Research Article

A Structural SVM Based Approach for Binary Classification under Class Imbalance

Fan Cheng,^{1,2} Kang Yang,² and Lei Zhang¹

¹Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, No. 3, Feixi Road, Hefei, Anhui 230039, China

²School of Computer, Anhui University, No. 3, Feixi Road, Hefei 230039, China

Correspondence should be addressed to Fan Cheng; chengfan@mail.ustc.edu.cn

Received 4 January 2015; Accepted 4 May 2015

Academic Editor: Haibo He

Copyright © 2015 Fan Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Class imbalance situations, where one class is rare compared to the other, arise frequently in machine learning applications. It is well known that the usual misclassification error is not suitable in such settings. A wide range of performance measures such as AM and QM have been proposed for this problem. However, due to computational difficulties, few learning techniques have been developed to directly optimize for AM or QM metric. To fill the gap, in this paper, we present a general structural SVM framework for directly optimizing AM and QM. We define the loss functions oriented to AM and QM, respectively, and adopt the cutting plane algorithm to solve the outer optimization. For the inner problem of finding the most violated constraint, we propose two efficient algorithms for the AM and QM problem. Empirical studies on the various imbalanced datasets justify the effectiveness of the proposed approach.

1. Introduction

Classification problem with class imbalance where one class is rare compared to the other is a common yet important problem in supervised learning. It arises in many applications, ranging from medical diagnosis and text retrieval to credit risk prediction and fraud detection [1–4]. Due to its practical importance, it has been identified as one of the ten most challenging problems in data mining research [5]. For reasons of simplicity and with no loss in generality, only binary classification problems under class imbalance are considered in this paper. However, it is important to keep in mind that the class imbalance problem is pervasive in other areas as well such as multiclass classification and association rule mining.

It is well known that the usual binary learning algorithms are ill-suited in the imbalanced domains, because those classifiers will cause a bias towards the majority class and result in a lower sensitivity in detecting the minority class examples [6]. In the literature of solving class imbalance problems, a variety of approaches have been proposed, which

can be mainly categorized into two groups: the data-oriented methods and the algorithm-oriented methods.

The data-oriented methods use various sampling techniques to oversample instances in the minor class [6–8] or undersample those in the major class [9, 10], so that the resulting data is balanced. A typical example is the SMOTE approach [6] which increases the number of minor class instances by creating synthetic samples.

The second group algorithm-oriented methods aim at the extension and modification of existing classification algorithms so that they can be more effective in dealing with imbalanced data. For example, Liu et al. and Kang and Ramamohanarao have presented two different modified decision tree algorithms for improving the standard C4.5, such as CCPDT [11] and HeDEx [12], while Köknar-Tezel et al., Joachims et al., and Lipton et al. have proposed various approaches to improve traditional SVM's performance on the imbalanced settings [13–22].

Those two groups are both effective and it is difficult to say which one is better. However, since, in this paper, our

goal is to improve the existing statistical learning algorithm, in the following we are interested in algorithm-oriented method and propose a modified SVM approach by directly optimizing imbalance measure. It seems that our algorithm is similar to the algorithms in [15–22]; however, we design different objective functions and use different optimization techniques with theirs. More specifically, this paper makes the following contributions.

- (1) We adopt 1-slack structural SVM as the framework and define the loss functions oriented to AM and QM, which are rarely considered in the literature of optimizing imbalance metrics.
- (2) We show that the QM loss is a lower bound of the AM one, which means our QM classifier may be more accurate than the AM one.
- (3) For the inner computational challenge of the AM loss, we propose to decompose it nicely and apply the Hardy-Littlewood-Polya inequality to solve it in $O(n \log n)$ time, while, for the case of QM, such decomposition is impossible. We present an efficiently greedy method for solving this problem, which also requires $O(n \log n)$ time.
- (4) Empirical evaluations on the imbalanced datasets demonstrate that the proposed algorithms are not only significantly better than standard binary learning algorithm but also competitive to other existing imbalanced algorithms.

The remainder of the paper is organized as follows. In Section 2 the related work is presented. Section 3 discusses the details of our proposed algorithms and the empirical results on the benchmark datasets are reported in Section 4. Section 5 concludes the paper and discusses the future work.

2. Related Work

2.1. Problem Setup and Notations. As discussed in Introduction, in this paper, we only consider the binary classification problem. Given training dataset $D = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in R^d$ is the i th example and $y_i \in \{+1, -1\}$ is the corresponding class label. The binary classification problem is to construct a classifier function $f(x)$, which gives generalization performance. We assume that the classifier function is of the form $f(x) = w \cdot x$ and the decision function of the form $\bar{y} = \text{sign}(f(x))$ is used when finding the label of an unseen example. Note that we have not included the bias term in the classifier function for notational convenience. However, it can be incorporated in a straightforward way.

In machine learning area, a common way to find the linear parameter $w \in R^d$ is minimizing a regularized risk function:

$$w = \arg \min_w \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n l(x_i, y_i, w), \quad (1)$$

where $C > 0$ is a constant that controls the trade-off between training error minimization and margin maximization. l is a suitable loss function which measures the discrepancy between a true label y_i and a predicted value from using w . Different loss functions yield different learners. One of the most famous loss functions is the hinge loss in SVM, which has the form of $l(x_i, y_i, w) = \max(0, 1 - y_i w x_i)$.

2.2. Relevant Background. Standard SVM has been used to optimize an estimation of classification error on the training set and was shown to be a very powerful tool for classification problems when data is balanced. However, if the data is highly imbalanced, classification error is not always a good measure, and the standard SVM can be misleading. To solve this problem, a number of modified algorithms have been proposed. For example, Köknar-Tezel and Latecki [13] and Shao et al. [14] proposed approaches to improve SVM on imbalanced datasets, which they called GP and WLTSVM, respectively. But their works are both focused on improving sampling techniques (e.g., modifying SMOTE in GP) for SVM and do not solve the problem of training bias in the design of SVM learning algorithm per se. Recently, with the advances in learning to rank, direct optimization of the ranking measure technique has been extended to design SVM for imbalanced setting and a variety of algorithm-oriented methods have been proposed. Joachims [15] and Aioli [16] presented algorithms to optimize AUC for the imbalanced data, and the experimental results on the unbalanced sets proved their effectiveness. Along the lines of the above works, Paisitkriangkrai and Narasimhan et al. further gave algorithms by optimizing partial AUC and successfully applied their approaches to the real-world tasks [17–19]. Optimizing the F-measure is another popular method for imbalance learning. Joachims [15], Chinta et al. [20], Maratea et al. [21], and Lipton et al. [22] used different approximates to the F-measure and designed different classifiers. Numerical experiments on the benchmark datasets demonstrated their algorithms' effectiveness.

However, it is well known that, in evaluating imbalanced setting, there are many other performance measures besides AUC and F-measure, which include AM (arithmetic mean) [23] and QM (quadratic mean) [24]. The AM is the arithmetic mean of the true positive and true negative rates and can be defined as

$$\text{AM} = \frac{\text{TPR} + \text{TNR}}{2}. \quad (2)$$

The QM is a quadratic mean measure and is defined as

$$\text{QM} = 1 - \frac{((\text{FPR})^2 + (\text{FNR})^2)}{2}, \quad (3)$$

where

$$\begin{aligned} \text{TPR} &= P(\text{sign}(f(x)) = 1 \mid y = 1), \\ \text{TNR} &= P(\text{sign}(f(x)) = -1 \mid y = -1), \\ \text{FPR} &= P(\text{sign}(f(x)) = 1 \mid y = -1) = 1 - \text{TNR}, \\ \text{FNR} &= P(\text{sign}(f(x)) = -1 \mid y = 1) = 1 - \text{TPR}. \end{aligned} \quad (4)$$

Although AM and QM are popular in the imbalanced setting, surprisingly, little has been focused on designing the algorithms based on them. Until very recently, Menon provided a consistent algorithm, which aimed at directly optimizing AM measure [25]. This approach is effective, but it is only suitable for the AM measure; whether it can be extended to other measures such as QM is still unknown. In contrast to Menon's work, in this paper, we will present a general learning framework, whose loss function allows us to incorporate different imbalanced measure. We exploit it for optimizing AM and QM. In the following, we will discuss our approach in detail.

3. DOPMID: Direct Optimization of Performance Measure for Imbalanced Dataset

3.1. The Framework of DOPMID. We referred to the classifier we presented as DOPMID (Direct Optimization of Performance Measure for Imbalanced Dataset). The framework of DOPMID is based on structural SVM proposed by Joachims et al. [26]. Specifically, we use the 1-slack SVM formulation, presented in (OP1) (optimization problem 1), to learn a linear $w \in \mathbb{R}^d$. Note that the following approach can be extended to nonlinear function/non-Euclidean instance spaces by using kernels [27]:

$$\begin{aligned} \text{(OP1)} \quad \min_{w, \xi \geq 0} \quad & \frac{1}{2} w^T w + C\xi \\ \text{s.t.} \quad & \forall \bar{y} \in Y : w^T [\psi(x, y) - \psi(x, \bar{y})] \\ & \geq \Delta(y, \bar{y}) - \xi. \end{aligned} \quad (5)$$

For simplicity, in the paper, we assume that the training dataset $D = \{(x_i, y_i)\}_{i=1}^n$ has been ordered by the positive instances ahead of the negative ones, and we define $x = \{x_1^+, \dots, x_{\#pos}^+, x_1^-, \dots, x_{\#neg}^-\}$, $y = \{y_1^+, \dots, y_{\#pos}^+, y_1^-, \dots, y_{\#neg}^-\}$, where $\#pos$ is the number of the positive instances, $\#neg$ is the number of the negative instances, and $(\#pos) + (\#neg) = n$. $\bar{y}$ stands for any possible permutation of predicted list from using the parameter w . ψ represents a mapping function from input list to output list. $\Delta(y, \bar{y}) \rightarrow \mathbb{R}$ is a function used to measure the difference between the real

output y and the predicted output $\bar{y}$. This function must satisfy the following conditions:

$$\begin{aligned} (1) \quad & \text{for } \forall (y = \bar{y}), \quad \Delta(y, \bar{y}) = 0; \\ (2) \quad & \text{for } \forall (y \neq \bar{y}), \quad \Delta(y, \bar{y}) > 0. \end{aligned} \quad (6)$$

In contrast to the traditional SVM which has n slack ξ_i , there is only a single slack variable ξ in the (OP1) above. We refer to it as the "1-slack" SVM.

3.2. The Loss Functions Oriented to AM and QM. For the framework above, we need to further define the functions $\psi(x, y)$ and $\Delta(y, \bar{y})$, in order to determine the optimization target.

In this paper, we first define $\psi(x, y)$ as

$$\psi(x, y) = \sum_{i=1}^n y_i x_i. \quad (7)$$

Then we define $\Delta(y, \bar{y})$ oriented to AM and QM, respectively, as

$$\begin{aligned} \Delta_{AM}(y, \bar{y}) &= \frac{((\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)) / \#pos) + ((\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)) / \#neg)}{2}, \end{aligned} \quad (8)$$

$$\begin{aligned} \Delta_{QM}(y, \bar{y}) &= \frac{((\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)) / \#pos)^2 + ((\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)) / \#neg)^2}{2}. \end{aligned} \quad (9)$$

In equality (8) and (9), the function $1(\cdot)$ is an indicator function, which can be demonstrated as

$$1(a) = \begin{cases} 1 & \text{if } a \text{ is true} \\ 0 & \text{others.} \end{cases} \quad (10)$$

It is obvious that $\Delta_{AM}(y, \bar{y})$ and $\Delta_{QM}(y, \bar{y})$ defined in (8) and (9) satisfy the constraint conditions in (6). It has been proved that if the function Δ satisfies (6), the slack ξ is a convex upper bound on the training loss regularized by the norm of the weight vector [26].

In the following, we will show the fact that although ξ_{AM} , ξ_{QM} are both upper bound, ξ_{QM} is a lower bound than ξ_{AM} .

Lemma 1. ξ_{QM} defined by (9) is a lower bound than ξ_{AM} defined by (8).

Proof. Since the slack ξ is a convex upper bound on the training loss, we can rewrite (OP1) as

$$\begin{aligned} \min_{w, \xi \geq 0} \quad & \frac{1}{2} w^T w \\ & + C \left\{ \frac{\Delta(y, \bar{y}) - w^T [\psi(x, y) - \psi(x, \bar{y})]}{\xi} \right\}. \end{aligned} \quad (11)$$

Then we replace (8) and (9) with $\Delta(y, \bar{y})$ and get the AM bound and QM bound, respectively:

Step 1. **Input:** $x = \{x_1^+, \dots, x_{\#pos}^+, x_1^-, \dots, x_{\#neg}^-\}$, $y = \{y_1^+, \dots, y_{\#pos}^+, y_1^-, \dots, y_{\#neg}^-\}$, $C, \varepsilon > 0$
 Step 2. $W \leftarrow \emptyset$
 Step 3. **repeat**
 Step 4. $(w, \xi) \leftarrow \min_{w, \xi \geq 0} (1/2)w^T w + C\xi$
 s.t. $\forall \bar{y} \in W: w^T [\psi(x, y) - \psi(x, \bar{y})] \geq \Delta(y, \bar{y}) - \xi$
 Step 5. $\hat{y} \leftarrow \arg \max_{\bar{y} \in Y} \{\Delta(y, \bar{y}) + w^T \psi(x, \bar{y})\}$ (OP2)
 Step 6. $W \leftarrow W \cup \{\hat{y}\}$
 Step 7. **until** $\Delta(y, \hat{y}) - w^T [\psi(x, y) - \psi(x, \hat{y})] \leq \xi + \varepsilon$
 Step 8. **Output:** (w, ξ)

ALGORITHM 1: Cutting plane algorithm for solving (OP1) within tolerance ε .

$$\xi_{AM} = \left[\frac{\left(\left(\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i) \right) / \#pos \right) + \left(\left(\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i) \right) / \#neg \right)}{2} \right] - w^T [\psi(x, y) - \psi(x, \bar{y})], \quad (12)$$

$$\xi_{QM} = \left[\frac{\left(\left(\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i) \right) / \#pos \right)^2 + \left(\left(\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i) \right) / \#neg \right)^2}{2} \right] - w^T [\psi(x, y) - \psi(x, \bar{y})], \quad (13)$$

$$\begin{aligned} \xi_{QM} - \xi_{AM} = & \left[\frac{\left(\left(\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i) \right) / \#pos \right)^2 + \left(\left(\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i) \right) / \#neg \right)^2}{2} \right] \\ & - \left[\frac{\left(\left(\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i) \right) / \#pos \right) + \left(\left(\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i) \right) / \#neg \right)}{2} \right]. \end{aligned} \quad (14)$$

We can simplify (14) as

$$\begin{aligned} & \left(\frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos} \right)^2 + \left(\frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg} \right)^2 \\ & - \left(\frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos} \right) \\ & - \left(\frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg} \right). \end{aligned} \quad (15)$$

Since

$$0 \leq \left(\frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos} \right) \leq 1, \quad (16)$$

we obtain

$$\left(\frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos} \right)^2 \leq \frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos}. \quad (17)$$

Since

$$0 \leq \left(\frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg} \right) \leq 1, \quad (18)$$

we obtain

$$\left(\frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg} \right)^2 \leq \frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg}. \quad (19)$$

Combining inequality (17) and (19), we get

$$\begin{aligned} & \left(\frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos} \right)^2 + \left(\frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg} \right)^2 \\ & - \left(\frac{\sum_{i=1}^{\#pos} 1(\bar{y}_i \neq y_i)}{\#pos} \right) \\ & - \left(\frac{\sum_{i=\#pos+1}^n 1(\bar{y}_i \neq y_i)}{\#neg} \right) \leq 0 \end{aligned} \quad (20)$$

which means that $\xi_{QM} - \xi_{AM} \leq 0$ and proves the claim. $\square$

We can solve the (OP1) by substituting (7), (8), and (9) with (5), but unfortunately there is still a question: for each $\bar{y} \in Y$, inequality (5) has an exponential number of constraints. To solve this problem, we propose to use the cutting plane algorithm, which is based on the fact that, for any $\varepsilon > 0$, a small subset of the constraints is sufficient to find an ε -approximate solution to the problem. The detail of the cutting plane algorithm is shown in Algorithm 1.

The algorithm starts with no constraints and iteratively finds for each possible output $\hat{y}$ associated with the most violated constraint. If the corresponding constraint is violated by more than ε we introduce $\hat{y}$ into working set W and resolve (OP1) with the updated W . It can be shown that Algorithm 1's outer loop is guaranteed to halt within $O(1/\varepsilon)$ iterations for any desired precision ε [26].

Since the quadratic program in each iteration of this algorithm is of constant size, the only bottleneck in the algorithm is how to solve the (OP2), which is known as the problem of "finding the most violated constraint." In the

following, we will show how it can be performed efficiently for the AM loss and QM loss, respectively.

3.3. Efficient Algorithms for Finding the Most Violated Constraint. First of all, we rewrite (7) as

$$\psi(x, y) = \sum_{i=1}^n y_i x_i = \sum_{i=1}^{\#pos} y_i x_i + \sum_{i=\#pos+1}^n y_i x_i. \quad (21)$$

3.3.1. The Algorithm for AM Loss. Substituting (8) and (21) with (OP2), it follows that

$$\begin{aligned} & \arg \max_{\hat{y} \in Y} \{ \Delta(y, \hat{y}) + w^T \psi(x, \hat{y}) \} \\ &= \arg \max_{\hat{y} \in Y} \left\{ \frac{((\sum_{i=1}^{\#pos} 1(\hat{y}_i \neq y_i)) / \#pos) + ((\sum_{i=\#pos+1}^n 1(\hat{y}_i \neq y_i)) / \#neg)}{2} + \sum_{i=1}^{\#pos} w^T \hat{y}_i x_i + \sum_{i=\#pos+1}^n w^T \hat{y}_i x_i \right\} \\ &= \arg \max_{\hat{y} \in Y} \left\{ 1 - \frac{((\sum_{i=1}^{\#pos} 1(\hat{y}_i = y_i)) / \#pos) + ((\sum_{i=\#pos+1}^n 1(\hat{y}_i = y_i)) / \#neg)}{2} + \sum_{i=1}^{\#pos} w^T \hat{y}_i x_i + \sum_{i=\#pos+1}^n w^T \hat{y}_i x_i \right\} \quad (OP3) \\ &= \arg \max_{\hat{y} \in Y} \left\{ \sum_{i=1}^{\#pos} \left[w^T \hat{y}_i x_i - \frac{1(\hat{y}_i = y_i)}{2 \cdot \#pos} \right] + \sum_{i=\#pos+1}^n \left[w^T \hat{y}_i x_i - \frac{1(\hat{y}_i = y_i)}{2 \cdot \#neg} \right] \right\} \\ &= \arg \max_{\hat{y} \in Y} \left\{ \sum_{i=1}^{\#pos} \Gamma_i^{\#pos} + \sum_{i=\#pos+1}^n \Gamma_i^{\#neg} \right\}, \end{aligned}$$

where $\Gamma_i^{\#pos} = w^T \hat{y}_i x_i - 1(\hat{y}_i = y_i) / (2 \cdot \#pos)$ and $\Gamma_i^{\#neg} = w^T \hat{y}_i x_i - 1(\hat{y}_i = y_i) / (2 \cdot \#neg)$.

We can calculate $\Gamma_i^{\#pos}$ and $\Gamma_i^{\#neg}$ according to current weight w^T ; hence we apply the Hardy-Littlewood-Polya inequality and observe that (OP3) is maximized by sorting the terms $\Gamma_i^{\#pos}$, $\Gamma_i^{\#neg}$ in decreasing order. Note that this permutation is easily obtained by applying Quick Sort in $O(n \log n)$ time.

3.3.2. The Algorithm for QM Loss. Unlike for the AM loss, the (OP2) can be decomposed linearly in the instances. The (OP2) for the QM loss is quite different and needs a substantially extended algorithm, which we will describe in the following. First, we substitute (9) and (21) with (OP2). This gives

$$\begin{aligned} & \arg \max_{\hat{y} \in Y} \{ \Delta(y, \hat{y}) + w^T \psi(x, \hat{y}) \} \\ &= \arg \max_{\hat{y} \in Y} \left\{ \frac{((\sum_{i=1}^{\#pos} 1(\hat{y}_i \neq y_i)) / \#pos)^2 + ((\sum_{i=\#pos+1}^n 1(\hat{y}_i \neq y_i)) / \#neg)^2}{2} + \sum_{i=1}^{\#pos} w^T \hat{y}_i x_i + \sum_{i=\#pos+1}^n w^T \hat{y}_i x_i \right\}. \quad (OP4) \end{aligned}$$

From the above, we can see that the decomposition technology used in (OP3) is not suitable for (OP4), since $(\sum_{i=1}^{\#pos})^2$ and $(\sum_{i=\#pos+1}^n)^2$ can no longer be absorbed in $(\sum_{i=1}^{\#pos})$ and $(\sum_{i=\#pos+1}^n)$. To solve this problem, in the following, we will provide a more trick optimization for the (OP4). The

algorithm we proposed is based on the fact that, for each $\bar{y}_i$ in Y , there is only two possible values, which denote +1, -1. So in the following we will present an efficient algorithm, which can find $\arg \max_{\hat{y} \in Y}$ in $O(n \log n)$ time (see Algorithm 2).

The idea behind Algorithm 2 is the fact that the most violated constraint $\hat{y}$ must have the following form: k ($0 \leq$


```

Step 1. Input:  $x = \{x_1^+, \dots, x_{\#pos}^+, x_1^-, \dots, x_{\#neg}^-\}$ ,  $y = \{y_1^+, \dots, y_{\#pos}^+, y_1^-, \dots, y_{\#neg}^-\}$ , current weight  $w^T$ 
Step 2. Sort  $\{i : y_i^+, \dots, y_{\#pos}^+\} \rightarrow (i_1^P, \dots, i_{\#pos}^P)$  in descending order by current  $w^T x_i^+$ 
Step 3. Sort  $\{i : y_i^-, \dots, y_{\#neg}^-\} \rightarrow (i_1^N, \dots, i_{\#neg}^N)$  in descending order by current  $w^T x_i^-$ 
Step 4.  $\hat{y} = (y_{i_1^P}^P, \dots, y_{i_{\#pos}^P}^P, y_{i_1^N}^N, \dots, y_{i_{\#neg}^N}^N)$ 
Step 5.  $f_{\max} \leftarrow$  compute (OP4) with  $\hat{y}$ 
Step 6. for  $k = 0, \dots, \#pos$  do
Step 7.    $l \leftarrow \#pos - k$ 
Step 8.    $\widehat{y}_{i_1^P}^P = +1, \dots, \widehat{y}_{i_k^P}^P = +1$  and  $\widehat{y}_{i_{k+1}^P}^P = -1, \dots, \widehat{y}_{i_{\#pos}^P}^P = -1$ 
Step 9.   for  $t = 0, \dots, \#neg$  do
Step 10.     $v \leftarrow \#neg - t$ 
Step 11.     $\widehat{y}_{i_1^N}^N = +1, \dots, \widehat{y}_{i_v^N}^N = +1$  and  $\widehat{y}_{i_{v+1}^N}^N = -1, \dots, \widehat{y}_{i_{\#neg}^N}^N = -1$ 
Step 12.     $f_{\text{com}} \leftarrow$  compute (OP4) with  $\widehat{y}_{i_1^P}^P, \dots, \widehat{y}_{i_{\#pos}^P}^P, \widehat{y}_{i_1^N}^N, \dots, \widehat{y}_{i_{\#neg}^N}^N$ 
Step 13.    if  $f_{\text{com}} > f_{\max}$ 
Step 14.       $f_{\max} = f_{\text{com}}$ 
Step 15.       $\hat{y} = (\widehat{y}_{i_1^P}^P, \dots, \widehat{y}_{i_{\#pos}^P}^P, \widehat{y}_{i_1^N}^N, \dots, \widehat{y}_{i_{\#neg}^N}^N)$ 
Step 16.    end if
Step 17.  end for
Step 18. end for
Step 19. Output:  $(\hat{y})$ 

```

ALGORITHM 2: Algorithm for computing arg max with QP4 for QM loss.

$k \leq \#pos$) positive instances are labeled positive and other positive ones are labeled negative; t ($0 \leq t \leq \#neg$) negative instances are classified as negative and other negative ones are classified as positive. So we can get $\hat{y}$ by testing each $\hat{y}_i$ with $+1, -1$. Specifically, Algorithm 2 starts with a “perfect classification” (Step 2 to Step 5) and then uses the greedy algorithm to find the most violated constraint $\hat{y}$ with maximum value for (OP4) (Step 6 to Step 18).

Algorithm 2 is very efficient, and its running time can be split into two parts. The first part is the sort (Step 2, Step 3), which requires $O(n \log n)$. The second part is the following steps, which requires $O(\#pos \cdot \#neg)$ time. Though in the worst case this is $O(n^2)$, the number of the positive instances in the imbalanced datasets is very small, which means the running time for the second part is simply $O(n)$. So Algorithm 2 has complexity of $O(n \log n)$.

4. Experiments

4.1. Datasets and Baselines. The main goal of our experiments is to evaluate whether the classifiers we proposed can outperform the existing binary classifiers in the imbalanced setting. In particular, we select three datasets with varying degrees of class imbalance, taken from the libsvm dataset repository (downloaded from: <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>), which are named as satimage, wla, and vowel. The characteristics of those sets are summarized in Table 1.

The “#Examples” and “#Features” denote the number of examples and features, respectively. “Min(%)” represents the proportion of examples in the minority class.

On those imbalanced datasets, we compare our classifiers (DOPMID-AM, DOPMID-QM) against the following classifiers: SVM-Hinge [28], safe-level-SMOTE [7], CCPDT [11],

TABLE 1: Characteristics of the imbalanced datasets.

Dataset	#Examples	#Features	Min (%)
Satimage	6435	36	0.0973
wla	49749	300	0.0291
Vowel	990	10	0.0909

SVM-F1 [22], and AM-Consist [25]. Among them, SVM-Hinge is a traditional balanced binary classifier which seeks to minimize the hinge loss, and the latter four algorithms are all for the imbalanced setting. We choose those four as comparisons because we are interested in how our algorithms perform, when compared with other imbalanced algorithms. In the following experiments, unless otherwise noted, the parameters C of our algorithms are all chosen in the set $\{2^{-10}, \dots, 2^{10}\}$, and the error parameters ε of algorithms are set to 0.001.

4.2. Experimental Results. The experimental results are reported in terms of AM [23], QM [24], and F1 [15], which are all commonly adopted as the performance metrics for evaluating imbalanced learning classifiers. More specifically, we compare our proposed classifiers from the following two aspects.

4.2.1. Comparison with SVM-Hinge. We design this comparison in order to see whether the algorithms we proposed above can really be better-behaved than the standard SVM on the imbalanced datasets. The experimental results on the three sets are illustrated in Figure 1.

From Figure 1, we can see that as expected, on all the three datasets, DOPMID-AM and DOPMID-QM are significantly better than SVM-Hinge in terms of all the measures. Due

FIGURE 1: The performance comparison between DOPMID and SVM-Hinge.

to space limitation, we only report the statistics measured by AM as an example. When compared with SVM-Hinge, DOPMID-AM increases 32.21% on satimage set, 14.42% on w1a set, and 6.87% on the vowel set, while, for the DOPMID-QM, the improvement corresponds to 33.80%, 14.42%, and 13.87%, respectively. All the results prove the effectiveness of our method and once again indicate that we can obtain a more accurate unbalanced classifier by directly optimizing the imbalanced evaluation metrics.

Meanwhile, it can be seen from the figures that DOPMID-QM outperforms DOPMID-AM on most of the points. More specifically, on the experimental sets, DOPMID-QM is better than DOPMID-AM with 6 out of all the 9 points and is similar to DOPMID-AM with the other three points. These results, thus, suggest that DOPMID-QM can be more accurate than DOPMID-AM. This may be due to the fact that the loss function of DOPMID-QM is lower than the DOPMID-AM's, which can create more precise classifier. This observation is

also consistent with Chen's conjecture that it is possible to create more accurate model by defining tighter bound [29].

4.2.2. Comparison with Other Imbalanced Algorithms. In the second section, we are interested in how our algorithms perform, when compared with other imbalanced binary classifiers. More specifically, we compare our DOPMID with safe-level-SMOTE, CCPDT, and SVM-F1 and AM-Consist. We choose those four algorithms, because as discussed in Introduction, they represent two different methods for dealing with imbalanced problem. Safe-level-SMOTE uses over-sampling technique and belongs to the data-oriented method. We adopt it instead of SMOTE, since it can produce better accuracy than SMOTE by using different weight degrees on the synthetic examples [7]. The latter three algorithms all belong to algorithm-oriented methods. CCPDT is an efficient decision tree algorithm, which improved C4.5 on imbalanced datasets by using Fisher's exact test to prune branches of tree.

FIGURE 2: The performance comparison between DOPMID and other imbalanced binary algorithms.

SVM-F1 and AM-Consist are both SVM based approaches. SVM-F1 modified traditional SVM by optimizing F1 measure, while AM-Consist is a very recently proposed consistent classifier that aims at optimizing AM measure, which is the same as DOPMID-AM. We include it to make the effectiveness of our algorithms more convincing. It should be noted that, in the paper [25], the authors have proposed two consistent AM algorithms named as Plugin and Balanced, respectively. In our experiments, we select the Balanced as the AM-Consist, because a detailed analysis in their supplementary material shows that the Balanced performs better than the Plugin. Figure 2 depicts the behaviors of those algorithms on the satimage, wla, and vowel datasets.

As can be seen from Figure 2, the performance of algorithms in comparison varies from one dataset to another, and there is no one algorithm that can outperform other algorithms on all the datasets. For example, when measured by AM, safe-level-SMOTE performs best on vowel set, while it is the worst one on the other two sets. SVM-F1 performs well on satimage and wla in terms of F1; however, it yields poor performances on wla and vowel in terms of QM. Similarly, CCPDT achieves a better performance than AM-Consist on satimage dataset, but it is worse than AM-Consist on wla set. Different from those comparison algorithms, DOPMID-QM we proposed appears to perform more stably across all the datasets. Statistics show that when compared

with other imbalanced classifiers, DOPMID-QM performs best on 4 of 9 points and is the second best on 4 of 9 points. It is the third best one on the wla when measured by F1. All those demonstrate that even compared with others imbalanced binary classifiers, our DOPMID-QM approach is still effective and is suitable for the imbalanced settings.

Finally, we compare our DOPMID-AM with AM-Consist, because they both improve traditional SVM by optimizing AM measure. The results from Figure 2 are somewhat surprising where DOPMID-AM wins 5 of 9 points and fails in 4 points. More specifically, when compared with DOPMID-AM, AM-Consist performs better on wla set and yields poorer performances on the other two sets. One possible explanation is that AM-Consist is a consistent algorithm and may be more suitable for the set with large number of examples (such as wla dataset).

5. Conclusion

AM and QM are popular used performance measures in the imbalanced setting. In this paper, we have proposed a structural SVM based method, termed DOPMID for optimizing them. Specifically, we designed the objective functions oriented to AM and QM, respectively, and showed that the QM function is a tighter bound of the AM one. For the problem that the objective functions have exponential number of constraints, we introduce the cutting plane algorithm for outer optimization, which only needs $O(1/\epsilon)$ time. Then, for the inner computational challenge of the AM loss, we presented to decompose it nicely and applied the Hardy-Littlewood-Polya inequality to solve it, while, for the QM loss, we proposed an efficiently greedy algorithm, which still only required $O(n \log n)$ time. Our experiments on the imbalanced datasets showed that DOPMID is superior to the existing baseline techniques in terms of performance and stability. In future work, we hope to extend our approach to the multiclassification under class imbalance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Humanities and Social Sciences Project of Chinese Ministry of Education (Grant no. 13YJC870003), the Natural Science Foundation of China (Grant no. 61402002), and Key Program of Natural Science Project of Educational Commission of Anhui Province (Grant no. KJ2015A070), Youth Foundation of Anhui University (Grant no. KJQN1119), the Doctor Foundation of Anhui University (Grant no. 01001902), and the Foundation for the Key Teacher by Anhui University.

References

- [1] D. Vassiss, B. A. Kampouraki, and P. Belsis, "Using neural networks and SVMs for automatic medical diagnosis: a comprehensive review," in *4th International Conference on Integrated Information (IC-ININFO '14)*, vol. 1644 of *AIP Conference Proceedings*, pp. 32–36, AIP Publishing, Madrid, Spain, September 2014.
- [2] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.
- [3] A. Kshirsagar and L. Dole, "A review on data mining methods for identity crime detection," *International Journal of Electrical, Electronics and Computer Systems*, vol. 2, no. 1, pp. 312–318, 2014.
- [4] Q. Wu, Y. Ye, H. Zhang, M. K. Ng, and S. Ho, "ForesTexter: an efficient random forest algorithm for imbalanced text categorization," *Knowledge-Based Systems*, vol. 67, no. 9, pp. 105–116, 2014.
- [5] Q. Yang and X.-D. Wu, "10 Challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol. 5, no. 4, pp. 597–604, 2006.
- [6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [7] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: safe-level-synthetic minority over-sampling TEchnique for handling the class imbalanced problem," in *Advances in Knowledge Discovery and Data Mining: Proceedings of the 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27–30, 2009*, vol. 5476 of *Lecture Notes in Computer Science*, pp. 475–482, Springer, Berlin, Germany, 2009.
- [8] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE—majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, 2014.
- [9] M. Z. Zhu, C. Xu, and Y.-F. B. Wu, "IFME: information filtering by multiple examples with under-sampling in a digital library environment," in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '13)*, pp. 107–110, ACM, July 2013.
- [10] J. L. Hsu, P. C. Hung, and H. Y. Lin, "Applying under-sampling techniques and cost-sensitive learning methods on risk assessment of breast cancer," *Journal of Medical Systems*, vol. 39, no. 4, pp. 1–13, 2015.
- [11] W. Liu, S. Chawla, D. A. Cieslak, and N. V. Chawla, "A robust decision tree algorithms for imbalanced data sets," in *Proceedings of the 10th SIAM International Conference on Data Mining (SDM '10)*, pp. 766–777, Sydney, Australia, December 2010.
- [12] S. Kang and K. Ramamohanarao, "A robust classifier for imbalanced datasets," in *Advances in Knowledge Discovery and Data Mining*, pp. 212–223, Springer, 2014.
- [13] S. Köknar-Tezel and L. J. Latecki, "Improving SVM classification on imbalanced data sets in distance spaces," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM '09)*, pp. 259–267, December 2009.
- [14] Y.-H. Shao, W.-J. Chen, J.-J. Zhang, Z. Wang, and N.-Y. Deng, "An efficient weighted Lagrangian twin support vector machine for imbalanced data classification," *Pattern Recognition*, vol. 47, no. 9, pp. 3158–3167, 2014.

- [15] T. Joachims, "A Support Vector Method for multivariate performance measures," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 377–384, ACM, August 2005.
- [16] F. Aioli, "Convex AUC optimization for top-N recommendation with implicit feedback," in *Proceedings of the 8th ACM Conference on Recommender Systems*, pp. 293–296, ACM, Silicon Valley, Calif, USA, October 2014.
- [17] S. Paisitkriangkrai, C. Shen, and A. V. D. Hengel, "Efficient pedestrian detection by directly optimizing the partial area under the ROC curve," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 1057–1064, IEEE, Sydney, Australia, December 2013.
- [18] H. Narasimhan and S. Agarwal, "A structural SVM based approach for optimizing partial AUC," in *Proceedings of the 30th International Conference on Machine Learning (ICML '13)*, pp. 516–524, June 2013.
- [19] H. Narasimhan and S. Agarwal, "SVMpAUCtight: a new support vector method for optimizing partial AUC based on a tight convex upper bound," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '13)*, pp. 167–175, ACM, 2013.
- [20] P. M. Chinta, P. Balamurugan, and S. Shevade, "Optimizing F-measure with non-convex loss and sparse linear classifiers," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '13)*, pp. 1–8, IEEE, August 2013.
- [21] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted F-measure and kernel scaling for imbalanced data learning," *Information Sciences*, vol. 257, no. 2, pp. 331–341, 2014.
- [22] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," in *Machine Learning and Knowledge Discovery in Databases*, vol. 8725, pp. 225–239, Springer, Berlin, Germany, 2014.
- [23] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," *Communications in Computer and Information Science*, vol. 51, pp. 461–471, 2009.
- [24] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, and C. L. Giles, "Neural network classification and prior class probabilities," in *Neural Networks: Tricks of the Trade*, vol. 1524 of *Lecture Notes in Computer Science*, pp. 299–313, Springer, Berlin, Germany, 1998.
- [25] A. Menon, H. Narasimhan, and S. Agarwal, "On the statistical consistency of algorithms for binary classification under class imbalance," in *Proceedings of the 30th International Conference on Machine Learning (ICML '13)*, pp. 603–611, 2013.
- [26] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.
- [27] C. N. J. Yu and T. Joachims, "Training structural svms with kernels using sampled cuts," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 794–802, August 2008.
- [28] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, Kluwer Academic Publishers, 2002.
- [29] W. Chen, T. Y. Liu, and Y. Y. Lan, "Ranking measures and loss functions in learning to rank," in *Advances in Neural Information Processing Systems*, pp. 315–323, 2009.