Learning Imbalanced Multi-class Data with Optimal Dichotomy Weights

Xu-Ying Liu*†, Qian-Qian Li* and Zhi-Hua Zhou†
*Key Laboratory of Computer Network and Information Integration, MOE, Southeast University, China
†National Key Laboratory for Novel Software Technology, Nanjing University, China
Email: {liuxy,liqianqian}@seu.edu.cn, zhouzh@lamda.nju.edu.cn

***Abstract*—Class-imbalance is very common in real data mining tasks. Previous studies focused on binary-class imbalance problem, whereas multi-class imbalance problem is more challenging. Error correcting output codes (ECOC) technique can be applied to class-imbalance problem; however, the standard ECOC aims at maximizing accuracy, ignoring the fact that, when class-imbalance is really a problem, the minority classes are more important than the majority classes. To enable ECOC to tackle multi-class imbalance, it is desired to have an appropriate code matrix, an effective learning strategy and a decoding strategy emphasizing the minority classes. In this paper, based on the aforementioned consideration, we propose the imECOC method which works on dichotomies to handle both the between-class imbalance and within-class imbalance. As the dichotomy classifiers contribute differently to the final prediction, imECOC assigns weights to dichotomies and uses weighted distance for decoding, where the optimal dichotomy weights are obtained by minimizing a weighted loss in favor of the minority classes. Experimental results on fourteen data sets show that, imECOC performs significantly better than many state-of-the-art multi-class imbalance learning methods, no matter whether multi-class F1, G-mean or AUC are used as evaluation measures.**

Keywords*—*Class-imbalance learning, Multi-class; ECOC

I. Introduction

Many real data mining tasks suffer from class-imbalance, i.e., some classes have much fewer data than the other classes, whereas the minority classes (i.e., the smaller classes) are more important. Conventional learning methods often try to pursue a high accuracy by assuming that all classes have similar sizes, leading to the fact that the minority class examples are often overlooked and misclassified to the majority classes (i.e., the bigger classes). However, accuracy is not an adequate evaluation measure when there is class-imbalance because it ignores the larger importance of minority classes. Instead, F1 measure, G-mean and AUC [4] are suitable evaluation measures for class-imbalance problems.

Class-imbalance learning has been considered as one of the most challenging problems in machine learning and data mining [31], whereas previous studies generally focused on binary-class data [6], [11], [14], [17], [18], [20], [30]. It is noteworthy that class-imbalance is more common in multi-class data, and multi-class imbalance is more challenging than binary-class imbalance. One obvious reason lies in the fact that binary-class imbalance is bilateral, whereas multi-class imbalance is multilateral. There are also higher degree of concept complexity, more small disjuncts, and even higher imbalance rate (the size of the largest class divided by the size of the smallest class). There are a few work studied multi-class imbalance [12], [16], [19], [21], [26], [27], [29], and they will be briefly reviewed in Section II.

To tackle multi-class imbalance, it is important to consider the multilateral relation. ECOC (Error Correcting Output Codes) [1], [2], [9], [25] is a popular method for multi-class learning, working by decomposing the multi-class task to a series of binary class subtasks (dichotomies) and then constructing a binary classifier from each dichotomy. A test instance is evaluated by all the dichotomy classifiers and then assigned to the nearest class in the code space. The bilateral imbalance in each dichotomy is evidently easier to handle than the original multilateral imbalance. However, standard ECOC aims at maximizing accuracy, disabling its usage in class-imbalance tasks as aforementioned, accuracy is not adequate in class-imbalance tasks. Thus, to enable ECOC to handle multi-class imbalance, we need to consider: (a) How to generate code matrix to identify the minority classes? (b) How to address the bilateral class-imbalance of dichotomy and the multilateral class-imbalance of the original task simultaneously? (c) How to deal with under-performed classifiers as some dichotomies might be quite difficult? (d) How to design the decoding strategy in favor of the minority classes?

In this paper, we propose the imECOC method. The basic idea contains two ingredients: (1) both of the between-class and the within-class imbalance are considered in each dichotomy; (2) considering the dichotomy classifiers should contribute differently to the final prediction, imECOC assigns weights to dichotomies and uses weighted distance for decoding, where the optimal weights are obtained by minimizing a weighed loss in favor of the minority classes. The first and second ingredients help tackle (a) and (b, c, d) respectively. Experimental results on fourteen data sets show that, imECOC performs significantly better than many state-of-the-art multi-class imbalance learning methods, no matter whether F1, G-mean and AUC are used as evaluation measures.

The rest of the paper is organized as follows. Section II briefly reviews some related work. Section III introduces ECOC. Section IV proposes the imECOC method. Section V reports our experimental results. Finally, Section VI concludes.

II. Related Work

The binary-class imbalance learning methods can be roughly grouped into 5 categories: (1) Balancing the training set via sampling. Two most simple but popular sampling methods are random over- and under-sampling. There are many

other sampling methods. Among them, SMOTE [6], the state-of-the-art method in the literature, reduces the over-fitting of over-sampling by adding synthetic minority class examples via interpolating between the minority class examples and their neighbors. (2) Cost-sensitive learning methods [32], [33]. They assign higher and lower costs to the minority and majority class examples, respectively. (3) Refining the placement of decision boundary. The most straightforward way is threshold-moving, which moves the decision threshold such that the minority class examples are easier to be classified correctly. (4) Methods particularly designed for class-imbalance problem. (5) Ensemble-based methods, such as [20]. Please refer to [14], [15] for surveys.

There are a few studies [12], [16], [19], [21], [26], [27], [29] on multi-class imbalance. Most of them were published very recently [12], [16], [19], [29]. [26] uses genetic algorithm to determine the optimal cost vector and exploits a multi-class cost-sensitive learning method. MC-HDDT [16] is a decision tree method using a multi-class splitting criterion in favor of the minority classes. DyS [19] is a neural network dynamically sampling data to update the weights during the learning procedure. OvNC [29] combines boosting and over-sampling for the imbalance problems with "multi-minority" and "multi-majority" classes. Among the methods for imbalanced multi-class data, much attention was paid to reduction-based methods, especially OVA (one versus all) and OVO (one versus one). As two representatives, [27] uses OVA and OVO reduction and designs a decision rule method to handle the binary-class imbalance in each subtask, and OAHO [21] reduces a multi-class learning task to a series of binary-class subtasks by grouping the minority classes together to decrease the imbalance rate.

III. ECOC

Given training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, with $\mathbf{x}$ in domain $\mathcal{X} \in \mathbb{R}^d$ and y in domain $\mathcal{Y} = \{1, 2, \ldots, k\}$. Let $\{C_1, C_2, ..., C_k\}$ denote the set of classes sorted in ascending order of class sizes, i.e., $n_1 \leq n_2 \leq \ldots \leq n_k$, where n_i is the number of examples of C_i.

ECOC [9] has three stages: coding, learning and decoding. In the coding stage, ECOC decomposes a multi-class problem with k classes into ℓ dichotomies with the codewords (code matrix) $\mathbf{M} \in \{-1, 0, +1\}^{k \times \ell}$. A column of $\mathbf{M}$ encodes a partition of k classes into -1 and $+1$ classes[1]. When a class is encoded with 0, it is excluded from the training set of the dichotomy, which is known as ternary ECOC [1]. The popular reduction methods OVA and OVO are special cases of ternary ECOC. A row of the $\mathbf{M}$ represents the codeword of a class. We denote the codeword of class C_i as $\mathbf{M}_i$.

In the learning stage, suppose the learning algorithm trains a dichotomy classifier $h_i : \mathcal{X} \rightarrow \{+1, -1\}$ for the i-th dichotomy. Some algorithms can train dichotomy classifiers with real-valued outputs. In this case, we denote the i-th dichotomy classifier as $f_i : \mathcal{X} \rightarrow \mathcal{R}$ for more general formulation. Then, in the decoding stage, each classifier $f_i, \forall i = 1, \ldots, \ell$, predicts a value for a test instance $\mathbf{x}$, resulting in the codeword of $\mathbf{x}$: $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_\ell(\mathbf{x}))$. Then the test instance is assigned to the "closest" class in the code space. That is,

$$\hat{y} = \arg\min_r d(\mathbf{F}(\mathbf{x}), \mathbf{M}_r),$$

where, $d : \mathbb{R}^\ell \times \mathbb{R}^\ell \rightarrow \mathbb{R}$ is a distance function of two codewords. A common distance function is Hamming distance:

$$d_H(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{\ell} \frac{1 - \text{sign}(u_i \cdot v_i)}{2}, \quad (1)$$

where, $\mathbf{u}$ and $\mathbf{v}$ are two vectors of length ℓ, and $\text{sign}(x)$ is $+1$ if $x > 0$, -1 if $x < 0$ and 0 if $x = 0$. Some other popular distance functions include Euclidean distance and loss-based distance [1].

ECOC originates from error correcting codes, and the discriminative power of ECOC depends heavily on the error-correcting ability of the code matrix. The error-correcting ability involves two keys: large codeword distance between any pair of classes, and independent bit errors. A ECOC code matrix should satisfy [9]:

- Row separation. Each row of the code matrix should be well-separated in distance from each of the other rows.
- Column separation. Each bit function f_i should not be correlated with the functions to be learned for the other bit positions $f_j, j \neq i$. This can be achieved by insisting that each column of the code matrix is well-separated in distance from every other columns and their complementary.

IV. The Proposed imECOC Method

A. Problem Analysis

Class-imbalance problem is caused mainly by class-imbalance (large difference among class sizes) associating the importance of the minority classes, it is also affected by other factors, including concept complexity, separability among classes, small disjuncts, etc.

- Concept complexity and the absolute amount of data in each class play very important roles in class-imbalance problem [17], [30]. When the degree of concept complexity is very high, large amount of data is required to learn a concept for each class successfully. The lack of data is a great cause of class-imbalance problem.
- When classes have many overlaps, the degree of concept complexity is usually high. Therefore, the more overlaps the classes have, the more the class-imbalance harms the standard methods [24]. When data is near linearly separable, the degree of concept complexity is very low, and class-imbalance barely harms. On the contrary, the performance would generally decrease when normal balancing method is used in such cases [20].
- There are many small disjuncts when data scatters around. Such small disjuncts usually have higher error

[1]To avoid confusion, we call the classes in the original imbalanced multi-class data the minority classes or the majority classes, and the two classes in a dichotomy of ECOC the small class or the large class when they are imbalanced.

rates than large disjuncts [30], and can hardly be classified with a guarantee on accuracy, even with a large training set.

On multi-class data, the degree of concept complexity is usually much higher than that on binary-class data, multiple classes may have a lot of overlaps and small disjuncts occur frequently. Thus class-imbalance problem is more serious in multi-class data. Moreover, the multilateral relation among classes makes the problem even more complex. For example, multi-minority cases (i.e., one majority class and multiple minority classes) and multi-majority cases (i.e., one minority class and multiple majority classes) probably have very different properties [29].

It is natural to decompose a multi-class imbalance learning task to a series of binary-class learning subtasks which are much easier to handle. In such a way, by combining the boundaries of all dichotomy classifiers, the multilateral relation can be easily revealed. ECOC works in this manner, and has shown to be a good choice with some other advantages on overcoming the above challenges: (1) The imbalance rate of a dichotomy could be smaller than original imbalance rate in many cases, for example, the original minority classes merge together or the original majority classes are excluded in some dichotomies. (2) In a dichotomy, the contradiction of high degree of concept complexity and the lack of data could be alleviated from both sides. This advantage is particularly important when some original classes have very few examples. (3) The small disjuncts problem could be relieved when several original classes merged together or some original classes are excluded in some dichotomies.

In class-imbalance problem, the minority class examples are more important, and they should be treated appropriately in a good imbalance learning method. Since ECOC aims at maximizing accuracy, which means all examples have the same importance, it can't be directly used for imbalanced multi-class data. Standard ECOC has 3 stages: coding, learning and decoding. None of them considers the difference between a minority class example and a majority class example. So the technical difficulties exist in every stage.

(D1) How to generate code matrix to identify the minority classes?

Since the minority classes are more important when there is class-imbalance problem, the error-correcting abilities of the minority classes should be more important than that of the majority classes. Generally speaking, a code matrix good for standard classification is not suitable for class-imbalanced classification since it treats all examples equally. For example, in an extreme case, the smallest minority class is encoded as the positive class and all the other classes are encoded as the negative class in a dichotomy. The class-imbalance is largely enhanced and the minority class examples will be greatly overlooked, leading to significant decrease of the error-correcting ability on the minority class.

(D2) How to address the bilateral class-imbalance of dichotomy and the multilateral class-imbalance of the original task simultaneously?

The positive class and negative class in a dichotomy are generally imbalanced since the multi-class data is imbalanced. Standard ECOC treats all examples equally to maximize dichotomy's accuracy, so it would overlook the small class examples. To learn binary-class concepts successfully, the bilateral imbalance between dichotomy's two classes should be addressed. On the other hand, the multilateral class-imbalance of the original task must also be addressed to classify the minority classes correctly.

(D3) How to deal with the under-performed classifiers as some dichotomies might be quite difficult?

A classifier may fail to learn the dichotomy's binary concept or fail to recognize the original minority classes, or fail in both cases as some dichotomies might be quite difficult. The under-performed classifiers should be distinguished and properly handled, otherwise it will be harmful to the decoding stage. But it is hard to set a certain rule to determine a classifier's effectiveness. Inappropriate qualifying can be very misleading.

(D4) How to design the decoding strategy in favor of the minority classes?

In the decoding stage, a test instance is assigned to the class with the minimal distance (or loss) to its predicted code. This implies that a minority class example has the same importance as a majority class example. To make the minority class examples easier to be classified correctly, the decision region of a minority class should be enlarged. But there is multilateral class-imbalance, so it is a challenge to implement.

In fact, these difficulties correlate with each other, which makes it more difficult to adapt ECOC to imbalanced multi-class data.

B. imECOC Method

The proposed imECOC method tackles the above difficulties by two key ingredients: (1) To solve the learning difficulty, balance is achieved not only between the positive and negative classes (large and small classes) of a dichotomy, but also among the subclasses (the original minority and majority classes) within each class in the learning process. (2) imECOC overcomes the other difficulties in an unified strategy in the decoding stage. Considering the dichotomy classifiers should contribute differently to the final prediction since they have different effectiveness, imECOC assigns weights to dichotomies and uses weighted distance for decoding, where the optimal dichotomy weights are obtained by minimizing a weighed loss in favor of the minority classes. This process ensures the decoding is effective for the original imbalanced multi-class classification.

We will explain how the method tackles these difficulties in detail in the coding, learning and decoding stage separately.

1) Coding Stage: A code matrix should be suitable for imbalanced data. One possible way is to remove the inadequate dichotomy codes from a code matrix of standard ECOC. However, similar to qualifying dichotomy classifiers in (D3), it is hard to determine how good a code is. In fact, whether a code matrix is suitable to imbalanced data can be reflected by the effectiveness of the dichotomy classifiers. Therefore, handling inadequate dichotomy codes can be transformed to handling under-performed classifiers, which will be solved in

the decoding stage. We will continue to discuss this problem in Section IV-B3. Therefore, imECOC can work with a general code matrix of standard ECOC. Some popular ones are Dense coding and Sparse coding [1], complete coding [9], OVA and OVO which are special cases of ECOC coding. There are some problem-dependent coding methods [22], [23]. Since they aim at maximizing accuracy, which contradicts class-imbalance learning, so we focus on problem-independent coding here.

2) Learning Stage: The positive class and the negative class in a dichotomy are generally imbalanced since the original data is imbalanced. We call this kind of imbalance the *between-class imbalance*. On the other hand, within each of the positive and negative class of a dichotomy, the subclasses are with different sizes. We call this kind of imbalance the *within-class imbalance*. For example, there is a data set with 4 classes $\{C_1, C_2, C_3, C_4\}$, and $n_1 = 10, n_2 = 50, n_3 = 100, n_4 = 1000$. A dichotomy code is $[+1, -1, +1, -1]$, so the small class of the dichotomy is $A_+ = \{C_1, C_3\}$ with size $|A_+| = 110$, and the large class is $A_- = \{C_2, C_4\}$ with size $|A_-| = 1050$. The between-class imbalance is the imbalance between A_+ and A_-, the imbalance rate is $|A_-|/|A_+| = 9.55$. Both of A_+ and A_- have within-class imbalance in this example. The within-class imbalance of A_+ is the imbalance between its subclasses C_1 and C_3, the imbalance rate is $n_3/n_1 = 10$. The within-class imbalance of A_- is the imbalance between C_2 and C_4, the imbalance rate is $n_4/n_2 = 20$.

The between-class imbalance should be handled to avoid overlooking the small class examples, especially when the original minority classes are contained in the small class. In the example, if A_+ is overlooked, both of its subclasses C_1 and C_3 will be overlooked, where C_1 is the most important class among the original classes. The within-class imbalance should also be handled to ensure examples of the original minority class have lager chance to be classified correctly than the other examples in the same category, especially when a minority class is in the dichotomy's small class. For another example, if the class size is 10, 20, 190, 200, respectively, and the dichotomy code is $[+1, -1, -1, +1]$, then there is no between-class imbalance, the within-class imbalance of A_+ and A_- is $n_4/n_1 = 20$ and $n_3/n_2 = 9.5$, respectively. If the within-class imbalance was not handled, C_1 and/or C_2 would be overlooked. Therefore, the between-class and the within-class imbalance must be handled simultaneously. Based on this idea, we design BWC-weighting method (Between- and Within-Class weighting) for imECOC to perform binary-class classification in a dichotomy.

In the t-th dichotomy, the classes with $\mathbf{M}(i,t) = 0$ are excluded from the training set. let $A_+^t = \{C_i | \mathbf{M}(i,t) = +1\}$ and $A_-^t = \{C_i | \mathbf{M}(i,t) = -1\}$ be the positive and negative class, $N_+^t = \sum_{C_i \in A_+^t} n_i$ and $N_-^t = \sum_{C_i \in A_-^t} n_i$ be the number of positive and negative examples, $|A_+|$ and $|A_-|$ be the number of subclasses in A_+ and A_-, respectively. To achieve the between-class balance, the overall importance of (examples in) A_+ and A_- is expected to have the same value, i.e., $max(N_+, N_-)$. To achieve within-class balance in A_+, every subclass is expected to have the same overall importance, i.e., $max(N_+, N_-)/|A_+|$. If C_i is among the subclasses, then the importance of each example of C_i is $max(N_+, N_-)/(|A_+|n_i)$. Similarly, to achieve the within-class balance of A_-, if C_i is among the subclasses, then the importance of each example of C_i is $max(N_+, N_-)/(|A_-|n_i)$.

Formally, let the example importance factor be $\mathbf{c}^t = [c_1^t, c_2^t, \ldots, c_k^t]$ with c_i^t indicating the importance of an example of class C_i in the t-th dichotomy, which is determined as follows:

$$c_i^t = \frac{\max(N_+^t, N_-^t)}{|A_z^t| n_i}, \tag{2}$$

with

$$z = \arg_{z \in \{+,-\}} C_i \in A_z^t.$$

Thus, the between-class balance and the multi-class balance are achieved simultaneously.

BWC-weighting is a general method that can invoke different class-imbalance learning methods[2]. For example, various over-sampling or under-sampling can be used by making class sizes proportional to their class importance after sampling, that is,

$$\frac{n_i^t}{n_j^t} = \frac{n_i c_i^t}{n_j c_j^t}, \forall i, j = 1, \ldots, k, i \neq j. \tag{3}$$

where, n_i^t is the new class size of C_i in the t-th dichotomy. BWC-weighting can also invoke instance-weighting methods, such as C4.5CS [28], by directly using $\mathbf{c}^t$ as instance weights for examples of the original k classes in the t-th dichotomy. Other methods such as threshold-moving and cost-sensitive learning methods [32], [33] can also be used.

There are two straightforward reduction methods to handle imbalanced multi-class data: (1) Balance the original data and then perform ECOC on the balanced training set. (2) Reduce the imbalanced multi-class task to a series of binary-class subtasks, then apply a binary-class imbalance learning method to each dichotomy. The first solution suffers from fixed importance of each class, such that the between-class balance could not be achieved, leading to under-performed dichotomies with great chance. As for the second solution, when a minority class merges with some other majority classes in a dichotomy, it becomes small disjuncts and tends to be greatly overwhelmed. Experimental results in Section V verify their inferiority to imECOC.

3) Decoding Stage: Under-performed classifier would fail to learn dichotomy concept or fail to recognize the original minority class examples in the learning process as discussed in (D3) in Section IV-A. Inadequate dichotomy code could also result in under-performed classifier. No matter what the cause is, the under-performed classifiers can be handled in the same way. Different classifiers should play different roles in the final prediction to achieve the best classification ability. Therefore, it is reasonable to assign a weight to each dichotomy to represent its classifier's importance (or reliability). The better the performance of a dichotomy classifier, the higher weight it is expected to receive. Since the code bit corresponding to a better classifier is also expected to contribute more in recovering the class assignments from the codes, the weights here are further used to calculate the weighted distance for decoding. Thus, the difficulties of the inadequate codes and the under-performed classifiers are solved in the unified strategy.

[2]Sampling, instance-weighting and threshold-moving methods can be formalized to a general method using example importance. Please refer to [33].

There are some weighted decoding methods designed for ECOC [10], [22]. All these methods predefine the weights and aim to maximize accuracy. Different from them, imECOC learns the optimal dichotomy weights by minimizing a loss function in favor of the minority class examples. This overcomes the difficulty of decoding.

Define the t-th *bit distance* of example $\mathbf{x}$ to class C_r as

$$b_t(\mathbf{x}, r) = d_{bit}(f_t(\mathbf{x}), \mathbf{M}(r, t)), \tag{4}$$

which measures the distance between the real-valued output of the dichotomy classifier f_t and class C_r's dichotomy code $\mathbf{M}(r, t)$, where $d_{bit} : \mathbb{R} \times \{-1, 0, +1\} \to \mathbb{R}$ is some distance function of two scalars. In imECOC, we define the bit distance function as:

$$d_{bit}(u, v) = \frac{1 - uv}{2}. \tag{5}$$

It is different from a bit of Hamming distance in Eq.(1) since $\text{sign}(\cdot)$ function is not used. Let

$$\mathbf{b}(\mathbf{x}, r) = (b_1(\mathbf{x}, r), \ldots, b_\ell(\mathbf{x}, r))^{\mathrm{T}} \tag{6}$$

be the bit distance vector of example $\mathbf{x}$ and class C_r, $\mathbf{w} \in \mathbb{R}^{\ell \times 1}$ be the dichotomy weight vector, then the weighted distance between a test instance $\mathbf{x}$ and class C_r in the code space is $\mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, r)$, and the class with the minimal distance is assigned to $\mathbf{x}$:

$$\hat{y} = \arg\min_r \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, r). \tag{7}$$

If example $\mathbf{x}$'s true class label is y, with $y \in \{1, \ldots, k\}$, the weighed distance of $\mathbf{x}$ to class C_y should be smaller than the distance to any other class, that is, $\mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, y) < \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, r), \forall r \neq y$. Since the prediction is the class with the minimal weighted distance, the difference between the minimal distance to any incorrect class and the distance to the true class indicates the confidence of prediction. Similar to large margin methods, we define the *margin* of example $\mathbf{x}$ to represent the confidence of prediction:

$$margin(\mathbf{x}) = \min\left(\mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, r)\right) - \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, y), \forall r \neq y.$$

When $\mathbf{x}$ is correctly classified, $margin(\mathbf{x}) > 0$. When $\mathbf{x}$ is misclassified, $margin(\mathbf{x}) < 0$. The larger the margin, the better the classification. By introducing the slack variable ξ, $\xi \geq 0$, regarding the loss of prediction, the above margin constrains can be formulated as:

$$\min\left(\mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, r) - \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, y)\right) \geq \Delta(y, r) - \xi, \forall r \neq y,$$

where, $\Delta(y, r)$ is the reference distance between C_y and C_r. The value is suggested to be 1 when there is no domain knowledge.

In class-imbalance learning, the importance factors of the examples in different classes are not the same. The loss of misclassifying a minority class example should be larger than that of a majority class example. Define γ_i as example $\mathbf{x}_i$'s weight, which is proportional to the inverse of its class size:

$$\gamma_i = \frac{\max_j(n_j)}{n_{y_i}}. \tag{8}$$

Then, the weighted loss of example $\mathbf{x}_i$ is $\gamma_i\xi_i$, and the overall loss of the training set is $\sum_{i=1}^{n} \gamma_i\xi_i$.

Algorithm 1 The imECOC Algorithm

Require: Training set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, y = \{1, \ldots, k\}$; the λ parameter in Eq. 9; a binary-class learning algorithm L.

%Training

generate code matrix: $\mathbf{M} \in \{-1, 0, +1\}^{k \times \ell}$;
for $t = 1, ..., \ell$ **do**
  $D^t = \emptyset$
  for $i = 1, ..., n$ **do**
    if $\mathbf{M}(y_i, t) \neq 0$ **then**
      $D^t = D^t \cup (\mathbf{x}_i, \mathbf{M}(y_i, t))$
    end if
  end for
  calculate the example weights $\mathbf{c}^t$ for the t-th dichotomy according to Eq. 2.
  $f_t \leftarrow L(D^t, \mathbf{c}^t)$: learn a classifier using the weighted examples
  calculate the bit distance vector for the training set, $\mathbf{b}(\mathbf{x}_i, r), \forall \mathbf{x}_i \in D, \forall r = 1, ..., k$, according to Eq. 4, 5, 6
  obtain the optimal dichotomy weights $\mathbf{w}$ according to Eq. 9
end for

%Testing

for a test instance $\mathbf{x}$, calculate the bit distance vector $\mathbf{b}(\mathbf{x}, r), \forall r = 1, ..., k$ according to Eq. 4, 5, 6
Output: $\hat{y}(\mathbf{x}) = \arg\min_r \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}, r)$ (Eq. 7)

The problem of optimizing dichotomy weights is formulated as follows:

$$\begin{aligned} &\min_{\mathbf{w}, \xi_i} ||\mathbf{w}||^2 + \lambda \sum_{i=1}^{n} \gamma_i\xi_i \\ s.t.\ &\min\left(\mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}_i, r) - \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}_i, y_i)\right) \geq \Delta(y_i, r) - \xi_i, \forall r \neq y_i \\ &\xi_i \geq 0 \end{aligned}$$

where, λ is the penalty parameter, and $||\mathbf{w}||^2$ is a regularizer to prevent over-fitting. This optimization problem is equivalent to:

$$\begin{aligned} &\min_{\mathbf{w}, \xi_i} ||\mathbf{w}||^2 + \lambda \sum_{i=1}^{n} \gamma_i\xi_i \\ s.t.\ &\mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}_i, r) - \mathbf{w}^{\mathrm{T}}\mathbf{b}(\mathbf{x}_i, y_i) \geq \Delta(y_i, r) - \xi_i, \\ &\qquad\qquad \forall r = 1, \ldots, k \\ &\xi_i \geq 0 \end{aligned} \tag{9}$$

It can be solved by quadratic programming. The pseudo code of imECOC is shown in Algorithm 1.

V. Experiments

We compare imECOC with two series of methods. Series A includes 6 methods based on ECOC, and Series B includes 3 state-of-the-art methods designed for handling imbalanced multi-class data.

(A1) **ECOC**: standard ECOC.

(A2) **Under**: under-sampling + ECOC. All classes except the smallest class are under-sampled to have the same size as the smallest class. Then ECOC is applied to the balanced data.

(A3) **Over**: over-sampling + ECOC. All classes except the largest class are over-sampled using random over-sampling

with replacement to have the same size as the largest class. Then ECOC is applied to the balanced data.

(A4) **SMOTE**: SMOTE + ECOC. All classes except the largest class are over-sampled using SMOTE to have the same size as the largest class. Then ECOC is applied to the balanced data.

(A5) **ECOChddt**: ECOC + HDDT. In each dichotomy of ECOC, HDDT [8] is used for imbalanced binary-class subtask, which is a decision tree method with a novel splitting criteria for imbalanced binary-class data. The method is extended to a multi-class method MC-HDDT [16]. Since their experiments showed ECOC + HDDT performs better than MC-HDDT, we do not compare with MC-HDDT in addition.

(A6) **imECOC**: our method. In the learning process, instance-weighting is used to handle between-class and within-class imbalance. The weighted examples are further used for training by CART [5]. λ is set to 1.

(B1) **DyS**: a multi-class imbalance learning method using dynamic sampling in neural networks [19]. The number of nodes is selected from $\{5, 10, 15, 20\}$ and the number of iterations is selected from $\{100, 500, 1000\}$ as suggested in [19].

(B2) **SMB**: SMOTEBoost [7]. It is a well known boosting-based class-imbalance learning method. Many previous work showed SMOTEBoost is very effective for binary-class imbalance problem. It can also be used for multi-class imbalance problem. The default settings are used.

(B3) **OvNC**: a sampling based boosting method for multi-class imbalance learning [29]. The default settings are used: λ is 9 and the number of ensembles is 51.

For fair comparison, methods in Series A except imECOC use CART [5] with default settings as the base learning method, and use Hamming distance for decoding. We exploit 3 problem-independent coding methods for methods in Series A, including OVA, Dense and Sparse codes[3]. The code length of OVA is k, the number of classes. To get Dense and Sparse codes, the default parameters are used. The length of Dense code is $10log_2(k)$, and the -1 and $+1$ code symbols are randomly generated with equal probability of 0.5. The length of Sparse code is $15log_2(k)$, and the -1, $+1$, 0 code symbols are randomly generated with probability of 0.25, 0.25 and 0.5, respectively.

14 UCI data sets [3] are used, with information shown in Table I. The data sets ended up with a number are derived from the original data sets by merging some classes or sampling from the original classes. *It should be noted that, most of the 14 data sets have few examples in the smallest class, and the the maximal imbalance rate is very large (generally larger than 10). These data sets are selected to reflect the common case of imbalanced multi-class data in real-world.*

Since "solar4" has only 4 classes, Dense code can not be obtained, and complete code (the code length is $2^{(k-1)} - 1$) is used instead. For each data set, we randomly select 50% data as the training set and the other 50% data as the test set, with class distributions in the training and test set being the same. The process is repeated 20 times and the average results are recorded.

[3]We also run experiments on OVO codes. The results are similar to that of OVA. Due to page limitation, we will show the results in a longer version.

TABLE I. INFORMATION OF DATA SETS. IN "CLASS SIZES", $x * n$ MEANS THERE ARE n CLASSES HAVE x EXAMPLES, AND THE MAXIMAL IMBALANCE RATE IS SHOWN IN "[ ]".

Dataset	Size	Class	Class Sizes
Glass	214	6	9 13 17 29 70 76 [8]
ecoli6	336	6	9 20 35 52 77 143 [72]
dermatology	366	6	20 49 52 61 72 112 [6]
soybean	683	19	8 14 15 16 20*9 44*2 88 91*2 92 [11]
vowel1	990	11	9 18 27 36 45 54 63 72 81 90 90 [10]
solar4	1372	4	20 40 141 1171 [58]
yeast	1484	10	5 20 30 35 44 51 163 244 429 463 [93]
segment1	669	7	10 20 20 41 83 165 330 [33]
segment2	1090	7	10 10 40 40 330 330 330 [33]
satimage	6435	6	626 703 707 1358 1508 1533 [2]
satimage1	2308	6	25 50 100 200 400 1533 [61]
satimage2	3341	6	50 50 100 100 1508 1533 [30]
isolet1	2450	26	10*2 20 *2 30*2 40*2 50*2 60*2 70*2 80*2 90*2 100*2 150*3 300*3 [30]
isolet2	2050	26	20*10 50*10 150*3 300*3 [15]

We adopt multi-class F-measure, G-mean [26] and MAUC [13] as evaluation measures. Let $cm(i, j)$ be the number of examples of C_i being classified to C_j, then the precision P_i and recall R_i of C_i are defined as:

$$P_i = \frac{cm(i,i)}{\sum_{j=1}^{k} cm(j,i)}, \;\; R_i = \frac{cm(i,i)}{n_i}.$$

The F-measure of C_i and the multi-class F-measure (F1) are:

$$F_i = \frac{2P_iR_i}{P_i + R_i}, \;\; \text{F-measure} = \frac{1}{k}\sum\nolimits_{i=1}^{k} F_i.$$

G-mean is defined as :

$$\text{G-mean} = \left(\prod\nolimits_{i=1}^{k} R_i\right)^{1/k}.$$

And MAUC is defined as:

$$\text{MAUC} = \frac{2}{k(k-1)} \sum_{i<j} \hat{A}(i,j)$$

$$\hat{A}(i,j) = (\hat{A}(i|j) + \hat{A}(j|i))/2,$$

where, $\hat{A}(i|j)$ is the probability that a randomly drawn member of class i will have a lower estimated probability of belonging to class j than a randomly drawn member of class j.

A poor performance of a class will decrease G-mean value most, and decrease F1 and MAUC values less. When most of the classes are the majority classes, the bad AUC values of the minority classes can be largely overwhelmed by many other good AUC results of the majority classes. So, among the 3 measures, G-mean is most sensitive to the minority class performance, and MAUC is least sensitive.

Table II, III and IV shows the F1, G-mean and MAUC results, respectively. For the convenience of comparison, we list the results of methods in Series B next to the results of methods in Series A. The best results are in bold face. We also carry out the paired t-tests and sign tests both with significance level 95% to compare the methods. The win/tie/lose counts of imECOC over the other methods are indicated by the rows named "w/t/l". Bold face implies the sign test is significant. And Fig. 1 shows some results of the F-measure values of all

TABLE II. F1 RESULTS. THE BEST RESULTS ARE HIGHLIGHTED BY BOLDFACE (PAIRED t-TESTS WITH 95% SIGNIFICANCE LEVEL, AND SIGN-TESTS WITH 95% SIGNIFICANCE LEVEL).

	F1	ECOC	Under	Over	SMOTE	ECOChddt	imECOC	DyS	SMB	OvNC
OVA	glas	.469 ± .052	.287 ± .088	.521 ± .075	.496 ± .051	.532 ± .050	**.774 ± .025**	.617 ± .051	.673 ± .064	.591 ± .047
	ecol	.630 ± .043	.447 ± .071	.635 ± .040	.636 ± .037	.642 ± .033	**.868 ± .017**	.752 ± .042	.761 ± .032	.706 ± .033
	derm	.865 ± .041	.734 ± .078	.846 ± .037	.836 ± .043	.872 ± .034	**.963 ± .005**	.878 ± .094	.859 ± .288	.940 ± .016
	soyb	.880 ± .018	.537 ± .055	.872 ± .015	.887 ± .015	.875 ± .021	.915 ± .004	.937 ± .018	**.956 ± .010**	.930 ± .019
	vowe	.517 ± .027	.235 ± .054	.534 ± .039	.526 ± .035	.567 ± .032	**.861 ± .013**	.719 ± .031	.784 ± .039	.536 ± .049
	sola	.275 ± .022	.149 ± .051	.246 ± .021	.230 ± .021	.276 ± .017	**.486 ± .043**	.276 ± .026	.282 ± .020	.295 ± .027
	yeas	.409 ± .025	.144 ± .040	.371 ± .029	.357 ± .019	.392 ± .022	**.819 ± .022**	.523 ± .026	.538 ± .027	.498 ± .023
	seg1	.769 ± .046	.436 ± .069	.741 ± .047	.750 ± .034	.771 ± .045	**.971 ± .017**	.862 ± .041	.895 ± .035	.876 ± .039
	seg2	.794 ± .033	.416 ± .072	.736 ± .026	.751 ± .028	.796 ± .031	**.986 ± .013**	.874 ± .031	.907 ± .022	.861 ± .026
	sati	.805 ± .008	.772 ± .008	.806 ± .008	.800 ± .010	.816 ± .008	**.977 ± .004**	.845 ± .012	.883 ± .006	.860 ± .008
	sat1	.784 ± .014	.587 ± .052	.774 ± .014	.779 ± .021	.803 ± .016	**.968 ± .006**	.859 ± .023	.890 ± .024	.893 ± .03
	sat2	.773 ± .013	.594 ± .036	.761 ± .016	.755 ± .013	.798 ± .014	**.978 ± .009**	.749 ± .023	.876 ± .016	.860 ± .014
	iso1	.492 ± .017	.175 ± .019	.501 ± .018	.521 ± .023	.643 ± .018	**1.000 ± .001**	.008 ± .000	.843 ± .016	.820 ± .015
	iso2	.447 ± .015	.235 ± .023	.443 ± .017	.474 ± .023	.614 ± .020	**.999 ± .001**	.010 ± .001	.841 ± .017	.834 ± .021
	avg	.636	.411	.628	.628	.671	**.898**	.636	.785	.750
	w/t/l	**14/0/0**	**14/0/0**	**14/0/0**	**14/0/0**	**14/0/0**	–	**13/0/1**	**12/1/1**	**13/0/1**
Sparse	glas	.632 ± .053	.517 ± .072	.659 ± .061	.659 ± .054	.629 ± .058	**.823 ± .005**	.617 ± .051	.673 ± .064	.591 ± .047
	ecol	.749 ± .048	.680 ± .039	.726 ± .050	.738 ± .045	.724 ± .050	**.913 ± .008**	.752 ± .042	.761 ± .032	.706 ± .033
	derm	.955 ± .023	.915 ± .036	.942 ± .024	.942 ± .032	.941 ± .024	**.970 ± .000**	.878 ± .094	.859 ± .288	.940 ± .016
	soyb	.946 ± .014	.814 ± .043	.950 ± .011	.949 ± .012	.936 ± .020	.927 ± .002	.937 ± .018	**.956 ± .010**	.930 ± .019
	vowe	.678 ± .034	.396 ± .040	.695 ± .025	.697 ± .051	.677 ± .043	**.894 ± .000**	.719 ± .031	.784 ± .039	.536 ± .049
	sola	.259 ± .015	.220 ± .040	.252 ± .019	.253 ± .017	.281 ± .021	**.547 ± .043**	.276 ± .026	.282 ± .020	.295 ± .027
	yeas	.498 ± .056	.252 ± .063	.510 ± .028	.493 ± .030	.477 ± .031	**.895 ± .010**	.523 ± .026	.538 ± .027	.498 ± .023
	seg1	.857 ± .071	.680 ± .061	.844 ± .053	.850 ± .057	.851 ± .054	**.985 ± .002**	.862 ± .041	.895 ± .035	.876 ± .039
	seg2	.893 ± .037	.664 ± .055	.884 ± .025	.877 ± .027	.886 ± .030	**1.000 ± .000**	.874 ± .031	.907 ± .022	.861 ± .026
	sati	.866 ± .009	.848 ± .012	.866 ± .010	.868 ± .008	.871 ± .009	**.996 ± .001**	.845 ± .012	.883 ± .006	.860 ± .008
	sat1	.840 ± .041	.683 ± .052	.879 ± .026	.886 ± .020	.878 ± .036	**.980 ± .001**	.859 ± .023	.890 ± .024	.893 ± .03
	sat2	.838 ± .027	.691 ± .040	.840 ± .019	.845 ± .019	.854 ± .021	**1.000 ± .001**	.749 ± .023	.876 ± .016	.860 ± .014
	iso1	.719 ± .019	.386 ± .022	.731 ± .023	.760 ± .019	.760 ± .016	**1.000 ± .000**	.008 ± .000	.843 ± .016	.820 ± .015
	iso2	.682 ± .035	.557 ± .035	.708 ± .037	.748 ± .028	.736 ± .027	**1.000 ± .000**	.010 ± .001	.841 ± .017	.834 ± .021
	avg	.744	.593	.749	.755	.750	**.924**	.636	.785	.750
	w/t/l	**13/0/1**	**14/0/0**	**13/0/1**	**13/0/1**	**13/1/0**	–	**13/0/1**	**12/1/1**	**13/1/0**
Dense	glas	.622 ± .058	.532 ± .056	.644 ± .064	.641 ± .061	.644 ± .051	**.825 ± .004**	.617 ± .051	.673 ± .064	.591 ± .047
	ecol	.767 ± .041	.656 ± .053	.741 ± .043	.764 ± .054	.739 ± .038	**.919 ± .005**	.752 ± .042	.761 ± .032	.706 ± .033
	derm	.957 ± .016	.912 ± .028	.950 ± .027	.952 ± .023	.945 ± .021	**.970 ± .000**	.878 ± .094	.859 ± .288	.940 ± .016
	soyb	.945 ± .016	.831 ± .024	.953 ± .010	.952 ± .010	.927 ± .024	.928 ± .001	.937 ± .018	**.956 ± .010**	.930 ± .019
	vowe	.688 ± .037	.410 ± .037	.720 ± .032	.728 ± .043	.710 ± .046	**.894 ± .000**	.719 ± .031	.784 ± .039	.536 ± .049
	sola	.274 ± .018	.190 ± .048	.248 ± .019	.239 ± .022	.276 ± .015	**.511 ± .037**	.276 ± .026	.282 ± .020	.295 ± .027
	yeas	.474 ± .041	.247 ± .048	.488 ± .038	.470 ± .030	.436 ± .022	**.880 ± .015**	.523 ± .026	.538 ± .027	.498 ± .023
	seg1	.885 ± .042	.674 ± .055	.867 ± .043	.856 ± .050	.876 ± .044	**.986 ± .000**	.862 ± .041	.895 ± .035	.876 ± .039
	seg2	.908 ± .035	.586 ± .068	.873 ± .031	.868 ± .039	.894 ± .027	**1.000 ± .000**	.874 ± .031	.907 ± .022	.861 ± .026
	sati	.877 ± .006	.858 ± .009	.877 ± .007	.878 ± .006	.883 ± .005	**.998 ± .000**	.845 ± .012	.883 ± .006	.860 ± .008
	sat1	.840 ± .028	.709 ± .053	.879 ± .018	.886 ± .025	.882 ± .027	**.981 ± .000**	.859 ± .023	.890 ± .024	.893 ± .03
	sat2	.838 ± .021	.690 ± .025	.847 ± .018	.852 ± .016	.860 ± .018	**1.000 ± .000**	.749 ± .023	.876 ± .016	.860 ± .014
	iso1	.732 ± .016	.391 ± .026	.756 ± .022	.791 ± .015	.765 ± .022	**1.000 ± .000**	.008 ± .000	.843 ± .016	.820 ± .015
	iso2	.691 ± .023	.554 ± .027	.721 ± .024	.774 ± .026	.730 ± .031	**1.000 ± .000**	.010 ± .001	.841 ± .017	.834 ± .021
	avg	.750	.589	.755	.761	.755	**.921**	.636	.785	.750
	w/t/l	**13/0/1**	**14/0/0**	**13/0/1**	**13/0/1**	**13/1/0**	–	**13/1/0**	**12/1/1**	**13/1/0**

individual classes[4]. The x-axis indicates the class sizes, which are sorted in ascending order.

imECOC's F1 and G-mean values are the best on all data sets except soybean. It significantly outperforms all the other methods no matter what code is used. imECOC also achieves the best MAUC except on 2 data sets.

The improvement on G-mean of imECOC over the other methods is most significant, followed by the improvements of F1 and MAUC values. Many methods have very low or even near 0 G-mean values on some data sets, while imECOC's G-mean values are all above 0.82. This indicates imECOC is very good at detecting classes which are very difficult to be classified correctly (usually the minority classes), while not harming the performances of the other classes much. Fig. 1 shows imECOC's advantages more clearly. Except imECOC, all the other methods in the figure generally perform worse when the class size is smaller, especially when there are only few examples. These results clearly show that imECOC is the best method on handling the minority class examples.

For F1 and G-mean, Sparse and Dense codes are the best two codes for imECOC, which perform very similarly. For MAUC, OVA, Sparse and Dense codes perform very similarly to imECOC. When time efficiency is important, OVA is also a good choice, since imECOC with OVA code is significantly better than all the other methods (with no matter what codes) on all of F1, G-mean and MAUC measures.

Among the other methods in Series A, since the smallest class has very few examples in general, it is not surprising that Under is the worst method since it ignores too much information. Even standard ECOC is significantly better than it. Over, SMOTE and ECOChddt are all comparable to ECOC on F1 measure. But they are better at detecting the minority classes, which is implied by the G-mean results. SMOTE is the most effective one among them. On MAUC values, only SMOTE is better than ECOC, Over and ECOChddt are comparable to ECOC. HDDT is very effective for binary-class imbalance problems. But its failure shows it is useful only for handling between-class imbalance, there is a big gap between handling the bilateral class-imbalance and handling

[4]For the seek of clearness, we only show the results of some typical methods.

TABLE III. G-MEAN RESULTS.THE BEST RESULTS ARE HIGHLIGHTED BY BOLDFACE (PAIRED t-TESTS WITH 95% SIGNIFICANCE LEVEL, AND SIGN-TESTS WITH 95% SIGNIFICANCE LEVEL).

	G-mean	ECOC	Under	Over	SMOTE	ECOChddt	imECOC	DyS	SMB	OvNC
OVA	glas	.325 ± .224	.180 ± .199	.456 ± .217	.492 ± .178	.480 ± .214	**.821 ± .018**	.491 ± .222	.570 ± .212	.571 ± .205
	ecol	.649 ± .069	.371 ± .206	.680 ± .050	.680 ± .048	.681 ± .051	**.906 ± .012**	.735 ± .055	.730 ± .052	.696 ± .048
	derm	.890 ± .044	.762 ± .084	.873 ± .037	.863 ± .043	.896 ± .034	**.966 ± .006**	.808 ± .278	.860 ± .295	.938 ± .020
	soyb	.880 ± .030	.000 ± .000	.885 ± .023	.905 ± .025	.876 ± .034	.920 ± .003	.937 ± .020	**.951 ± .012**	.936 ± .026
	vowe	.459 ± .115	.025 ± .079	.506 ± .048	.505 ± .046	.511 ± .129	**.874 ± .014**	.657 ± .158	.761 ± .045	.486 ± .177
	sola	.026 ± .063	.203 ± .060	.152 ± .108	.267 ± .041	.052 ± .082	**.800 ± .019**	.135 ± .120	.207 ± .082	.081 ± .115
	yeas	.076 ± .156	.009 ± .040	.227 ± .193	.289 ± .173	.196 ± .202	**.878 ± .011**	.045 ± .139	.297 ± .251	.380 ± .227
	seg1	.784 ± .074	.280 ± .221	.750 ± .068	.778 ± .054	.780 ± .081	**.981 ± .009**	.866 ± .052	.869 ± .059	.864 ± .059
	seg2	.836 ± .049	.413 ± .133	.763 ± .039	.813 ± .042	.844 ± .047	**.996 ± .004**	.909 ± .015	.887 ± .035	.891 ± .039
	sati	.807 ± .008	.781 ± .008	.810 ± .009	.807 ± .011	.820 ± .009	**.982 ± .002**	.834 ± .024	.868 ± .007	.862 ± .009
	sat1	.812 ± .041	.673 ± .051	.826 ± .026	.853 ± .028	.852 ± .033	**.983 ± .002**	.907 ± .022	.848 ± .042	.866 ± .047
	sat2	.781 ± .030	.748 ± .032	.783 ± .026	.827 ± .027	.834 ± .029	**.997 ± .001**	.866 ± .019	.825 ± .026	.875 ± .024
	iso1	.020 ± .090	.000 ± .000	.083 ± .170	.239 ± .246	.260 ± .296	**1.000 ± .000**	.000 ± .000	.509 ± .384	.741 ± .261
	iso2	.000 ± .000	.020 ± .062	.034 ± .105	.257 ± .216	.298 ± .277	**1.000 ± .001**	.000 ± .000	.797 ± .023	.842 ± .019
	avg	.525	.319	.559	.612	.599	**.936**	.585	.713	.716
	w/t/l	**14/0/0**	**14/0/0**	**14/0/0**	**14/0/0**	**14/0/0**	–	**13/0/1**	**12/1/1**	**13/0/1**
Sparse	glas	.519 ± .190	.523 ± .088	.514 ± .270	.637 ± .077	.485 ± .217	**.852 ± .003**	.491 ± .222	.570 ± .212	.571 ± .205
	ecol	.696 ± .068	.679 ± .066	.656 ± .168	.704 ± .068	.648 ± .165	**.937 ± .007**	.735 ± .055	.730 ± .052	.696 ± .048
	derm	.951 ± .029	.914 ± .041	.934 ± .030	.937 ± .044	.937 ± .028	**.973 ± .000**	.808 ± .278	.860 ± .295	.938 ± .020
	soyb	.936 ± .019	.825 ± .052	.946 ± .014	.949 ± .015	.921 ± .032	.929 ± .002	.937 ± .020	**.951 ± .012**	.936 ± .026
	vowe	.538 ± .237	.333 ± .151	.568 ± .246	.652 ± .164	.446 ± .303	**.906 ± .000**	.657 ± .158	.761 ± .045	.486 ± .177
	sola	.000 ± .000	.274 ± .085	.083 ± .096	.257 ± .035	.011 ± .047	**.833 ± .018**	.135 ± .120	.207 ± .082	.081 ± .115
	yeas	.020 ± .090	.090 ± .128	.133 ± .208	.348 ± .208	.082 ± .169	**.930 ± .007**	.045 ± .139	.297 ± .251	.380 ± .227
	seg1	.812 ± .103	.705 ± .075	.803 ± .074	.791 ± .198	.808 ± .075	**.990 ± .001**	.866 ± .052	.869 ± .059	.864 ± .059
	seg2	.820 ± .200	.765 ± .054	.853 ± .048	.870 ± .045	.855 ± .050	**1.000 ± .000**	.909 ± .015	.887 ± .035	.891 ± .039
	sati	.854 ± .018	.853 ± .010	.855 ± .016	.863 ± .011	.859 ± .015	**.997 ± .001**	.834 ± .024	.868 ± .007	.862 ± .009
	sat1	.735 ± .193	.823 ± .029	.839 ± .057	.861 ± .043	.822 ± .082	**.987 ± .000**	.907 ± .022	.848 ± .042	.866 ± .047
	sat2	.758 ± .067	.869 ± .023	.770 ± .054	.822 ± .046	.790 ± .052	**1.000 ± .000**	.866 ± .019	.825 ± .026	.875 ± .024
	iso1	.035 ± .154	.254 ± .193	.169 ± .301	.283 ± .356	.102 ± .250	**1.000 ± .000**	.000 ± .000	.509 ± .384	.741 ± .261
	iso2	.148 ± .265	.556 ± .136	.435 ± .297	.526 ± .313	.551 ± .241	**1.000 ± .000**	.000 ± .000	.797 ± .023	.842 ± .019
	avg	.559	.605	.611	.679	.594	**.952**	.585	.713	.716
	w/t/l	**13/1/0**	**14/0/0**	**13/0/1**	**13/0/1**	**13/1/0**	–	**13/1/0**	**12/1/1**	**13/1/0**
Dense	glas	.567 ± .152	.481 ± .216	.568 ± .203	.650 ± .070	.588 ± .148	**.854 ± .001**	.491 ± .222	.570 ± .212	.571 ± .205
	ecol	.719 ± .054	.676 ± .055	.675 ± .165	.746 ± .075	.704 ± .064	**.941 ± .004**	.735 ± .055	.730 ± .052	.696 ± .048
	derm	.955 ± .017	.923 ± .026	.949 ± .029	.954 ± .020	.940 ± .026	**.973 ± .000**	.808 ± .278	.860 ± .295	.938 ± .020
	soyb	.936 ± .021	.850 ± .028	.950 ± .011	**.954 ± .010**	.912 ± .039	.931 ± .001	.937 ± .020	.951 ± .012	.936 ± .026
	vowe	.563 ± .246	.391 ± .100	.670 ± .161	.726 ± .045	.553 ± .287	**.906 ± .000**	.657 ± .158	.761 ± .045	.486 ± .177
	sola	.032 ± .058	.281 ± .052	.107 ± .112	.260 ± .081	.047 ± .075	**.822 ± .017**	.135 ± .120	.207 ± .082	.081 ± .115
	yeas	.044 ± .136	.144 ± .139	.180 ± .228	.394 ± .204	.022 ± .097	**.935 ± .004**	.045 ± .139	.297 ± .251	.380 ± .227
	seg1	.847 ± .073	.705 ± .060	.839 ± .063	.840 ± .068	.851 ± .059	**.990 ± .000**	.866 ± .052	.869 ± .059	.864 ± .059
	seg2	.881 ± .060	.723 ± .075	.847 ± .048	.874 ± .060	.872 ± .042	**1.000 ± .000**	.909 ± .015	.887 ± .035	.891 ± .039
	sati	.864 ± .008	.865 ± .009	.866 ± .009	.872 ± .008	.870 ± .007	**.998 ± .000**	.834 ± .024	.868 ± .007	.862 ± .009
	sat1	.733 ± .185	.837 ± .034	.847 ± .032	.867 ± .043	.827 ± .053	**.987 ± .000**	.907 ± .022	.848 ± .042	.866 ± .047
	sat2	.775 ± .039	.877 ± .016	.781 ± .036	.846 ± .020	.791 ± .035	**1.000 ± .000**	.866 ± .019	.825 ± .026	.875 ± .024
	iso1	.035 ± .156	.311 ± .162	.144 ± .296	.456 ± .383	.145 ± .298	**1.000 ± .000**	.000 ± .000	.509 ± .384	.741 ± .261
	iso2	.326 ± .303	.595 ± .031	.607 ± .147	.736 ± .038	.577 ± .203	**1.000 ± .000**	.000 ± .000	.797 ± .023	.842 ± .019
	avg	.591	.618	.645	.727	.621	**.953**	.585	.713	.716
	w/t/l	**13/1/0**	**14/0/0**	**13/0/1**	**13/0/1**	**14/0/0**	–	**13/1/0**	**12/1/1**	**13/1/0**

the multilateral class-imbalance of the original classification. It also suggests that appropriately handling the difficulty of (D2) is important. So, SMOTE is better than the other popular sampling methods when handling imbalanced multi-class data.

Among the 3 methods in Series B, SMB is the best on F1 and MAUC measures. It is significantly better than all the other methods except imECOC. The 3 methods are comparable to each other on G-mean measure. DyS is the second best method. The poor performance on data sets "isolet1" and "isolet2" is probably because it is very hard to determine the number of hidden nodes and the number of iterations when the number of classes is very large. DyS is comparable to OvNC on F1 and G-mean measures, but better than OvNC on MAUC. Compared with methods in Series A, DyS and OvNC are generally comparable to Over, SMOTE and ECOChttd, especially when Sparse and Dense codes are used. The performances of Over, SMOTE and ECOChttd with Sparse and Dense code are competitive.

The conclusions include: (1) imECOC beats all the compared methods greatly on all of F1, G-mean and MAUC measures. (2) Even with the shortest code OVA, imECOC still performs significantly better than any other methods. (3) Sparse and Dense codes are recommended for imECOC. When time efficiency is important, OVA is also recommended. (4) imECOC is very good at classifying the classes difficult to recognize, especially the minority classes (with few examples). (5) SMOTEBoost is the second best method. It performs stable on different measures. (6) Over, SMOTE and ECOChttd are competitive when Sparse and Dense codes are used.

VI. CONCLUSION

Multi-class imbalance is much more challenging than binary-class imbalance. ECOC aims at maximizing accuracy, and to enable it to handle multi-class imbalance, we need to consider: (a) How to generate code matrix to identify the minority classes? (b) How to address the bilateral class-imbalance and the multilateral class-imbalance of the original task simultaneously? (c) How to deal with the under-performed classifiers? (d) How to design the decoding strategy in favor of the minority classes?

The proposed imECOC method tackles these difficulties by two key ingredients: (1) In the learning stage, each dichotomy's

TABLE IV. MAUC RESULTS. THE BEST RESULTS ARE HIGHLIGHTED BY BOLDFACE (PAIRED t-TESTS WITH 95% SIGNIFICANCE LEVEL, AND SIGN-TESTS WITH 95% SIGNIFICANCE LEVEL).

	MAUC	ECOC	Under	Over	SMOTE	ECOChddt	imECOC	DyS	SMB	OvNC
OVA	glas	.836 ± .043	.834 ± .056	.869 ± .037	.875 ± .029	.872 ± .029	**.959 ± .005**	.915 ± .025	.952 ± .013	.890 ± .030
	ecol	.919 ± .025	.862 ± .025	.926 ± .019	.927 ± .017	.920 ± .018	**.984 ± .002**	.967 ± .013	.967 ± .011	.927 ± .014
	derm	.991 ± .007	.980 ± .011	.989 ± .006	.989 ± .006	.990 ± .005	.994 ± .001	.993 ± .007	**.998 ± .002**	.993 ± .004
	soyb	.994 ± .002	.923 ± .023	.995 ± .002	.996 ± .002	.994 ± .002	.994 ± .001	**.999 ± .000**	**.999 ± .000**	.996 ± .002
	vowe	.875 ± .015	.742 ± .039	.880 ± .019	.888 ± .021	.900 ± .023	**.986 ± .001**	.977 ± .006	**.986 ± .005**	.884 ± .031
	sola	.540 ± .015	.537 ± .029	.524 ± .025	.570 ± .037	.540 ± .016	**.967 ± .005**	.623 ± .033	.593 ± .027	.576 ± .027
	yeas	.785 ± .029	.688 ± .034	.818 ± .020	.822 ± .018	.828 ± .015	**.992 ± .001**	**.922 ± .006**	**.922 ± .007**	.801 ± .012
	seg1	.963 ± .015	.894 ± .032	.950 ± .015	.963 ± .015	.965 ± .016	**.998 ± .000**	.987 ± .006	.991 ± .005	.983 ± .014
	seg2	.969 ± .008	.911 ± .031	.962 ± .010	.971 ± .011	.975 ± .009	**1.000 ± .000**	.994 ± .002	.994 ± .004	.988 ± .006
	sati	.962 ± .002	.960 ± .002	.963 ± .002	.963 ± .003	.966 ± .003	**1.000 ± .000**	.985 ± .001	.991 ± .001	.975 ± .002
	sat1	.963 ± .007	.929 ± .018	.967 ± .006	.971 ± .007	.972 ± .007	**.997 ± .001**	.994 ± .002	.994 ± .003	.989 ± .005
	sat2	.955 ± .007	.954 ± .009	.954 ± .007	.966 ± .007	.966 ± .007	**1.000 ± .000**	.991 ± .002	.994 ± .003	.986 ± .006
	iso1	.873 ± .008	.760 ± .017	.884 ± .008	.908 ± .010	.935 ± .009	**1.000 ± .000**	.500 ± .000	.997 ± .001	.991 ± .001
	iso2	.850 ± .011	.792 ± .015	.856 ± .011	.892 ± .012	.926 ± .008	**1.000 ± .000**	.500 ± .000	.997 ± .001	.991 ± .004
	avg	.891	.840	.895	.907	.911	**.991**	.882	.955	.926
	w/t/l	**12/2/0**	**14/0/0**	**13/1/0**	**13/0/1**	**13/1/0**	–	**12/1/1**	**11/1/2**	**12/2/0**
Sparse	glas	.937 ± .019	.892 ± .034	.946 ± .018	.942 ± .021	.938 ± .019	**.956 ± .012**	.915 ± .025	.952 ± .013	.890 ± .030
	ecol	.963 ± .012	.956 ± .014	.961 ± .012	.965 ± .011	.961 ± .012	**.985 ± .005**	.967 ± .013	.967 ± .011	.927 ± .014
	derm	**.998 ± .002**	.997 ± .002	**.998 ± .002**	**.998 ± .001**	.996 ± .002	.994 ± .001	.993 ± .007	**.998 ± .002**	.993 ± .004
	soyb	**.999 ± .000**	.995 ± .003	**.999 ± .000**	**.999 ± .000**	**.999 ± .000**	.993 ± .001	**.999 ± .000**	**.999 ± .000**	.996 ± .002
	vowe	.974 ± .004	.900 ± .011	.978 ± .004	.980 ± .005	.973 ± .007	**.987 ± .002**	.977 ± .006	.986 ± .005	.884 ± .031
	sola	.544 ± .032	.593 ± .033	.544 ± .031	.584 ± .029	.560 ± .032	**.973 ± .003**	.623 ± .033	.593 ± .027	.576 ± .027
	yeas	.903 ± .015	.834 ± .030	.909 ± .011	.907 ± .010	.900 ± .011	**.990 ± .002**	.922 ± .006	.922 ± .007	.801 ± .012
	seg1	.987 ± .009	.969 ± .011	.985 ± .009	.984 ± .009	.986 ± .009	**.998 ± .001**	.987 ± .006	.991 ± .005	.983 ± .014
	seg2	.995 ± .004	.976 ± .011	.994 ± .003	.992 ± .004	.992 ± .005	**1.000 ± .000**	.994 ± .002	.994 ± .004	.988 ± .006
	sati	.989 ± .001	.988 ± .001	.989 ± .001	.990 ± .002	.990 ± .001	**1.000 ± .000**	.985 ± .001	.991 ± .001	.975 ± .002
	sat1	.988 ± .004	.983 ± .005	.991 ± .005	.992 ± .004	.992 ± .004	**.998 ± .001**	.994 ± .002	.994 ± .003	.989 ± .005
	sat2	.987 ± .005	.989 ± .003	.990 ± .003	.992 ± .003	.991 ± .004	**1.000 ± .000**	.991 ± .002	.994 ± .003	.986 ± .006
	iso1	.990 ± .003	.954 ± .006	.993 ± .002	.994 ± .002	.994 ± .003	**1.000 ± .000**	.500 ± .000	.997 ± .001	.991 ± .001
	iso2	.989 ± .004	.984 ± .003	.991 ± .003	.993 ± .002	.993 ± .003	**1.000 ± .000**	.500 ± .000	.997 ± .001	.991 ± .004
	avg	.946	.929	.948	.951	.947	**.991**	.882	.955	.926
	w/t/l	**12/0/2**	**12/0/2**	**11/1/2**	**12/0/2**	**12/0/2**	–	**12/1/1**	**10/2/2**	**12/2/0**
Dense	glas	.932 ± .021	.912 ± .020	.935 ± .025	.936 ± .026	.939 ± .025	**.962 ± .007**	.915 ± .025	.952 ± .013	.890 ± .030
	ecol	.962 ± .013	.949 ± .015	.958 ± .013	.962 ± .013	.953 ± .011	**.985 ± .003**	.967 ± .013	.967 ± .011	.927 ± .014
	derm	**.999 ± .001**	.996 ± .003	.998 ± .002	**.999 ± .001**	.997 ± .002	.994 ± .001	.993 ± .007	.998 ± .002	.993 ± .004
	soyb	.998 ± .001	.994 ± .003	.998 ± .002	.998 ± .002	.997 ± .002	.993 ± .001	**.999 ± .000**	**.999 ± .000**	.996 ± .002
	vowe	.969 ± .008	.874 ± .025	.972 ± .009	.976 ± .008	.969 ± .011	**.986 ± .001**	.977 ± .006	**.986 ± .005**	.884 ± .031
	sola	.525 ± .022	.583 ± .045	.516 ± .037	.568 ± .031	.528 ± .018	**.972 ± .004**	.623 ± .033	.593 ± .027	.576 ± .027
	yeas	.866 ± .022	.804 ± .031	.884 ± .017	.890 ± .013	.844 ± .020	**.992 ± .001**	.922 ± .006	.922 ± .007	.801 ± .012
	seg1	.990 ± .007	.965 ± .013	.987 ± .009	.986 ± .009	.988 ± .008	**.998 ± .000**	.987 ± .006	.991 ± .005	.983 ± .014
	seg2	.991 ± .006	.969 ± .009	.992 ± .005	.991 ± .005	.993 ± .004	**1.000 ± .000**	.994 ± .002	.994 ± .004	.988 ± .006
	sati	.988 ± .001	.987 ± .002	.989 ± .001	.989 ± .001	.989 ± .001	**1.000 ± .000**	.985 ± .001	.991 ± .001	.975 ± .002
	sat1	.983 ± .004	.986 ± .006	.992 ± .003	.993 ± .003	.993 ± .004	**.997 ± .001**	.994 ± .002	.994 ± .003	.989 ± .005
	sat2	.982 ± .006	.990 ± .003	.990 ± .004	.992 ± .003	.990 ± .003	**1.000 ± .000**	.991 ± .002	.994 ± .003	.986 ± .006
	iso1	.988 ± .003	.944 ± .008	.990 ± .003	.993 ± .002	.990 ± .003	**1.000 ± .000**	.500 ± .000	.997 ± .001	.991 ± .001
	iso2	.985 ± .003	.981 ± .003	.989 ± .003	.993 ± .003	.989 ± .003	**1.000 ± .000**	.500 ± .000	.997 ± .001	.991 ± .004
	avg	.940	.924	.942	.948	.940	**.991**	.882	.955	.926
	w/t/l	**12/0/2**	**12/1/1**	**12/0/2**	**12/0/2**	**12/0/2**	0/14/0	**12/1/1**	**11/1/2**	**12/2/0**

between-class and within-class imbalances are both handled. (2) In the decoding stage, considering that the dichotomy classifiers should contribute differently to the final prediction, imECOC assigns weights to dichotomies to reflect their reliability, and uses the weighted distance to decode. The optimal dichotomy weights are obtained by minimizing a weighed loss in favor of the minority classes. Experimental results show that, imECOC is significantly better than many state-of-the-art multi-class imbalance learning methods, no matter whether F1, G-mean and MAUC are used as evaluation measures.

It is interesting to study whether imECOC can be improved by using problem-dependent codes instead of problem-independent codes.

ACKNOWLEDGEMENT

We want to thank the anonymous reviewers for helpful comments and suggestions. X.-Y. Liu was supported by NSFC (61105046), SRFDP (Specialized Research Fund for the Doctoral Program of Higher Education, by Ministry of Education, 20110092120029), and Open Foundation of National Key Laboratory for Novel Software Technology of China (KFKT2011B01). Z.-H. Zhou was supported by 973 Program (2010CB327903) and NSFC (61021062).

REFERENCES

[1] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2001.

[2] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva. Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):113–126, 2009.

[3] C. Blake, E. Keogh, and C. J. Merz. UCI repository of machine learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA.

[4] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(6):1145–1159, 1997.

[5] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. CRC Press, 1984.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

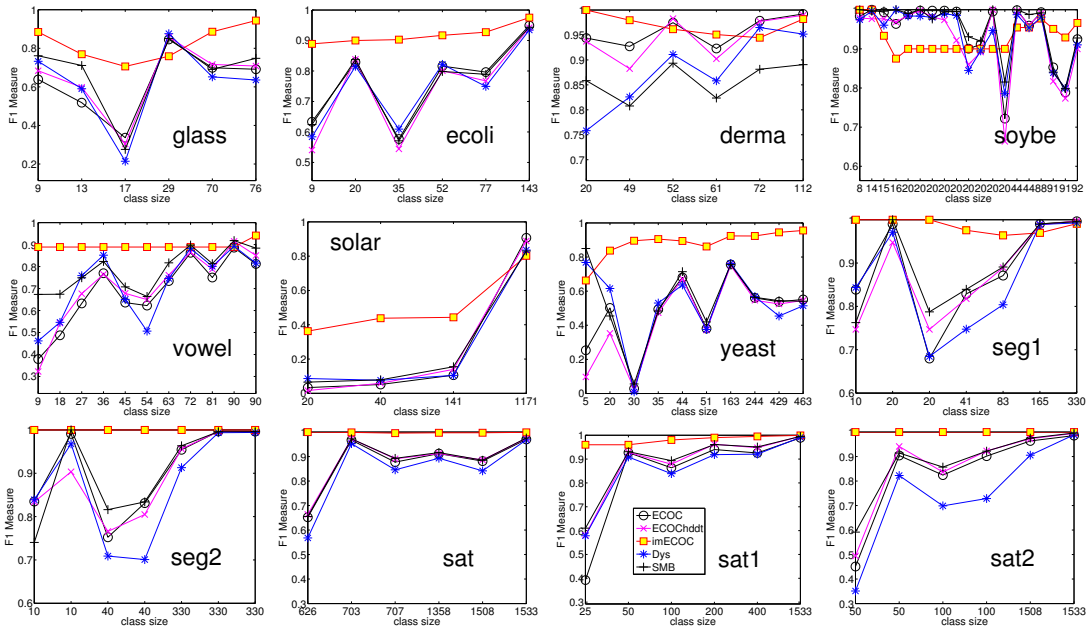

Fig. 1. F1 measure for each Class (methods in Series A are with Dense Code). The x- and y-axis indicates class size and F1 measure, respectively.

[7] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, Cavtat-Dubrovnik, Croatia, 2003.

[8] D. A. Cieslak and N. V. Chawla. Learning decision trees for unbalanced data. In *Proceedings of ECML/PKDD 2008*, pages 241–256, Antwerp, Belgium, 2008.

[9] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[10] S. Escalera, O. Pujol, and P. Radeva. On the decoding process in ternary error-correcting output codes. *IEEE Trans Pattern Analysis and Machine Intellegence*, 32(1):120–134, 2010.

[11] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: Misclassification cost-sensitive boosting. In *Proceedings of the 16th International Confernece on Machine Learning*, pages 97–105, Bled, Slovenia, 1999.

[12] A. Fernández, V. López, M. Galar, M. José del Jesus, and F. Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 2013.

[13] D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[14] H. He and E. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[15] H. He and Y. Ma, editors. *Imbalanced Learning: Foundations, Algorithms, and Applications*. IEEE Press, 2013.

[16] T. R. Hoens, Q. Qian, N. Chawla, and Z.-H. Zhou. Building decision trees for the multi-class imbalance problem. In *Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2012.

[17] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[18] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the 14th International Conference on Machine Learning*, pages 179–186, Nashville, TN, 1997.

[19] M. Lin, K. Tang, and X. Yao. Dynamic sampling approach to training neural networks for multiclass imbalance classification. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4):647–660, 2013.

[20] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 39(2):539–550, 2009.

[21] G. Ou and Y. L. Murphey. Multi-class pattern classification using neural networks. *Pattern Recognition*, 40(1):4–18, Jan. 2007.

[22] O. Pujol, S. Escalera, and P. Radeva. An incremental node embedding technique for error correcting output codes. *Pattern Recognition*, 41(2):713–725, Feb. 2008.

[23] G. Rätsch, A. J. Smola, and S. Mika. Adapting codes and embeddings for polychotomies. In *Advances in Neural Information Processing Systems*, volume 15, pages 513–520. 2002.

[24] C. P. Ronaldo, E. A. P. A. B. Gustavo, and C. M. Maria. Class imbalance versus class overlaping: an analysis of a learning system behavior. *Lecture Notes in Computer Science*, 2972:312–321, 2004.

[25] N. Singh-Miller and M. Collins. Learning label embeddings for nearest-neighbor multi-class classification with an application to speech recognition. In *Advances in Neural Information Processing Systems*, volume 22, pages 1678–1686. 2009.

[26] Y. Sun, M. S. Kamel, and Y. Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Proceedings of the 6th International Conference on Data Mining*, pages 592–602, 2006.

[27] A. C. Tan, D. Gilbert, and Y. Deville. Multi-class protein fold classification using a new ensemble machine learning approach. In *Proceedings of the 14th International Conference on Genome Informatics*, pages 206–217, Yokohama, Japan, 2003.

[28] K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

[29] S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B – Cybernetics*, 42(4):1119–1130, 2012.

[30] G. M. Weiss. Mining with rarity: A unifying framework. *ACM SIGKDD Explorations*, 6(1):7–19, 2004.

[31] Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.

[32] Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.

[33] Z.-H. Zhou and X.-Y. Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3):232–257, 2010.