistical learning theory and granular computing theory [82], [83], [84]. The major characteristics of GSVMs are two-fold. First, GSVMs can effectively analyze the inherent data distribution by observing the trade-offs between the local significance of a subset of data and its global correlation. Second, GSVMs improve the computational efficiency of SVMs through use of parallel computation. In the context of imbalanced learning, the GSVM-RU method takes advantage of the GSVM by using an iterative learning procedure that uses the SVM itself for undersampling [81]. Concretely, since all minority (positive) examples are considered to be informative, a positive information granule is formed from these examples. Then, a linear SVM is developed using the positive granule and the

remaining examples in the data set (i.e., S_{maj}); the negative examples that are identified as support vectors by this SVM, the so-called "negative local support vectors" (NLSVs), are formed into a negative information granule and are removed from the original training data to obtain a smaller training data set. Based on this reduced training data set, a new linear SVM is developed, and again, the new set of NLSVs is formed into a negative granule and removed from the data set. This procedure is repeated multiple times to obtain multiple negative information granules. Finally, an aggregation operation that considers global correlation is used to select specific sample sets from those iteratively developed negative information granules, which are then combined with all positive samples to develop a final SVM model. In this way, the GSVM-RU method uses the SVM itself as a mechanism for undersampling to sequentially develop multiple information granules with different informative samples, which are later combined to develop a final SVM for classification.

3.3.3 Kernel Modification Methods for Imbalanced Learning

In addition to the aforementioned sampling and ensemble kernel-based learning methods, another major category of kernel-based learning research efforts focuses more concretely on the mechanics of the SVM itself; this group of methods is often called kernel modification method.

One example of kernel modification is the kernel classifier construction algorithm proposed in [85] based on orthogonal forward selection (OFS) and the regularized orthogonal weighted least squares (ROWLSs) estimator. This algorithm optimizes generalization in the kernel-based learning model by introducing two major components that deal with imbalanced data distributions for two-class data sets. The first component integrates the concepts of leave-one-out (LOO) cross validation and the area under curve (AUC) evaluation metric (see Section 4.2) to develop an LOO-AUC objective function as a selection mechanism of the most optimal kernel model. The second component takes advantage of the cost sensitivity of the parameter estimation cost function in the ROWLS algorithm to assign greater weight to erroneous data examples in the minority class than those in the majority class.

Other examples of kernel modification are the various techniques for adjusting the SVM class boundary. These methods apply boundary alignment techniques to improve SVM classification [76], [86], [87]. For instance, in [76], three algorithmic approaches for adjusting boundary skews were presented: the boundary movement (BM) approach, the biased penalties (BPs) approach, and the class-boundary alignment (CBA) approach. Additionally, in [86] and [87], the kernel-boundary alignment (KBA) algorithm was proposed which is based on the idea of modifying the kernel matrix generated by a kernel function according to the imbalanced data distribution. The underlying theoretical foundation of the KBA method builds on the adaptive conformal transformation (ACT) methodology, where the conformal transformation on a kernel function is based on the consideration of the feature-space distance and the class-imbalance ratio [88]. By generalizing the foundation of ACT, the KBA method tackles the imbalanced learning problem by modifying the kernel matrix in the feature space. Theoretical analyses and empirical studies showed that this method not only provides competitive accuracy, but it can also be applied to both vector data and sequence data by modifying the kernel matrix.

In a more integrated approach of kernel based learning, Liu and Chen [89] and [90] propose the total margin-based adaptive fuzzy SVM kernel method (TAF-SVM) to improve SVM robustness. The major beneficial characteristics of TAF-SVM are three-fold. First, TAF-SVM can handle overfitting by "fuzzifying" the training data, where certain training examples are treated differently according to their relative importance. Second, different cost algorithms are embedded into TAF-SVM, which allows this algorithm to self-adapt to different data distribution skews. Last, the conventional soft-margin maximization paradigm is replaced by the total margin paradigm, which considers both the misclassified and correctly classified data examples in the construction of the optimal separating hyperplane.

A particularly interesting kernel modification method for imbalanced learning is the k-category proximal support vector machine (PSVM) with Newton refinement [91]. This method essentially transforms the soft-margin maximization paradigm into a simple system of k linear equations for either linear or nonlinear classifiers, where k is the number of classes. One of the major advantages of this method is that it can perform the learning procedure very fast because this method requires nothing more sophisticated than solving this simple system of linear equations. Lastly, in the presence of extremely imbalanced data sets, Raskutti and Kowalcyzk [74] consider both sampling and dataspace weighting compensation techniques in cases where SVMs completely ignore one of the classes. In this procedure, two balancing modes are used in order to balance the data: a *similarity detector* is used to learn a discriminator based predominantly on positive examples, and a *novelty detector* is used to learn a discriminator using primarily negative examples.

Several other kernel modification methods exist in the community including the support cluster machines (SCMs) for large-scale imbalanced data sets [92], the kernel neural gas (KNG) algorithm for imbalanced clustering [93], the P2PKNNC algorithm based on the k-nearest neighbors classifier and the P2P communication paradigm [94], the hybrid kernel machine ensemble (HKME) algorithm including a binary support vector classifier (BSVC) and a one-class support vector classifier ($\nu - SVC$) with Gaussian radial basis kernel function [95], and the Adaboost relevance vector machine (RVM) [96], amongst others. Furthermore, we would like to note that for many kernel-based learning methods, there is no strict distinction between the aforementioned two major categories of Sections 3.3.2 and 3.3.3. In many situations, learning methods take a hybrid approach where sampling and ensemble techniques are integrated with kernel modification methods for improved performance. For instance, [75] and [76] are good examples of hybrid solutions for imbalanced learning. In this section, we categorize kernel-based learning in two sections for better presentation and organization.

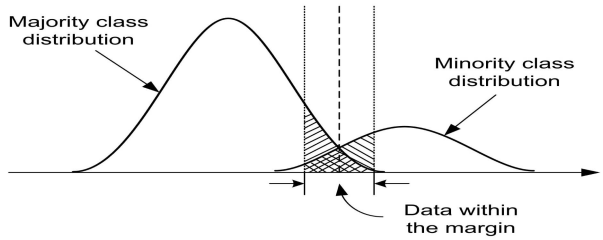


Fig. 8. Data imbalance ratio within and outside the margin [98].

3.3.4 Active Learning Methods for Imbalanced Learning

Active learning methods have also been investigated in the community for imbalanced learning problems. Traditionally, active learning methods are used to solve problems related to unlabeled training data. Recently, however, various issues on active learning from imbalanced data sets have been discussed in literature [97], [98], [99], [100]. Moreover, we point out that active learning approaches for imbalanced learning are often integrated into kernel-based learning methods; as a result, we discuss both methods in the same light.

SVM-based active learning aims to select the most informative instances from the unseen training data in order to retrain the kernel-based model [99], i.e., those instances that are closest to the current hyperplane. Fig. 8 illustrates the motivation for the selection procedure for imbalanced data sets [98]. Assume that Fig. 8 represents the class distribution of an imbalanced data set, where the shaded region corresponds to the class distribution within the margin. In this case, the imbalance ratio of data within the margin is much smaller than the imbalance ratio of the entire data set. Motivated by this observation, Ertekin et al. [98] and [99] proposed an efficient SVM-based active learning method which queries a small pool of data at each iterative step of active learning instead of querying the entire data set. In this procedure, an SVM is trained on the given training data, after which the most informative instances are extracted and formed into a new training set according to the developed hyperplane. Finally, the procedure uses this new training set and all unseen training data to actively retrain the SVM using the LASVM online SVM learning algorithm [101] to facilitate the active learning procedure.

Ertekin et al. [98] and [99] also point out that the search process for the most informative instances can be computationally expensive because, for each instance of unseen data, the algorithm needs to recalculate the distance between each instance and the current hyperplane. To solve this problem, they proposed a method to effectively select such informative instances from a random set of training populations to reduce the computational cost for large-scale imbalanced data sets [98], [99]. Additionally, early stopping criteria for active learning are also discussed in this work which can be used to achieve faster convergence of the active learning process as compared to the random sample selection solution.

In addition to kernel-based integrations, active learning integrations with sampling techniques have also been investigated in the community. For instance, Zhu and Hovy [102] analyzed the effect of undersampling and oversampling techniques with active learning for the word sense disambiguation (WSD) imbalanced learning problem. The active learning method studied in this work is based on the uncertainty sampling methodology; here, the challenge is how to measure the uncertainty of an unlabeled instance in order to select the maximally uncertain instance to augment the training data. In this case, Entropy was used as a metric for determining uncertainty. Additionally, two stopping mechanisms based on maximum confidence and minimal error were investigated in [102]. Simulation results concluded that one can use max-confidence as the upper bound and min-error as the lower bound of the stopping conditions for active learning in this case. Another active learning sampling method is the simple active learning heuristic (SALH) approach proposed in [103]. The key idea of this method is to provide a generic model for the evolution of genetic programming (GP) classifiers by integrating the stochastic subsampling method and a modified Wilcoxon-Mann-Whitney (WMW) cost function [103]. The major advantages of the SALH method include the ability to actively bias the data distribution for learning, the existence of a robust cost function, and the improvement of the computational cost related to the fitness evaluation. Simulation results over six data sets were used to illustrate the effectiveness of this method.

3.4 Additional Methods for Imbalanced Learning

In closing our review of the state-of-the-art solutions for imbalanced learning, we would like to note that community solutions to handle the imbalanced learning problem are not solely in the form of sampling methods, cost-sensitive methods, kernel-based methods, and active learning methods. For instance, the one-class learning or novelty detection methods have also attracted much attention in the community [3]. Generally speaking, this category of approaches aims to recognize instances of a concept by using mainly, or only, a single class of examples (i.e., recognition-based methodology) rather than differentiating between instances of both positive and negative classes as in the conventional learning approaches (i.e., discrimination-based inductive methodology). Representative works in this area include the one-class SVMs [74], [104], [105], [106], [107], [108] and the autoassociator (or autoencoder) method [109], [110], [111], [112]. Specifically, Raskutti and Kowalcyzk [74] suggested that one-class learning is particularly useful in dealing with extremely imbalanced data sets with high feature space dimensionality. Additionally, Japkowicz [109] proposed an approach to train an autoassociator to reconstruct the positive class at the output layer, and it was suggested that under certain conditions, such as in multimodal domains, the one-class learning approach may be superior to the discrimination-based approaches. Meanwhile, Manevitz and Yousef [105] and [110] presented the successful applications of the one-class learning approach to the document classification domain based on SVMs and autoencoder, respectively. In [111], a comparison of different sampling methods and the one-class autoassociator method was presented, which provides useful suggestions about the advantages and limitations of both methods. The novelty detection approach based on redundancy

compression and nonredundancy differentiation techniques was investigated in [112]. Recently, Lee and Cho [107] suggested that novelty detection methods are particularly useful for extremely imbalanced data sets, while regular discrimination-based inductive classifiers are suitable for a relatively moderate imbalanced data sets.

Recently, the Mahalanobis-Taguchi System (MTS) has also been used for imbalanced learning [113]. The MTS was originally developed as a diagnostic and forecasting technique for multivariate data [114], [115]. Unlike most of the classification paradigms, learning in the MTS is performed by developing a continuous measurement scale using single-class examples instead of the entire training data. Because of its characteristics, it is expected that the MTS model will not be influenced by the skewed data distribution, therefore providing robust classification performance. Motivated by these observations, Su and Hsiao [113] presented an evaluation of the MTS model for imbalanced learning with comparisons to stepwise discriminate analysis (SDA), back-propagation neural networks, decision trees, and SVMs. This work showed the effectiveness of the MTS in the presence of imbalanced data. Moreover, Su and Hsiao [113] also present a probabilistic thresholding method based on the Chebyshev's theorem to systematically determine an appropriate threshold for MTS classification.

Another important example relates to the combination of imbalanced data and the small sample size problem, as discussed in Section 2. Two major approaches were proposed in [31] to address this issue. First, rank metrics were proposed as the training and model selection criteria instead of the traditional accuracy metric. Rank metrics helps facilitate learning from imbalanced data with small sample sizes and high dimensionality by placing a greater emphasis on learning to distinguish classes themselves instead of the internal structure (feature space conjunctions) of classes. The second approach is based on the multitask learning methodology. The idea here is to use a shared representation of the data to train extra task models related to the main task, therefore amplifying the effective size of the underrepresented class by adding extra training information to the data [31].

Finally, we would also like to note that although the current efforts in the community are focused on two-class imbalanced problems, multiclass imbalanced learning problems exist and are of equal importance. For instance, in [7], a cost-sensitive boosting algorithm AdaC2.M1 was proposed to tackle the class imbalance problem with multiple classes. In this paper, a genetic algorithm was used to search the optimum cost setup for each class. In [8], an iterative method for multiclass cost-sensitive learning was proposed based on three key ideas: iterative cost weighting, dataspace expansion, and gradient boosting with stochastic ensembles. In [9], a min-max modular network was proposed to decompose a multiclass imbalanced learning problem into a series of small two-class subproblems. Other works of multiclass imbalanced learning include the rescaling approach for multiclass cost-sensitive neural networks [10], [11], the ensemble knowledge for imbalance sample sets (eKISS) method [12], and others.

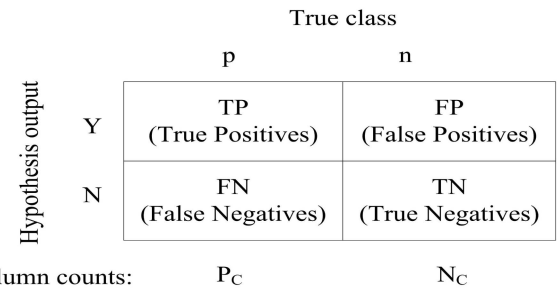


Fig. 9. Confusion matrix for performance evaluation.

As is evident, the range of existing solutions to the imbalanced learning problem is both multifaceted and well associated. Consequently, the assessment techniques used to evaluate these solutions share similar characteristics. We now turn our attention to these techniques.

4 Assessment Metrics for Imbalanced Learning

As the research community continues to develop a greater number of intricate and promising imbalanced learning algorithms, it becomes paramount to have standardized evaluation metrics to properly assess the effectiveness of such algorithms. In this section, we provide a critical review of the major assessment metrics for imbalanced learning.

4.1 Singular Assessment Metrics

Traditionally, the most frequently used metrics are *accuracy* and *error rate*. Considering a basic two-class classification problem, let $\{p, n\}$ be the true positive and negative class label and $\{Y, N\}$ be the predicted positive and negative class labels. Then, a representation of classification performance can be formulated by a *confusion matrix* (contingency table), as illustrated in Fig. 9.

In this paper, we use the minority class as the positive class and the majority class as the negative class. Following this convention, accuracy and error rate are defined as

$$Accuracy = \frac{TP + TN}{P_C + N_C}; \quad ErrorRate = 1 - accuracy. \quad (13)$$

These metrics provide a simple way of describing a classifier's performance on a given data set. However, they can be deceiving in certain situations and are highly sensitive to changes in data. In the simplest situation, if a given data set includes 5 percent of minority class examples and 95 percent of majority examples, a naive approach of classifying every example to be a majority class example would provide an accuracy of 95 percent. Taken at face value, 95 percent accuracy across the entire data set appears superb; however, on the same token, this description fails to reflect the fact that 0 percent of minority examples are identified. That is to say, the accuracy metric in this case does not provide adequate information on a classifier's functionality with respect to the type of classification required.

Many representative works on the ineffectiveness of accuracy in the imbalanced learning scenario exist in the community [14], [20], [47], [51], [58], [116], [117], [118]. The fundamental issue can be explained by evaluating the

confusion matrix in Fig. 9: The left column represents positive instances of the data set and the right column represents the negative instances. Therefore, the proportion of the two columns is representative of the class distribution of the data set, and any metric that uses values from both columns will be inherently sensitive to imbalances. As we can see from (13), *accuracy* uses both columns' information; therefore, as class distribution varies, measures of the performance will change even though the underlying fundamental performance of the classifier does not. As one can imagine, this can be very problematic when comparing the performance of different learning algorithms over different data sets because of the inconsistency of performance representation. In other words, in the presence of imbalanced data, it becomes difficult to make relative analysis when the evaluation metrics are sensitive to data distributions.

In lieu of accuracy, other evaluation metrics are frequently adopted in the research community to provide comprehensive assessments of imbalanced learning problems, namely, $precision, recall, F\text{-}measure$, and $G\text{-}mean$. These metrics are defined as:

$$Precision = \frac{TP}{TP + FP}, \tag{14}$$

$$Recall = \frac{TP}{TP + FN}, \tag{15}$$

$$F\text{-}Measure = \frac{(1+\beta)^2 \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision}, \tag{16}$$

where β is a coefficient to adjust the relative importance of precision versus recall (usually, $\beta = 1$):

$$G\text{-}mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}. \tag{17}$$

Intuitively, precision is a measure of exactness (i.e., of the examples labeled as positive, how many are actually labeled correctly), whereas recall is a measure of completeness (i.e., how many examples of the positive class were labeled correctly). These two metrics, much like accuracy and error, share an inverse relationship between each other. However, unlike accuracy and error, precision and recall are not both sensitive to changes in data distributions. A quick inspection on the precision and recall formulas readily yields that precision (14) is sensitive to data distributions, while recall (15) is not. On the other hand, that recall is not distribution dependent is almost superfluous because an assertion based solely on recall is equivocal, since recall provides no insight to how many examples are incorrectly labeled as positive. Similarly, precision cannot assert how many positive examples are labeled incorrectly. Nevertheless, when used properly, precision and recall can effectively evaluate classification performance in imbalanced learning scenarios. Specifically, the F-Measure metric (16) combines precision and recall as a measure of the effectiveness of classification in terms of a ratio of the weighted importance on either recall or precision as determined by the β coefficient set by the user. As a result, F-Measure provides more insight into the functionality of a classifier than the accuracy metric, however remaining sensitive to data distributions. Another metric, the G-Mean metric (17), evaluates the degree of inductive bias in terms of a ratio of positive accuracy and negative accuracy. Though F-Measure and G-Mean are great improvements over accuracy, they are still ineffective in answering more generic questions about classification evaluations. For instance, *how can we compare the performance of different classifiers over a range sample distributions?*

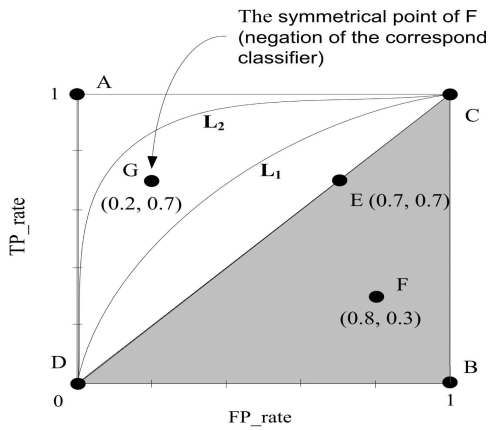


Fig. 10. ROC curve representation.

4.2 Receiver Operating Characteristics (ROC) Curves

In order to overcome such issues, the ROC assessment technique [119], [120] makes use of the proportion of two single-column-based evaluation metrics, namely, true positives rate (TP_rate) and false positives rate (FP_rate), which are defined as:

$$TP\_rate = \frac{TP}{P_C}; \quad FP\_rate = \frac{FP}{N_C}. \tag{18}$$

The ROC graph is formed by plotting TP_rate over FP_rate, and any point in ROC space corresponds to the performance of a single classifier on a given distribution. The ROC curve is useful because it provides a visual representation of the relative trade-offs between the benefits (reflected by true positives) and costs (reflected by false positives) of classification in regards to data distributions.

For hard-type classifiers that output only discrete class labels, each classifier will produce a (TP_rate, FP_rate) pair that corresponds to a single point in the ROC space. Fig. 10 illustrates a typical ROC graph with points A, B, C, D, E, F, and G representing ROC points and curves $L1$ and $L2$ representing ROC curves. According to the structure of the ROC graph, point A $(0, 1)$ represents a perfect classification. Generally speaking, one classifier is better than another if its corresponding point in ROC space closer to point A (upper left hand in the ROC space) than the other. Any classifier whose corresponding ROC point is located on the diagonal, such as point E in Fig. 10, is representative of a classifier that will provide a random guess of the class labels (i.e., a random classifier). Therefore, any classifier that appears in the lower right triangle of ROC space performs worse than random guessing, such as the classifier associated with

point F in the shaded area in Fig. 10. Nevertheless, a classifier that performs worse than random guessing does not mean that the classifier cannot provide useful information. On the contrary, the classifier is informative; however, the information is incorrectly applied. For instance, if one negates the classification results of classifier F, i.e., reverse its classification decision on each instance, then this will produce point G in Fig. 10, the symmetric classification point of F.

In the case of soft-type classifiers, i.e., classifiers that output a continuous numeric value to represent the confidence of an instance belonging to the predicted class, a threshold can be used to produce a series of points in ROC space. This technique can generate an ROC curve instead of a single ROC point, as illustrated by curves L1 and L2 in Fig. 10. In order to assess different classifiers' performance in this case, one generally uses the area under the curve (AUC) as an evaluation criterion [119], [120]. For instance, in Fig. 10, the $L2$ ROC curve provides a larger AUC measure compared to that of $L1$; therefore, the corresponding classifier associated with curve $L2$ can provide better average performance compared to the classifier associated with curve $L1$. Of course, one should also note that it is possible for a high AUC classifier to perform worse in a specific region in ROC space than a low AUC classifier [119], [120]. We additionally note that it is generally very straightforward to make hard-type classifiers provide soft-type outputs based on the observations of the intrinsic characteristics of those classifiers [54], [56], [121], [122].

4.3 Precision-Recall (PR) Curves

Although ROC curves provide powerful methods to visualize performance evaluation, they also have their own limitations. In the case of highly skewed data sets, it is observed that the ROC curve may provide an overly optimistic view of an algorithm's performance. Under such situations, the PR curves can provide a more informative representation of performance assessment [123].

Given a confusion matrix as in Fig. 9 and the definition of precision (14) and recall (15), the PR curve is defined by plotting precision rate over the recall rate. PR curves exhibit a strong correspondence to ROC curves: A curve dominates in ROC space *if and only if* it dominates in PR space [123]. However, an algorithm that optimizes the AUC in the ROC space is not guaranteed to optimize the AUC in PR space [123]. Moreover, while the objective of ROC curves is to be in the upper left hand of the ROC space, a dominant PR curve resides in the upper right hand of the PR space. PR space also characterizes curves analogous to the convex hull in the ROC space, namely, the achievable PR curve [123]. Hence, PR space has all the analogous benefits of ROC space, making it an effective evaluation technique. For space considerations, we refrain from providing a representative figure of PR space and instead direct interested readers to [123].

To see why the PR curve can provide more informative representations of performance assessment under highly imbalanced data, we consider a distribution where negative examples significantly exceed the number of positive examples (i.e., $N_c > P_c$). In this case, if a classifier's performance has a large change in the number of false positives, it will not significantly change the FP_rate since the denominator (Nc) is very large (see (18)). Hence, the ROC graph will fail to capture this phenomenon. The precision metric, on the other hand, considers the ratio of TP with respect to TP+FP (see Fig. 9 and (14)); hence, it can correctly capture the classifier's performance when the number of false positives drastically change [123]. Hence, as evident by this example, the PR curve is an advantageous technique for performance assessment in the presence of highly skewed data. As a result, many of the current research work in the community use PR curves for performance evaluations and comparisons [124], [125], [126], [127].

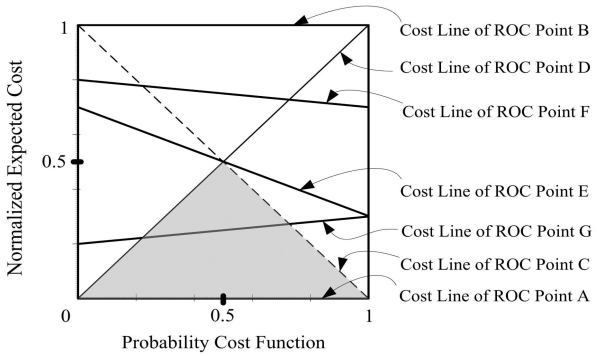


Fig. 11. Cost curve representation.

4.4 Cost Curves

Another shortcoming of ROC curves is that they lack the ability to provide confidence intervals on a classifier's performance and are unable to infer the statistical significance of different classifiers' performance [128], [129]. They also have difficulties providing insights on a classifier's performance over varying class probabilities or misclassification costs [128], [129]. In order to provide a more comprehensive evaluation metric to address these issues, cost curves were proposed in [128], [129], [130]. A cost curve is a cost-sensitive evaluation technique that provides the ability to explicitly express a classifier's performance over varying misclassification costs and class distributions in a visual format. Thus, the cost curve method retains the attractive visual representation features of ROC analysis and further provides a technique that yields a broadened range of information regarding classification performance.

Generally speaking, the cost curve method plots performance (i.e., normalized expected cost) over operation points, which are represented by a probability cost function that is based on the probability of correctly classifying a positive sample. The cost space exhibits a duality with ROC space where each point in ROC space is represented by a line in cost space, and vice versa [128]. Any (FP, TP) classification pair in ROC space is related to a line in cost space by

$$E[C] = (1 - TP - FP) \times PCF(+) + FP, \qquad (19)$$

where E[C] is the expected cost and PCF $(+)$ is the probability of an example being from the positive class. Fig. 11 provides an example of cost space; here, we highlight the correspondence between the ROC points of

Fig. 10 and their lines in cost space. For instance, the bottom axis represents perfect classification, while the top axis represents the contrary case; these lines correspond to ROC points A and B, respectively.

With a collection of cost lines at hand, a cost curve is then created by selecting a classification line for each possible operation point. For example, a cost curve can be created that minimizes the normalized expected cost across all possible operation points. In particular, this technique allows for a clearer visual representation of classification performance compared to ROC curves, as well as more direct assessments between classifiers as they range over operation points.

4.5 Assessment Metrics for Multiclass Imbalanced Learning

While all of the assessment metrics discussed so far in this section are appropriate for two-class imbalanced learning problems, some of them can be modified to accommodate the multiclass imbalanced learning problems. For instance, Fawcett [119] and [120] discussed multiclass ROC graphs. For an n classes problem, the confusion matrix presented in Fig. 9 becomes an $n \times n$ matrix, with n correct classifications (the major diagonal elements) and $n^2 - n$ errors (the off-diagonal elements). Therefore, instead of representing the trade-offs between a single benefit (TP) and cost (FP), we have to manage n benefits and $n^2 - n$ costs. A straightforward way of doing this is to generate n different ROC graphs, one for each class [119], [120]. For instance, considering a problem with a total of W classes, the ROC graph i, ROC_i, plots classification performance using class w_i as the positive class and all other classes as the negative class. However, this approach compromises one of the major advantages of using ROC analysis for imbalanced learning problems: It becomes sensitive to the class skew because the negative class in this situation is the combination of $n - 1$ classes (see Sections 4.1 and 4.2).

Similarly, under the multiclass imbalanced learning scenario, the AUC values for two-class problems become multiple pairwise discriminability values [131]. To calculate such multiclass AUCs, Provost and Domingos [121] proposed a probability estimation-based approach: First, the ROC curve for each reference class w_i is generated and their respective AUCs are measured. Second, all of the AUCs are combined by a weight coefficient according to the reference class's prevalence in the data. Although this approach is quite simple in calculation, it is sensitive to the class skews for the same reason as mentioned before. To eliminate this constraint, Hand and Till [131] proposed the *M measure*, a generalization approach that aggregates all pairs of classes based on the inherent characteristics of the AUC. The major advantage of this method is that it is insensitive to class distribution and error costs. Interested readers can refer to [131] for a more detailed overview of this technique.

In addition to multiclass ROC analysis, the community has also adopted other assessment metrics for multiclass imbalanced learning problems. For instance, in cost-sensitive learning, it is natural to use *misclassification costs* for performance evaluation for multiclass imbalanced problems [8], [10], [11]. Also, Sun et al. [7] extend the G-mean definition (see (17)) to the geometric means of recall values of every class for multiclass imbalanced learning.

5 Opportunities and Challenges

The availability of vast amounts of raw data in many of today's real-world applications enriches the opportunities of learning from imbalanced data to play a critical role across different domains. However, new challenges arise at the same time. Here, we briefly discuss several aspects for the future research directions in this domain.

5.1 Understanding the Fundamental Problems

Currently, most of the research efforts in imbalanced learning focus on specific algorithms and/or case studies; only a very limited amount of theoretical understanding on the principles and consequences of this problem have been addressed. For example, although almost every algorithm presented in literature claims to be able to improve classification accuracy over certain benchmarks, there exist certain situations in which learning from the original data sets may provide better performance. This raises an important question: *to what extent do imbalanced learning methods help with learning capabilities?* This is a fundamental and critical question in this field for the following reasons. First, suppose there are specific (existing or future) techniques or methodologies that significantly outperform others across most (or, ideally, all) applicable domains, then rigorous studies of the underlying effects of such methods would yield fundamental understandings of the problem at hand. Second, as data engineering research methodologies materialize into real-world solutions, questions such as "how will this solution help" or "can this solution efficiently handle various types of data," become the basis on which economic and administrative decisions are made. Thus, the consequences of this critical question have wide-ranging effects in the advancement of this field and data engineering at large.

This important question follows directly from a previous proposition addressed by Provost in the invited paper for the AAAI 2000 Workshop on Imbalanced Data Sets [100]:

"[In regards to imbalanced learning,]... isn't the best research strategy to concentrate on how machine learning algorithms can deal most effectively with whatever data they are given?"

We believe that this fundamental question should be investigated with greater intensity both theoretically and empirically in order to thoroughly understand the essence of imbalanced learning problems. More specifically, we believe that the following questions require careful and thorough investigation:

1. What kind of assumptions will make imbalanced learning algorithms work better compared to learning from the original distributions?
2. To what degree should one balance the original data set?
3. How do imbalanced data distributions affect the computational complexity of learning algorithms?
4. What is the general error bound given an imbalanced data distribution?

5. Is there a general theoretical methodology that can alleviate the impediment of learning from imbalanced data sets for specific algorithms and application domains?

Fortunately, we have noticed that these critical fundamental problems have attracted growing attention in the community. For instance, important works are presented in [37] and [24] that directly relate to the aforementioned question 2 regarding the "level of the desired degree of balance." In [37], the rate of oversampling and undersampling was discussed as a possible aid for imbalanced learning. Generally speaking, though the resampling paradigm has had successful cases in the community, tuning these algorithms effectively is a challenging task. To alleviate this challenge, Estabrooks et al. [37] suggested that a combination of different expressions of resampling methods may be an effective solution to the tuning problem. Weiss and Provost [24] have analyzed, for a fixed training set size, the relationship between the class distribution of training data (expressed as the percentage of minority class examples) and classifier performance in terms of accuracy and AUC. This work provided important suggestions regarding "how do different training data class distributions affect classification performance" and "which class distribution provides the best classifier" [24]. Based on a thorough analysis of 26 data sets, it was suggested that if *accuracy* is selected as the performance criterion, the best class distribution tends to be near the naturally occurring class distribution. However, if the *AUC* is selected as the assessment metric, then the best class distribution tends to be near the balanced class distribution. Based on these observations, a "budget-sensitive" progressive sampling strategy was proposed to efficiently sample the minority and majority class examples such that the resulting training class distribution can provide the best performance.

In summary, the understanding of all these questions will not only provide fundamental insights to the imbalanced learning issue, but also provide an added level of comparative assessment between existing and future methodologies. It is essential for the community to investigate all of these questions in order for research developments to focus on the fundamental issues regarding imbalanced learning.

5.2 Need of a Uniform Benchmark Platform

Data resources are critical for research development in the knowledge discovery and data engineering field. Although there are currently many publicly available benchmarks for assessing the effectiveness of different data engineering algorithm/tools, such as the UCI Repository [132] and the NIST Scientific and Technical Databases [133], there are a very limited number of benchmarks, if any, that are solely dedicated to imbalanced learning problems. For instance, many of the existing benchmarks do not clearly identify imbalanced data sets and their suggested evaluation use in an organized manner. Therefore, many data sets require additional manipulation before they can be applied to imbalanced learning scenarios. This limitation can create a bottleneck for the long-term development of research in imbalanced learning in the following aspects:

1. lack of a uniform benchmark for standardized performance assessments;
2. lack of data sharing and data interoperability across different disciplinary domains;
3. increased procurement costs, such as time and labor, for the research community as a whole group since each research group is required to collect and prepare their own data sets.

With these factors in mind, we believe that a well-organized, publicly available benchmark specifically dedicated to imbalanced learning will significantly benefit the long-term research development of this field. Furthermore, as a required component, an effective mechanism to promote the interoperability and communication across various disciplines should be incorporated into such a benchmark to ultimately uphold a healthy, diversified community.

5.3 Need of Standardized Evaluation Practices

As discussed in Section 4, the traditional technique of using a singular evaluation metric is not sufficient when handling imbalanced learning problems. Although most publications use a broad assortment of singular assessment metrics to evaluate the performance and potential trade-offs of their algorithms, without an accompanied curve-based analysis, it becomes very difficult to provide any concrete relative evaluations between different algorithms, or answer the more rigorous questions of functionality. Therefore, it is necessary for the community to establish—as a standard—the practice of using the curve-based evaluation techniques described in Sections 4.2, 4.3, and 4.4 in their analysis. Not only because each technique provides its own set of answers to different fundamental questions, but also because an analysis in the evaluation space of one technique can be correlated to the evaluation space of another, leading to increased transitivity and a broader understanding of the functional abilities of existing and future works. We hope that a standardized set of evaluation practices for proper comparisons in the community will provide useful guides for the development and evaluation of future algorithms and tools.

5.4 Incremental Learning from Imbalanced Data Streams

Traditional static learning methods require representative data sets to be available at training time in order to develop decision boundaries. However, in many realistic application environments, such as Web mining, sensor networks, multimedia systems, and others, raw data become available continuously over an indefinite (possibly infinite) learning lifetime [134]. Therefore, new understandings, principles, methodologies, algorithms, and tools are needed for such stream data learning scenarios to efficiently transform raw data into useful information and knowledge representation to support the decision-making processes. Although the importance of stream data mining has attracted increasing attention recently, the attention given to imbalanced data streams has been rather limited. Moreover, in regards to incremental learning from imbalanced data streams, many important questions need to be addressed, such as:

1. How can we autonomously adjust the learning algorithm if an imbalance is introduced in the middle of the learning period?
2. Should we consider rebalancing the data set during the incremental learning period? If so, how can we accomplish this?
3. How can we accumulate previous experience and use this knowledge to adaptively improve learning from new data?
4. How do we handle the situation when newly introduced concepts are also imbalanced (i.e., the imbalanced concept drifting issue)?

A concrete understanding and active exploration in these areas can significantly advance the development of technology for real-world incremental learning scenarios.

5.5 Semisupervised Learning from Imbalanced Data

The semisupervised learning problem concerns itself with learning when data sets are a combination of labeled and unlabeled data, as opposed to fully supervised learning where all training data are labeled. The key idea of semisupervised learning is to exploit the unlabeled examples by using the labeled examples to modify, refine, or reprioritize the hypothesis obtained from the labeled data alone [135]. For instance, cotraining works under the assumption of two-viewed sets of feature spaces. Initially, two separate classifiers are trained with the labeled examples on two sufficient and conditionally independent feature subsets. Then, each classifier is used to predict the unlabeled data and recover their labels according to their respective confidence levels [136], [137]. Other representative works for semisupervised learning include the self-training methods [138], [139], semisupervised support vector machines [140], [141], graph-based methods [142], [143], and Expectation-Maximization (EM) algorithm with generative mixture models [144], [145]. Although all of these methods have illustrated great success in many machine learning and data engineering applications, the issue of semisupervised learning under the condition of imbalanced data sets has received very limited attention in the community. Some important questions include:

1. How can we identify whether an unlabeled data example came from a balanced or imbalanced underlying distribution?
2. Given an imbalanced training data with labels, what are the effective and efficient methods for recovering the unlabeled data examples?
3. What kind of biases may be introduced in the recovery process (through the conventional semisupervised learning techniques) given imbalanced, labeled data?

We believe that all of these questions are important not only for theoretical research development, but also for many practical application scenarios.

6 Conclusions

In this paper, we discussed a challenging and critical problem in the knowledge discovery and data engineering field, the imbalanced learning problem. We hope that our discussions of the fundamental nature of the imbalanced learning problem, the state-of-the-art solutions used to address this problem, and the several major assessment techniques used to evaluate this problem will serve as a comprehensive resource for existing and future knowledge discovery and data engineering researchers and practitioners. Additionally, we hope that our insights on the many opportunities and challenges available in this relatively new research area will help guide the potential research directions for the future development of this field.

References

[1] "Learning from Imbalanced Data Sets," *Proc. Am. Assoc. for Artificial Intelligence (AAAI) Workshop,* N. Japkowicz, ed., 2000, (Technical Report WS-00-05).

[2] "Workshop Learning from Imbalanced Data Sets II," *Proc. Int'l Conf. Machine Learning,* N.V. Chawla, N. Japkowicz, and A. Kolcz, eds., 2003.

[3] N.V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 1-6, 2004.

[4] H. He and X. Shen, "A Ranked Subspace Learning Method for Gene Expression Data Classification," *Proc. Int'l Conf. Artificial Intelligence,* pp. 358-364, 2007.

[5] M. Kubat, R.C. Holte, and S. Matwin, "Machine Learning for the Detection of Oil Spills in Satellite Radar Images," *Machine Learning,* vol. 30, nos. 2/3, pp. 195-215, 1998.

[6] R. Pearson, G. Goney, and J. Shwaber, "Imbalanced Clustering for Microarray Time-Series," *Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II,* 2003.

[7] Y. Sun, M.S. Kamel, and Y. Wang, "Boosting for Learning Multiple Classes with Imbalanced Class Distribution," *Proc. Int'l Conf. Data Mining,* pp. 592-602, 2006.

[8] N. Abe, B. Zadrozny, and J. Langford, "An Iterative Method for Multi-Class Cost-Sensitive Learning," *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 3-11, 2004.

[9] K. Chen, B.L. Lu, and J. Kwok, "Efficient Classification of Multi-Label and Imbalanced Data Using Min-Max Modular Classifiers," *Proc. World Congress on Computation Intelligence—Int'l Joint Conf. Neural Networks,* pp. 1770-1775, 2006.

[10] Z.H. Zhou and X.Y. Liu, "On Multi-Class Cost-Sensitive Learning," *Proc. Nat'l Conf. Artificial Intelligence,* pp. 567-572, 2006.

[11] X.Y. Liu and Z.H. Zhou, "Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem," *IEEE Trans. Knowledge and Data Eng.,* vol. 18, no. 1, pp. 63-77, Jan. 2006.

[12] C. Tan, D. Gilbert, and Y. Deville, "Multi-Class Protein Fold Classification Using a New Ensemble Machine Learning Approach," *Genome Informatics,* vol. 14, pp. 206-217, 2003.

[13] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *J. Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[14] H. Guo and H.L. Viktor, "Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost IM Approach," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 30-39, 2004.

[15] K. Woods, C. Doss, K. Bowyer, J. Solka, C. Priebe, and W. Kegelmeyer, "Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography," *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 7, no. 6, pp. 1417-1436, 1993.

[16] R.B. Rao, S. Krishnan, and R.S. Niculescu, "Data Mining for Improved Cardiac Care," *ACM SIGKDD Explorations Newsletter,* vol. 8, no. 1, pp. 3-10, 2006.

[17] P.K. Chan, W. Fan, A.L. Prodromidis, and S.J. Stolfo, "Distributed Data Mining in Credit Card Fraud Detection," *IEEE Intelligent Systems,* vol. 14, no. 6, pp. 67-74, Nov./Dec. 1999.

[18] P. Clifton, A. Damminda, and L. Vincent, "Minority Report in Fraud Detection: Classification of Skewed Data," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 50-59, 2004.

[19] P. Chan and S. Stolfo, "Toward Scalable Learning with Non-Uniform Class and Cost Distributions," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 164-168, 1998.

[20] G.M. Weiss, "Mining with Rarity: A Unifying Framework," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 7-19, 2004.

[21] G.M. Weiss, "Mining Rare Cases," *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers,* pp. 765-776, Springer, 2005.

[22] G.E.A.P.A. Batista, R.C. Prati, and M.C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 20-29, 2004.

[23] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intelligent Data Analysis,* vol. 6, no. 5, pp. 429-449, 2002.

[24] G.M. Weiss and F. Provost, "Learning When Training Data Are Costly: The Effect of Class Distribution on Tree Induction," *J. Artificial Intelligence Research,* vol. 19, pp. 315-354, 2003.

[25] R.C. Holte, L. Acker, and B.W. Porter, "Concept Learning and the Problem of Small Disjuncts," *Proc. Int'l J. Conf. Artificial Intelligence,* pp. 813-818, 1989.

[26] J.R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.

[27] T. Jo and N. Japkowicz, "Class Imbalances versus Small Disjuncts," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 40-49, 2004.

[28] N. Japkowicz, "Class Imbalances: Are We Focusing on the Right Issue?" *Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II,* 2003.

[29] R.C. Prati, G.E.A.P.A. Batista, and M.C. Monard, "Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior," *Proc. Mexican Int'l Conf. Artificial Intelligence,* pp. 312-321, 2004.

[30] S.J. Raudys and A.K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 3, pp. 252-264, Mar. 1991.

[31] R. Caruana, "Learning from Imbalanced Data: Rank Metrics and Extra Tasks," *Proc. Am. Assoc. for Artificial Intelligence (AAAI) Conf.,* pp. 51-57, 2000 (AAAI Technical Report WS-00-05).

[32] W.H. Yang, D.Q. Dai, and H. Yan, "Feature Extraction Uncorrelated Discriminant Analysis for High-Dimensional Data," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 5, pp. 601-614, May 2008.

[33] N.V. Chawla, "C4.5 and Imbalanced Data Sets: Investigating the Effect of Sampling Method, Probabilistic Estimate, and Decision Tree Structure," *Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II,* 2003.

[34] T.M. Mitchell, *Machine Learning.* McGraw Hill, 1997.

[35] G.M. Weiss and F. Provost, "The Effect of Class Distribution on Classifier Learning: An Empirical Study," Technical Report ML-TR-43, Dept. of Computer Science, Rutgers Univ., 2001.

[36] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," *Proc. Conf. AI in Medicine in Europe: Artificial Intelligence Medicine,* pp. 63-66, 2001.

[37] A. Estabrooks, T. Jo, and N. Japkowicz, "A Multiple Resampling Method for Learning from Imbalanced Data Sets," *Computational Intelligence,* vol. 20, pp. 18-36, 2004.

[38] D. Mease, A.J. Wyner, and A. Buja, "Boosted Classification Trees and Class Probability/Quantile Estimation," *J. Machine Learning Research,* vol. 8, pp. 409-439, 2007.

[39] C. Drummond and R.C. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under Sampling Beats Over-Sampling," *Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II,* 2003.

[40] X.Y. Liu, J. Wu, and Z.H. Zhou, "Exploratory Under Sampling for Class Imbalance Learning," *Proc. Int'l Conf. Data Mining,* pp. 965-969, 2006.

[41] J. Zhang and I. Mani, "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction," *Proc. Int'l Conf. Machine Learning (ICML '2003), Workshop Learning from Imbalanced Data Sets,* 2003.

[42] M. Kubat and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," *Proc. Int'l Conf. Machine Learning,* pp. 179-186, 1997.

[43] B.X. Wang and N. Japkowicz, "Imbalanced Data Set Learning with Synthetic Samples," *Proc. IRIS Machine Learning Workshop,* 2004.

[44] H. Han, W.Y. Wang, and B.H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," *Proc. Int'l Conf. Intelligent Computing,* pp. 878-887, 2005.

[45] H. He, Y. Bai, E.A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," *Proc. Int'l J. Conf. Neural Networks,* pp. 1322-1328, 2008.

[46] I. Tomek, "Two Modifications of CNN," *IEEE Trans. System, Man, Cybernetics,* vol. 6, no. 11, pp. 769-772, Nov. 1976.

[47] N.V. Chawla, A. Lazarevic, L.O. Hall, and K.W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," *Proc. Seventh European Conf. Principles and Practice of Knowledge Discovery in Databases,* pp. 107-119, 2003.

[48] H. Guo and H.L. Viktor, "Boosting with Data Generation: Improving the Classification of Hard to Learn Examples," *Proc. Int'l Conf. Innovations Applied Artificial Intelligence,* pp. 1082-1091, 2004.

[49] C. Elkan, "The Foundations of Cost-Sensitive Learning," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 973-978, 2001.

[50] K.M. Ting, "An Instance-Weighting Method to Induce Cost-Sensitive Trees," *IEEE Trans. Knowledge and Data Eng.,* vol. 14, no. 3, pp. 659-665, May/June 2002.

[51] M.A. Maloof, "Learning When Data Sets Are Imbalanced and When Costs Are Unequal and Unknown," *Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II,* 2003.

[52] K. McCarthy, B. Zabar, and G.M. Weiss, "Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes?" *Proc. Int'l Workshop Utility-Based Data Mining,* pp. 69-77, 2005.

[53] X.Y. Liu and Z.H. Zhou, "The Influence of Class Imbalance on Cost-Sensitive Learning: An Empirical Study," *Proc. Int'l Conf. Data Mining,* pp. 970-974, 2006.

[54] P. Domingos, "MetaCost: A General Method for Making Classifiers Cost-Sensitive," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 155-164, 1999.

[55] B. Zadrozny, J. Langford, and N. Abe, "Cost-Sensitive Learning by Cost-Proportionate Example Weighting," *Proc. Int'l Conf. Data Mining,* pp. 435-442, 2003.

[56] Y. Freund and R.E. Schapire, "Experiments with a New Boosting Algorithm," *Proc. Int'l Conf. Machine Learning,* pp. 148-156, 1996.

[57] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[58] Y. Sun, M.S. Kamel, A.K.C. Wong, and Y. Wang, "Cost-Sensitive Boosting for Classification of Imbalanced Data," *Pattern Recognition,* vol. 40, no. 12, pp. 3358-3378, 2007.

[59] W. Fan, S.J. Stolfo, J. Zhang, and P.K. Chan, "AdaCost: Misclassification Cost-Sensitive Boosting," *Proc. Int'l Conf. Machine Learning,* pp. 97-105, 1999.

[60] K.M. Ting, "A Comparative Study of Cost-Sensitive Boosting Algorithms," *Proc. Int'l Conf. Machine Learning,* pp. 983-990, 2000.

[61] M. Maloof, P. Langley, S. Sage, and T. Binford, "Learning to Detect Rooftops in Aerial Images," *Proc. Image Understanding Workshop,* pp. 835-845, 1997.

[62] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Chapman & Hall/CRC Press, 1984.

[63] C. Drummond and R.C. Holte, "Exploiting the Cost (In)Sensitivity of Decision Tree Splitting Criteria," *Proc. Int'l Conf. Machine Learning,* pp. 239-246, 2000.

[64] S. Haykin, *Neural Networks: A Comprehensive Foundation,* second ed. Prentice-Hall, 1999.

[65] M.Z. Kukar and I. Kononenko, "Cost-Sensitive Learning with Neural Networks," *Proc. European Conf. Artificial Intelligence,* pp. 445-449, 1998.

[66] P. Domingos and M. Pazzani, "Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier," *Proc. Int'l Conf. Machine Learning,* pp. 105-112, 1996.

[67] G.R.I. Webb and M.J. Pazzani, "Adjusted Probability Naive Bayesian Induction," *Proc. Australian Joint Conf. Artificial Intelligence,* pp. 285-295, 1998.

[68] R. Kohavi and D. Wolpert, "Bias Plus Variance Decomposition for Zero-One Loss Functions," *Proc. Int'l Conf. Machine Learning,* 1996.

[69] J. Gama, "Iterative Bayes," *Theoretical Computer Science,* vol. 292, no. 2, pp. 417-430, 2003.

[70] G. Fumera and F. Roli, "Support Vector Machines with Embedded Reject Option," *Proc. Int'l Workshop Pattern Recognition with Support Vector Machines,* pp. 68-82, 2002.

[71] J.C. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," *Advances in Kernel Methods: Support Vector Learning,* pp. 185-208, MIT Press, 1999.

[72] J.T. Kwok, "Moderating the Outputs of Support Vector Machine Classifiers," *IEEE Trans. Neural Networks,* vol. 10, no. 5, pp. 1018-1031, Sept. 1999.

[73] V.N. Vapnik, *The Nature of Statistical Learning Theory.* Springer, 1995.

[74] B. Raskutti and A. Kowalczyk, "Extreme Re-Balancing for SVMs: A Case Study," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 60-69, 2004.

[75] R. Akbani, S. Kwek, and N. Japkowicz, "Applying Support Vector Machines to Imbalanced Data Sets," *Lecture Notes in Computer Science,* vol. 3201, pp. 39-50, 2004.

[76] G. Wu and E. Chang, "Class-Boundary Alignment for Imbalanced Data Set Learning," *Proc. Int'l Conf. Data Mining (ICDM '03), Workshop Learning from Imbalanced Data Sets II,* 2003.

[77] F. Vilarino, P. Spyridonos, P. Radeva, and J. Vitria, "Experiments with SVM and Stratified Sampling with an Imbalanced Problem: Detection of Intestinal Contractions," *Lecture Notes in Computer Science,* vol. 3687, pp. 783-791, 2005.

[78] P. Kang and S. Cho, "EUS SVMs: Ensemble of Under sampled SVMs for Data Imbalance Problems," *Lecture Notes in Computer Science,* vol. 4232, pp. 837-846, 2006.

[79] Y. Liu, A. An, and X. Huang, "Boosting Prediction Accuracy on Imbalanced Data Sets with SVM Ensembles," *Lecture Notes in Artificial Intelligence,* vol. 3918, pp. 107-118, 2006.

[80] B.X. Wang and N. Japkowicz, "Boosting Support Vector Machines for Imbalanced Data Sets," *Lecture Notes in Artificial Intelligence,* vol. 4994, pp. 38-47, 2008.

[81] Y. Tang and Y.Q. Zhang, "Granular SVM with Repetitive Undersampling for Highly Imbalanced Protein Homology Prediction," *Proc. Int'l Conf. Granular Computing,* pp. 457- 460, 2006.

[82] Y.C. Tang, B. Jin, and Y.-Q. Zhang, "Granular Support Vector Machines with Association Rules Mining for Protein Homology Prediction," *Artificial Intelligence in Medicine,* special issue on computational intelligence techniques in bioinformatics, vol. 35, nos. 1/2, pp. 121-134, 2005.

[83] Y.C. Tang, B. Jin, Y.-Q. Zhang, H. Fang, and B. Wang, "Granular Support Vector Machines Using Linear Decision Hyperplanes for Fast Medical Binary Classification," *Proc. Int'l Conf. Fuzzy Systems,* pp. 138-142, 2005.

[84] Y.C. Tang, Y.Q. Zhang, Z. Huang, X.T. Hu, and Y. Zhao, "Granular SVM-RFE Feature Selection Algorithm for Reliable Cancer-Related Gene Subsets Extraction on Microarray Gene Expression Data," *Proc. IEEE Symp. Bioinformatics and Bioeng.,* pp. 290-293, 2005.

[85] X. Hong, S. Chen, and C.J. Harris, "A Kernel-Based Two-Class Classifier for Imbalanced Data Sets," *IEEE Trans. Neural Networks,* vol. 18, no. 1, pp. 28-41, Jan. 2007.

[86] G. Wu and E.Y. Chang, "Aligning Boundary in Kernel Space for Learning Imbalanced Data Set," *Proc. Int'l Conf. Data Mining,* pp. 265-272, 2004.

[87] G. Wu and E.Y. Chang, "KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 6, pp. 786-795, June 2005.

[88] G. Wu and E.Y. Chang, "Adaptive Feature-Space Conformal Transformation for Imbalanced-Data Learning," *Proc. Int'l Conf. Machine Learning,* pp. 816-823, 2003.

[89] Y.H. Liu and Y.T. Chen, "Face Recognition Using Total Margin-Based Adaptive Fuzzy Support Vector Machines," *IEEE Trans. Neural Networks,* vol. 18, no. 1, pp. 178-192, Jan. 2007.

[90] Y.H. Liu and Y.T. Chen, "Total Margin Based Adaptive Fuzzy Support Vector Machines for Multiview Face Recognition," *Proc. Int'l Conf. Systems, Man and Cybernetics,* pp. 1704-1711, 2005.

[91] G. Fung and O.L. Mangasarian, "Multicategory Proximal Support Vector Machine Classifiers," *Machine Learning,* vol. 59, nos. 1/2, pp. 77-97, 2005.

[92] J. Yuan, J. Li, and B. Zhang, "Learning Concepts from Large Scale Imbalanced Data Sets Using Support Cluster Machines," *Proc. Int'l Conf. Multimedia,* pp. 441-450, 2006.

[93] A.K. Qin and P.N. Suganthan, "Kernel Neural Gas Algorithms with Application to Cluster Analysis," *Proc. Int'l Conf. Pattern Recognition,* 2004.

[94] X.P. Yu and X.G. Yu, "Novel Text Classification Based on K-Nearest Neighbor," *Proc. Int'l Conf. Machine Learning Cybernetics,* pp. 3425-3430, 2007.

[95] P. Li, K.L. Chan, and W. Fang, "Hybrid Kernel Machine Ensemble for Imbalanced Data Sets," *Proc. Int'l Conf. Pattern Recognition,* pp. 1108-1111, 2006.

[96] A. Tashk, R. Bayesteh, and K. Faez, "Boosted Bayesian Kernel Classifier Method for Face Detection," *Proc. Int'l Conf. Natural Computation,* pp. 533-537, 2007.

[97] N. Abe, "Invited Talk: Sampling Approaches to Learning from Imbalanced Data Sets: Active Learning, Cost Sensitive Learning and Deyond," *Proc. Int'l Conf. Machine Learning, Workshop Learning from Imbalanced Data Sets II,* 2003.

[98] S. Ertekin, J. Huang, L. Bottou, and L. Giles, "Learning on the Border: Active Learning in Imbalanced Data Classification," *Proc. ACM Conf. Information and Knowledge Management,* pp. 127-136, 2007.

[99] S. Ertekin, J. Huang, and C.L. Giles, "Active Learning for Class Imbalance Problem," *Proc. Int'l SIGIR Conf. Research and Development in Information Retrieval,* pp. 823-824, 2007.

[100] F. Provost, "Machine Learning from Imbalanced Data Sets 101," *Proc. Learning from Imbalanced Data Sets: Papers from the Am. Assoc. for Artificial Intelligence Workshop,* 2000 (Technical Report WS-00-05).

[101] Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast Kernel Classifiers with Online and Active Learning," *J. Machine Learning Research,* vol. 6, pp. 1579-1619, 2005.

[102] J. Zhu and E. Hovy, "Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem," *Proc. Joint Conf. Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* pp. 783-790, 2007.

[103] J. Doucette and M.I. Heywood, "GP Classification under Imbalanced Data Sets: Active Sub-Sampling AUC Approximation," *Lecture Notes in Computer Science,* vol. 4971, pp. 266-277, 2008.

[104] B. Scholkopt, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation,* vol. 13, pp. 1443-1471, 2001.

[105] L.M. Manevitz and M. Yousef, "One-Class SVMs for Document Classification," *J. Machine Learning Research,* vol. 2, pp. 139-154, 2001.

[106] L. Zhuang and H. Dai, "Parameter Estimation of One-Class SVM on Imbalance Text Classification," *Lecture Notes in Artificial Intelligence,* vol. 4013, pp. 538-549, 2006.

[107] H.J. Lee and S. Cho, "The Novelty Detection Approach for Difference Degrees of Class Imbalance," *Lecture Notes in Computer Science,* vol. 4233, pp. 21-30, 2006.

[108] L. Zhuang and H. Dai, "Parameter Optimization of Kernel-Based One-Class Classifier on Imbalance Text Learning," *Lecture Notes in Artificial Intelligence,* vol. 4099, pp. 434-443, 2006.

[109] N. Japkowicz, "Supervised versus Unsupervised Binary-Learning by Feedforward Neural Networks," *Machine Learning,* vol. 42, pp. 97-122, 2001.

[110] L. Manevitz and M. Yousef, "One-Class Document Classification via Neural Networks," *Neurocomputing,* vol. 70, pp. 1466-1481, 2007.

[111] N. Japkowicz, "Learning from Imbalanced Data Sets: A Comparison of Various Strategies," *Proc. Am. Assoc. for Artificial Intelligence (AAAI) Workshop Learning from Imbalanced Data Sets,* pp. 10-15, 2000 (Technical Report WS-00-05).

[112] N. Japkowicz, C. Myers, and M. Gluck, "A Novelty Detection Approach to Classification," *Proc. Joint Conf. Artificial Intelligence,* pp. 518-523, 1995.

[113] C.T. Su and Y.H. Hsiao, "An Evaluation of the Robustness of MTS for Imbalanced Data," *IEEE Trans. Knowledge and Data Eng.,* vol. 19, no. 10, pp. 1321-1332, Oct. 2007.

[114] G. Taguchi, S. Chowdhury, and Y. Wu, *The Mahalanobis-Taguchi System.* McGraw-Hill, 2001.

[115] G. Taguchi and R. Jugulum, *The Mahalanobis-Taguchi Strategy.* John Wiley & Sons, 2002.

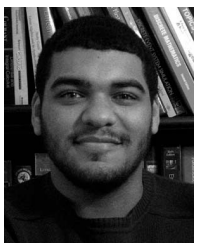
[116] M.V. Joshi, V. Kumar, and R.C. Agarwal, "Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements," *Proc. Int'l Conf. Data Mining,* pp. 257-264, 2001.

[117] F.J. Provost and T. Fawcett, "Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 43-48, 1997.

[118] F.J. Provost, T. Fawcett, and R. Kohavi, "The Case against Accuracy Estimation for Comparing Induction Algorithms," *Proc. Int'l Conf. Machine Learning,* pp. 445-453, 1998.

[119] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers," Technical Report HPL-2003-4, HP Labs, 2003.
[120] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006.
[121] F. Provost and P. Domingos, "Well-Trained Pets: Improving Probability Estimation Trees," CeDER Working Paper: IS-00-04, Stern School of Business, New York Univ., 2000.
[122] T. Fawcett, "Using Rule Sets to Maximize ROC Performance," *Proc. Int'l Conf. Data Mining,* pp. 131-138, 2001.
[123] J. Davis and M. Goadrich, "The Relationship between Precision-Recall and ROC Curves," *Proc. Int'l Conf. Machine Learning,* pp. 233-240, 2006.
[124] R. Bunescu, R. Ge, R. Kate, E. Marcotte, R. Mooney, A. Ramani, and Y. Wong, "Comparative Experiments on Learning Information Extractors for Proteins and Their Interactions," *Artificial Intelligence in Medicine,* vol. 33, pp. 139-155, 2005.
[125] J. Davis, E. Burnside, I. Dutra, D. Page, R. Ramakrishnan, V.S. Costa, and J. Shavlik, "View Learning for Statistical Relational Learning: With an Application to Mammography," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 677-683, 2005.
[126] P. Singla and P. Domingos, "Discriminative Training of Markov Logic Networks," *Proc. Nat'l Conf. Artificial Intelligence,* pp. 868-873, 2005.
[127] T. Landgrebe, P. Paclik, R. Duin, and A.P. Bradley, "Precision-Recall Operating Characteristic (P-ROC) Curves in Imprecise Environments," *Proc. Int'l Conf. Pattern Recognition,* pp. 123-127, 2006.
[128] R.C. Holte and C. Drummond, "Cost Curves: An Improved Method for Visualizing Classifier Performance," *Machine Learning,* vol. 65, no. 1, pp. 95-130, 2006.
[129] R.C. Holte and C. Drummond, "Cost-Sensitive Classifier Evaluation," *Proc. Int'l Workshop Utility-Based Data Mining,* pp. 3-9, 2005.
[130] R.C. Holte and C. Drummond, "Explicitly Representing Expected Cost: An Alternative to ROC Representation," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 198-207, 2000.
[131] D.J. Hand and R.J. Till, "A Simple Generalization of the Area under the ROC Curve to Multiple Class Classification Problems," *Machine Learning,* vol. 45, no. 2, pp. 171-186, 2001.
[132] UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/, 2009.
[133] NIST Scientific and Technical Databases, http://nist.gov/srd/online.htm, 2009.
[134] H. He and S. Chen, "IMORL: Incremental Multiple Objects Recognition Localization," *IEEE Trans. Neural Networks,* vol. 19, no. 10, pp. 1727-1738, Oct. 2008.
[135] X. Zhu, "Semi-Supervised Learning Literature Survey," Technical Report TR-1530, Univ. of Wisconsin-Madson, 2007.
[136] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. Workshop Computational Learning Theory,* pp. 92-100, 1998.
[137] T.M. Mitchell, "The Role of Unlabeled Data in Supervised Learning," *Proc. Int'l Colloquium on Cognitive Science,* 1999.
[138] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models," *Proc. IEEE Workshops Application of Computer Vision,* pp. 29-36, 2005.
[139] M. Wang, X.S. Hua, L.R. Dai, and Y. Song, "Enhanced Semi-Supervised Learning for Automatic Video Annotation," *Proc. Int'l Conf. Multimedia and Expo,* pp. 1485-1488, 2006.
[140] K.P. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," *Proc. Conf. Neural Information Processing Systems,* pp. 368-374, 1998.
[141] V. Sindhwani and S.S. Keerthi, "Large Scale Semi-Supervised Linear SVMs," *Proc. Int'l SIGIR Conf. Research and Development in Information Retrieval,* pp. 477-484, 2006.
[142] A. Blum and S. Chawla, "Learning from Labeled and Unlabeled Data Using Graph Mincuts," *Proc. Int'l Conf. Machine Learning,* pp. 19-26, 2001.
[143] D. Zhou, B. Scholkopf, and T. Hofmann, "Semi-Supervised Learning on Directed Graphs," *Proc. Conf. Neural Information Processing Systems,* pp. 1633-1640, 2004.
[144] A. Fujino, N. Ueda, and K. Saito, "A Hybrid Generative/Discriminative Approach to Semi-Supervised Classifier Design," *Proc. Nat'l Conf. Artificial Intelligence,* pp. 764-769, 2005.
[145] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labeled and Unlabelled Data," *Proc. Ann. Conf. Neural Information Processing Systems,* pp. 571-577, 1996.

Haibo He received the BS and MS degrees in electrical engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1999 and 2002, respectively, and the PhD degree in electrical engineering from Ohio University, Athens, in 2006. He is currently an assistant professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, New Jersey. His research interests include machine learning, data mining, computational intelligence, VLSI and FPGA design, and embedded intelligent systems design. He has served regularly on the organization committees and the program committees of many international conferences and has also been a reviewer for the leading academic journals in his fields, including the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Neural Networks*, the *IEEE Transactions on Systems, Man and Cybernetics* (*part A* and *part B*), and others. He has also served as a guest editor for several international journals, such as *Soft Computing* (Springer) and *Applied Mathematics and Computation* (Elsevier), among others. He has delivered several invited talks including the IEEE North Jersey Section Systems, Man & Cybernetics invited talk on "Self-Adaptive Learning for Machine Intelligence." He was the recipient of the Outstanding Master Thesis Award of Hubei Province, China, in 2002. Currently, he is the editor of the *IEEE Computational Intelligence Society (CIS) Electronic Letter (E-letter)*, and a committee member of the IEEE Systems, Man, and Cybernetic (SMC) Technical Committee on Computational Intelligence. He is a member of the IEEE, the ACM, and the AAAI.



Edwardo A. Garcia received the BS degree in mathematics from New York University, New York, and the BE degree in computer engineering from Stevens Institute of Technology, Hoboken, New Jersey, both in 2008. He currently holds research appointments with the Department of Electrical and Computer Engineering at Stevens Institute of Technology and with the Department of Anesthesiology at New York University School of Medicine. His research interests include machine learning, biologically inspired intelligence, cognitive neuroscience, data mining for medical diagnostics, and mathematical methods for f-MRI.