Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 5873769, 11 pages
http://dx.doi.org/10.1155/2016/5873769

*Research Article*

# Efficient Optimization of *F*-Measure with Cost-Sensitive SVM

**Fan Cheng,[1,2] Yuan Zhou,[2] Jian Gao,[2] and Shuangqiu Zheng[2]**

[1]*Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, No. 3, Feixi Road, Hefei, Anhui 230039, China*
[2]*School of Computer, Anhui University, No. 3, Feixi Road, Hefei 230039, China*

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

Received 15 December 2015; Revised 2 March 2016; Accepted 22 March 2016

Academic Editor: Hiroyuki Mino

*F*-measure is one of the most commonly used performance metrics in classification, particularly when the classes are highly imbalanced. Direct optimization of this measure is often challenging, since no closed form solution exists. Current algorithms design the classifiers by using the approximations to the *F*-measure. These algorithms are not efficient and do not scale well to the large datasets. To fill the gap, in this paper, we propose a novel algorithm, which can efficiently optimize *F*-measure with cost-sensitive SVM. First of all, we present an explicit transformation from the optimization of *F*-measure to cost-sensitive SVM. Then we adopt bundle method to solve the inner optimization. For the problem where the existing bundle method may have the fluctuations in the primal objective during iterations, an additional line search procedure is involved, which can alleviate the fluctuations problem and make our algorithm more efficient. Empirical studies on the large-scale datasets demonstrate that our algorithm can provide significant speedups over current state-of-the-art *F*-measure based learners, while obtaining better (or comparable) precise solutions.

## 1. Introduction

SVM (Support Vector Machine) as a powerful classification tool is well known for its strong theoretical foundation and good generalization ability. In the binary setting, it is often evaluated by accuracy (the rate of correct classification). However, accuracy is not always a good measure and may be misleading, when the setting is imbalanced. In this situation, utility function such as *F*-measure provides a better way for the classifier evaluation, since it is a trade-off between precision and recall [1]. As a popular performance metric, *F*-measure has been widely used in diverse applications such as information retrieval [2, 3], biometrics [4, 5], and natural language processing [6, 7].

Owing to its importance, it has been well studied in machine learning area, and many works have been focused on designing the *F*-measure based classifiers. However, directly optimizing *F*-measure is often difficult as the resulting optimization problem is nonconvex, and no closed form solution exists. Therefore, various approximation algorithms have been proposed, which mainly fall into two paradigms [8]. The Empirical Utility Maximization (EUM) approach learns a classifier having optimal performance on training data [9–16], while the decision-theoretic (DT) approach learns a probabilistic model and then predicts labels with maximum expected *F*-measure [17–20]. Since, in this paper, our aim is to design an efficient classifier for maximizing *F*-measure, and DT approach possibly needs high computational complexity for the prediction step [8], in the following, we are focused on the Empirical Utility Maximization approach.

As the *F*-measure is a nonconvex metric, EUM approach often designs convex surrogates for optimizing *F*-measure and results in the development of two types of methods. The first type belongs to the "direct method," which directly defines different surrogate objective functions for maximizing *F*-measure [9–15]. One representative work is SVMperf, which adopts structural SVM as surrogate framework, and uses cutting plane algorithm to solve the inner optimization [11]. This algorithm has many virtues (such as good generalization performance and rapid convergence speed) and is viewed as the most important and successful algorithm in EUM approach. Suzuki et al., Cheng et al., and Chinta et al. extended the work of SVMperf and applied it into different areas [12–15]. The second type belongs to the "indirect method" and is recently proposed by Parambath et al. [16]. It is a novel method, which solves the problem by transforming it into a cost-sensitive classification.

Although those two methods are effective and are known to work fairly well in many different areas, there is still one disadvantage with those methods, where both of them are not very efficient, which may prohibit them from the large-scale applications. Moreover, for the novel "indirect method," its key contribution is the theoretical part, and the authors presented a theoretical analysis that the optimal $F$-measure classifier can be obtained by the reduction to cost-sensitive classification. But how to convert the procedure of maximizing $F$-measure into a cost-sensitive problem, that paper does not give an explicit solution.

To fill the gap, in this paper we focus on binary classification and propose a novel algorithm, which can efficiently optimize $F$-measure with cost-sensitive SVM. It seems that our algorithm belongs to the "indirect method"; however, it uses similar optimization technique like SVMperf, which means our algorithm can be viewed as the combination of the "direct method" and the "indirect method." More specifically, this paper makes the following contributions:

(1) Different from Parambath's work, which only gives a theoretical analysis, we present an explicit transformation from maximizing $F$-measure to cost-sensitive SVM.

(2) For the new cost-sensitive problem, we propose to solve it with bundle method, which is similar to the cutting plane algorithm used in SVMperf and has $O(1/\varepsilon)$ rate of convergence.

(3) Different from SVMperf, which is bothered with the fluctuations in primal objective, an additional line search procedure is introduced to the bundle method, which can avoid this undesirable effect and make our algorithm more efficient.

(4) Empirical evaluations on the large-scale imbalanced datasets demonstrate that when compared with currently existing $F$-measure based classifiers, the learner we proposed can greatly reduce the training time, while obtaining better (or comparable) accuracy of the model.

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 algorithm 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 the introduction, in this paper, we only consider the binary classification problem. Given a training dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in R^d$ is $i$th example and $y_i \in \{+1, -1\}$ is the corresponding class label. For simplicity, we assume that positive instances are ahead of negative ones, which means $i \in \underbrace{\{1, 2, 3, \ldots, \#\text{pos}\}}_{\#\text{pos}}$ are the indexes of positive instances and the rest $i \in \underbrace{\{\#\text{pos} + 1, \ldots, n\}}_{\#\text{neg}}$ are those of the negatives. #pos and #neg denote the number of positive instances and the negative ones, respectively. Binary classification problem is to construct a classifier function $f(x)$, which gives good generalization performance. In this paper, we assume the classifier is of the form $f(x) = w \cdot x$ and the decision function $\overline{y} = \text{sign}(f(x))$ is used to find 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 to minimize a regularized risk function:

$$w = \arg\min_{w \in R^d} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} l(x_i, y_i, w), \tag{1}$$

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 parameter $w$. Different loss functions yield different learners. One of the most famous loss functions is the hinge loss in SVM, which has the form of $l_{\text{hinge}}(x_i, y_i, w) = \max(0, 1 - y_i w x_i)$.

*2.2. Relevant Background.* When given a SVM classifier $f$, its performance can be evaluated by Table 1.

Table 1: Confusion matrix.

| | $y = 1$ | $y = -1$ |
|---|---|---|
| $\text{sign}(f(x)) = 1$ | TP | FP |
| $\text{sign}(f(x)) = -1$ | FN | TN |

TP, TN, FP, and FN in Table 1 denote true positive, true negative, false positive, and false negative, respectively, and

$$\begin{aligned} \text{TP} &= \left|\{y_i = 1 : \text{sign}(f(x_i)) = 1\}\right|, \\ \text{TN} &= \left|\{y_i = -1 : \text{sign}(f(x_i)) = -1\}\right|, \\ \text{FP} &= \#\text{neg} - \text{TN}, \\ \text{FN} &= \#\text{pos} - \text{TP}. \end{aligned} \tag{2}$$

By using the confusion matrix in Table 1, precision and recall can be expressed as

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad \mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{3}$$

In the imbalanced learning, people often use weighted harmonic mean of precision and recall, which is named $F$-measure to evaluate the performance of a classifier [1], and it can be formally defined as

$$F\text{-measure} = \frac{\left(\beta^2+1\right)\cdot \mathrm{PRE}\cdot \mathrm{REC}}{\beta^2\cdot \mathrm{PRE}+\mathrm{REC}}, \tag{4}$$

where $\beta \geq 0$. It is obvious that if $\beta = 0$, $F$-measure is the precision and if $\beta \to +\infty$, $F$-measure turns to the recall. In practice, the most widely used $F$-measure is $F1$, which means $\beta = 1$.

Because of its popular usage in the imbalanced classification, various approaches have been proposed for maximizing $F$-measure. One main paradigm is Empirical Utility Maximization, which learns a classifier having optimal $F$-measure on the training data. However, direct optimization of $F$-measure is difficult as the resulting optimization problem is nonconvex. Thus, approximation techniques are often used instead (since these algorithms directly design approximation objective functions oriented to $F$-measure, they are termed as “direct method” [8]). For example, Musicant et al., Liu et al., and Joachims et al. have designed different surrogate algorithms for optimizing the $F$-measure [9–11]. Among them, the work of Joachims et al. (which referred to SVMperf) is the most important, since not only their work provides a general framework for optimizing any imbalanced measure, but also their inner optimization technique is efficient. This algorithm makes use of a cutting plane solver along the lines of the structural SVM and has $O(1/\varepsilon)$ rate of convergence for any desired precision $\varepsilon$. Based on this work, Suzuki et al. and Cheng et al. applied SVMperf to the CRF and topical classification [12, 13], while Chinta et al. and Dembczynski et al. further extended it to the sparse learning and multilabel learning [14, 15]. Recently, an “indirect method” for optimizing $F$-measure which used cost-sensitive technology has been proposed by Parambath et al. [16]. The authors took the advantage of the pseudolinearity of $F$-measure and presented a theoretical analysis that the optimal classifier for $F$-measure can be obtained by solving a cost-sensitive problem. Both “direct method” and “indirect method” are effective and are suitable for many various applications. However, those two methods have a common limitation that they are not very efficient, which may prohibit them from being used in the large-scale datasets. For the “direct method,” we take the SVMperf as an example, which is one of the most efficient algorithms in EUM. Although it has rapid convergence speed, its inner optimization can only guarantee the dual objective function increases monotonically and does not guarantee the primal objective decreases monotonically [21]. These fluctuations in primal objectives may slow down the practical convergence speed and make the SVMperf inefficient. Similar problem occurs on the “indirect method,” because this method should be implemented with other existing SVMs, which may also have these undesirable fluctuations during the iterations. Furthermore, for the recently proposed “indirect method,” the main contribution of it is the theoretical part, and the authors only present a theoretical analysis that maximizing $F$-measure can be reduced to a cost-sensitive classification. However, they do not give an explicit transformation from the optimization of $F$-measure to cost-sensitive SVM.

So in the following, by giving an explicit transformation, we will present a novel cost-sensitive SVM based algorithm that can maximize $F$-measure. The algorithm uses the bundle method as the inner optimizer and avoids the fluctuations in primal objective by adding a line search procedure, which means our algorithm can be more efficient than the existing algorithms such as [8, 11, 16].

## 3. Efficient Algorithm for Optimizing $F$-Measure with Cost-Sensitive SVM

*3.1. From Maximizing F-Measure to the Cost-Sensitive Classification.* Based on the definition of formula (4), $F$-measure can be further expressed as

$$\begin{aligned} F\text{-measure} &= \frac{\left(\beta^2+1\right)\cdot \mathrm{PRE}\cdot \mathrm{REC}}{\beta^2\cdot \mathrm{PRE}+\mathrm{REC}} = \frac{\left(\beta^2+1\right)\cdot \left(\mathrm{TP}/\left(\mathrm{TP}+\mathrm{FP}\right)\right)\cdot\left(\mathrm{TP}/\left(\mathrm{TP}+\mathrm{FN}\right)\right)}{\beta^2\cdot\left(\mathrm{TP}/\left(\mathrm{TP}+\mathrm{FP}\right)\right)+\mathrm{TP}/\left(\mathrm{TP}+\mathrm{FN}\right)} \\ &= \frac{\left(\beta^2+1\right)\cdot \mathrm{TP}\cdot\mathrm{TP}}{\beta^2\cdot\mathrm{TP}\cdot\left(\mathrm{TP}+\mathrm{FN}\right)+\mathrm{TP}\cdot\left(\mathrm{TP}+\mathrm{FP}\right)} = \frac{\left(\beta^2+1\right)\cdot\mathrm{TP}}{\beta^2\cdot\left(\mathrm{TP}+\mathrm{FN}\right)+\mathrm{TP}+\mathrm{FP}} \\ &= \frac{\left(\beta^2+1\right)\cdot\left(\#\mathrm{pos}-\mathrm{FN}\right)}{\beta^2\cdot\left(\#\mathrm{pos}-\mathrm{FN}+\mathrm{FN}\right)+\left(\#\mathrm{pos}-\mathrm{FN}\right)+\mathrm{FP}} = \frac{\left(\#\mathrm{pos}-\mathrm{FN}\right)}{\beta^2/\left(\beta^2+1\right)\cdot\left(\#\mathrm{pos}\right)+\left(\left(\#\mathrm{pos}-\mathrm{FN}\right)+\mathrm{FP}\right)/\left(\beta^2+1\right)} \\ &= \frac{1}{\beta^2/\left(\beta^2+1\right)\cdot\left(\#\mathrm{pos}\right)/\left(\#\mathrm{pos}-\mathrm{FN}\right)+\left(\left(\#\mathrm{pos}-\mathrm{FN}\right)+\mathrm{FP}\right)/\left(\left(\beta^2+1\right)\cdot\left(\#\mathrm{pos}-\mathrm{FN}\right)\right)}. \end{aligned} \tag{5}$$

By assuming #pos ≠ FN, we can find that maximizing $F$-measure is equivalent to minimizing the following problem:

$$\begin{gathered}\min_{w\in R^d}\left[\frac{\beta^2}{(\beta^2+1)}\cdot\frac{(\#\text{pos})}{(\#\text{pos}-\text{FN})}+\frac{1+\text{FP}/(\#\text{pos}-\text{FN})}{(\beta^2+1)}\right]\Longleftrightarrow\\ \min_{w\in R^d}\left[\frac{\beta^2}{(\beta^2+1)}\cdot\frac{(\#\text{pos})}{(\#\text{pos}-\text{FN})}+\frac{\text{FP}}{(\beta^2+1)(\#\text{pos}-\text{FN})}\right]\Longleftrightarrow\\ \min_{w\in R^d}\left[\frac{\beta^2(\#\text{pos})+\text{FP}}{(\beta^2+1)\cdot(\#\text{pos}-\text{FN})}\right]\Longleftrightarrow\\ \min_{w\in R^d}\left\{\beta^2(\#\text{pos})+\text{FP}-\vartheta\left[\left(\beta^2+1\right)\cdot(\#\text{pos}-\text{FN})\right]\right\},\end{gathered}\tag{6}$$

where $\vartheta$ is a positive constant. Since $\beta$ and #pos are both constants, formula (6) can be simplified as

$$\begin{gathered}\min_{w\in R^d}\left\{\text{FP}+\vartheta\left[\left(\beta^2+1\right)\cdot\text{FN}\right]\right\}\\ \Longleftrightarrow\min_{w\in R^d}\left[\text{FP}+\widehat{\vartheta}\cdot\text{FN}\right],\end{gathered}\tag{7}$$

where $\widehat{\vartheta}=\vartheta(\beta^2+1)$ is a positive constant. Based on the definitions of FP and FN, formula (7) can be rewritten as

$$\begin{gathered}\min_{w\in R^d}\left[\sum_{i=\#\text{pos}+1}^{n}I\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)+\widehat{\vartheta}\cdot\sum_{i=1}^{\#\text{pos}}I\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)\right]\Longleftrightarrow\\ \min_{w\in R^d}\left[C_p\cdot\sum_{i=1}^{\#\text{pos}}I\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)+C_n\cdot\sum_{i=\#\text{pos}+1}^{n}I\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)\right],\end{gathered}\tag{8}$$

where $C_p+C_n=1$ and $0\le C_p, C_n\le 1$ are the misclassification cost parameters for positive and negative classes, respectively. It is obvious that formula (8) is a cost-sensitive problem, and the lower the total cost, the better the classification performance.

### 3.2. Efficient Algorithm for the Cost-Sensitive SVM.

Based on formula (8), we can transform the maximization of $F$-measure problem to a cost-sensitive SVM, which is demonstrated as

$$w=\arg\min_{w\in R^d}\frac{1}{2}\|w\|^2+C\cdot\left[C_p\cdot\sum_{i=1}^{\#\text{pos}}l_{\text{hinge}}\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)+C_n\cdot\sum_{i=\#\text{pos}+1}^{n}l_{\text{hinge}}\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)\right].\tag{9}$$

Formula (9) is equivalent to

*OP1*:

$$w=\arg\min_{w\in R^d}\overbrace{\frac{\delta}{2}\|w\|^2+\underbrace{\left[C_p\cdot\sum_{i=1}^{\#\text{pos}}l_{\text{hinge}}\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)+C_n\cdot\sum_{i=\#\text{pos}+1}^{n}l_{\text{hinge}}\left(y_i\neq\text{sign}\left(f\left(x_i\right)\right)\right)\right]}_{R^{\text{cs}}_{\text{emp}}(w)}}^{J(w)},\tag{10}$$

where $\delta=1/C$ is a constant that controls the trade-off between training error minimization and margin maximization. For the OP1 above, we regard it as a regularized risk minimization problem and adopt the bundle method to solve it. Bundle method uses subgradients of the empirical risk function to approximate its piecewise linear lower bound, which is similar to the cutting plane algorithm (CPA) used in SVMperf. In contrast to CPA, the linear lower bound of bundle method is augmented with a stabilization term, which can guarantee a good quality solution [22]. By taking the first-order Taylor approximation to the empirical risk function, the lower bounder is tightened iteratively until the difference gap between the approximated lower bound and the real risk function is smaller than a predefined threshold $\varepsilon$. The whole algorithm can be described as shown in Algorithm 1.

It has been proved that the bundle method in Algorithm 1 had $O(1/\varepsilon)$ rate of convergence for any desired precision $\varepsilon$ [22]. Although the bundle method is effective, it also has the fluctuations in primal objective, which also occurred on the CPA of SVMperf. More specifically, when solving the subproblem of step (8), it always selects a new cutting plane such that the dual objective monotonically increases. However, this selection does not guarantee that the primal problem monotonically decreases, which means that the primal objective values can heavily fluctuate between iterations, and it may slow down the practical convergence speed of the algorithm.

To solve this problem and speed up the convergence, in the following, we will present an additional line search algorithm for step (8), which can guarantee that the primal objective function monotonically decreases and make our algorithm more efficient.

```
(1) **Input**: convergence threshold $\varepsilon > 0$;
(2) **Initialize**: weight vector $w_0$ and iteration index $t = 0$;
(3) **repeat**
(4) $t = t + 1$;
(5) Compute $\alpha_t = \partial_w R^{\mathrm{cs}}_{\mathrm{emp}}(w_{t-1})$;
(6) Compute bias $b_t = R^{\mathrm{cs}}_{\mathrm{emp}}(w_{t-1}) - w_{t-1}\alpha_t$;
(7) Update the lower bound $R_t^{\mathrm{BM}}(w) = \max_{1\le i\le t}\{w\cdot\alpha_i + b_i\}$;
(8) $w_t = \arg\min_{w\in R^d}\underbrace{((\delta/2)\|w\|^2 + R_t^{\mathrm{BM}}(w))}_{J_t(w)}$;
(9) Compute current gap $\varepsilon_t = \min_{0\le i\le t}(J(w_i) - J_t(w_t))$;
(10) **until** $\varepsilon_t \le \varepsilon$;
(11) **Output**: $w_t$ as the $w^*$ of *OP1*.
```

Algorithm 1: Bundle method for solving OP1 within tolerance $\varepsilon$.

*3.3. Efficient Line Search Algorithm.* First of all, we introduce an intermediate variable $w_t^c$ maintaining the best-so-best solution during the first $t$ iterations, which means $J(w_1^c),\dots,J(w_t^c)$ is a monotonically decrease sequence.

Secondly, the new $w_t^c$ is found by searching a line along the previous $w_{t-1}^c$ and the origin solution $w_t$, which gives the following:

*OP2*:

$$w_t^c = \min_{l\ge 0} J\left(w_{t-1}^c(1-l) + w_t l\right). \tag{11}$$

Finally, the new cutting plane is computed to approximate the primal objective $J$ at a point $w_t^d$, which lies in a vicinity of the $w_t^c$. More specifically, the variable $w_t^d$ is obtained by

$$w_t^d = w_t^c(1-\lambda) + w_t\lambda, \tag{12}$$

where $\lambda \in (0,1]$ is a predefined parameter. With the point $w_t^d$, the new cutting plane is given by $\alpha_{t+1} \in \partial_w R^{\mathrm{cs}}_{\mathrm{emp}}(w_t^d)$, and $b_{t+1} = R^{\mathrm{cs}}_{\mathrm{emp}}(w_t^d) - w_t^d\alpha_{t+1}$.

Similar to step (9) of Algorithm 1, a nature stopping condition for our improved algorithm is

$$J(w_t^c) - J_t(w_t) \le \varepsilon. \tag{13}$$

With those changes to bundle method, we can generate a monotonically decrease sequence of primal objective and achieve faster convergence. However, in practice, there is still one problem with our improved algorithm, which is how to compute the OP2 efficiently. So in the following, we will give an efficient algorithm for solving this problem, which only needs $O(n\log n)$ time.

Firstly, combining formulas (10) and (11), we can obtain

$$\begin{aligned}J\left(w_{t-1}^c(1-l)+w_t l\right) &= \frac{\delta}{2}\left\|w_{t-1}^c(1-l)+w_t l\right\|^2 + \Bigg[C_p \\ &\cdot \sum_{i=1}^{\#\mathrm{pos}} l_{\mathrm{hinge}}\left(w_{t-1}^c(1-l)+w_t l\right) + C_n \\ &\cdot \sum_{i=\#\mathrm{pos}+1}^{n} l_{\mathrm{hinge}}\left(w_{t-1}^c(1-l)+w_t l\right)\Bigg].\end{aligned} \tag{14}$$

We abbreviate $j(l) \equiv J(w_{t-1}^c(1-l) + w_t l)$ and get

$$\begin{aligned}j(l) &= \frac{\delta}{2}\left\|w_{t-1}^c + \left(w_t - w_{t-1}^c\right)l\right\|^2 \\ &\quad + \sum_{i=1}^{\#\mathrm{pos}} C_p \cdot l_{\mathrm{hinge}}\left(w_{t-1}^c(1-l)+w_t l\right) \\ &\quad + \sum_{i=\#\mathrm{pos}+1}^{n} C_n \cdot l_{\mathrm{hinge}}\left(w_{t-1}^c(1-l)+w_t l\right) \\ &= \frac{1}{2}\underbrace{\delta\left\|\left(w_t - w_{t-1}^c\right)\right\|^2}_{A_0} l^2 + \underbrace{\delta\left\langle w_{t-1}^c, w_t - w_{t-1}^c\right\rangle}_{B_0} l \\ &\quad + \underbrace{\frac{\delta}{2}\left\|w_{t-1}^c\right\|^2}_{Z_0} \\ &\quad + \underbrace{\sum_{i=1}^{\#\mathrm{pos}} C_p \cdot \max\left\{0, 1 - y_i x_i\left(w_{t-1}^c(1-l)+w_t l\right)\right\}}_{g_{\mathrm{pos}}(l)} \\ &\quad + \underbrace{\sum_{i=\#\mathrm{pos}+1}^{n} C_n \cdot \max\left\{0, 1 - y_i x_i\left(w_{t-1}^c(1-l)+w_t l\right)\right\}.}_{g_{\mathrm{neg}}(l)}\end{aligned} \tag{15}$$

That is,

$$\begin{aligned}j(l) &= \frac{1}{2}l^2 A_0 + lB_0 + Z_0 + g_{\mathrm{pos}}(l) + g_{\mathrm{neg}}(l) \\ &= \frac{1}{2}l^2 A_0 + lB_0 + Z_0 + \sum_{i=1}^{\#\mathrm{pos}} g_{\mathrm{pos}}^i(l) \\ &\quad + \sum_{i=\#\mathrm{pos}+1}^{n} g_{\mathrm{neg}}^i(l),\end{aligned} \tag{16}$$

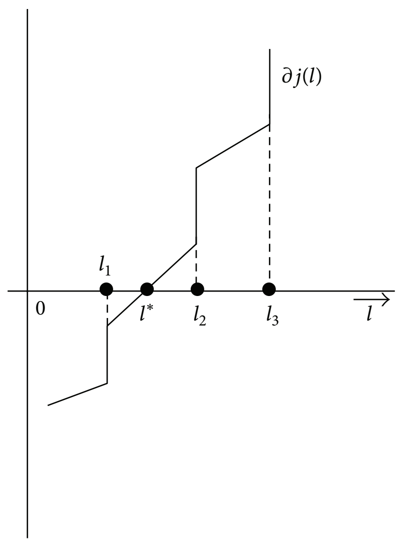
where

$$\begin{aligned}&g_{\mathrm{pos}}^i(l) \\ &\quad = \max\left\{0, C_p\left(1 - y_i x_i w_{t-1}^c\right) + C_p y_i x_i\left(w_{t-1}^c - w_t\right)l\right\} \\ &g_{\mathrm{neg}}^i(l) \\ &\quad = \max\left\{0, C_n\left(1 - y_i x_i w_{t-1}^c\right) + C_n y_i x_i\left(w_{t-1}^c - w_t\right)l\right\}.\end{aligned} \tag{17}$$

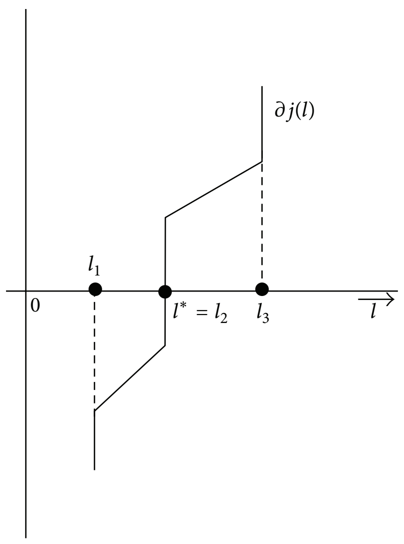
The OP2 is equivalent to solving $l^* = \arg\min_{l\ge 0} j(l)$. Since $j(l)$ is a convex function, its minimum is attained at the point $l^*$, where the subdifferential $\partial j(l)$ contains zero, which means $0 \in \partial j(l)$ hold. The subdifferential $\partial j(l)$ can be expressed as

$$\partial j(l) = A_0 l + B_0 + \sum_{i=1}^{\#\mathrm{pos}} \partial g_{\mathrm{pos}}^i(l) + \sum_{i=\#\mathrm{pos}+1}^{n} \partial g_{\mathrm{neg}}^i(l), \tag{18}$$

where



(a) *Case 1.* $l^*$ lies in the slash area



(b) *Case 2.* $l^*$ lies in the vertical area

Figure 1: The step function of $\partial j(l)$.

$$\partial g_{\text{pos}}^i(l) = \begin{cases} 0 & \text{if } C_p(1 - y_i x_i w_{t-1}^c) + C_p y_i x_i (w_{t-1}^c - w_t) l < 0 \\ C_p y_i x_i (w_{t-1}^c - w_t) & \text{if } C_p(1 - y_i x_i w_{t-1}^c) + C_p y_i x_i (w_{t-1}^c - w_t) l > 0 \\ [0, C_p y_i x_i (w_{t-1}^c - w_t)] & \text{if } C_p(1 - y_i x_i w_{t-1}^c) + C_p y_i x_i (w_{t-1}^c - w_t) l = 0 \end{cases}$$
$$\partial g_{\text{neg}}^i(l) = \begin{cases} 0 & \text{if } C_n(1 - y_i x_i w_{t-1}^c) + C_n y_i x_i (w_{t-1}^c - w_t) l < 0 \\ C_n y_i x_i (w_{t-1}^c - w_t) & \text{if } C_n(1 - y_i x_i w_{t-1}^c) + C_n y_i x_i (w_{t-1}^c - w_t) l > 0 \\ [0, C_n y_i x_i (w_{t-1}^c - w_t)] & \text{if } C_n(1 - y_i x_i w_{t-1}^c) + C_n y_i x_i (w_{t-1}^c - w_t) l = 0. \end{cases} \quad (19)$$

For formula (18), the first two terms constitute an ascending linear function $A_0 l + B_0$, since $A_0 = \delta \|w_t - w_{t-1}^c\|^2 \geq 0$. Note $A_0 = 0$ means that $w_t = w_{t-1}^c$, which indicates that our algorithm has converged to the optimum $w^*$. The latter two terms $\partial g_{\text{pos}}^i(l)$ and $\partial g_{\text{neg}}^i(l)$ are either constants or step functions by the definitions of formula (19). Hence, $\partial j(l)$ is a monotonically increasing function, which can be depicted as in Figure 1.

From Figure 1, we can begin with $l = 0$ to find the best solution of $l^* = \arg\min_{l \geq 0} j(l)$:

$$\partial j(0) = B_0 + \sum_{i=1}^{\#\text{pos}} \partial g_{\text{pos}}^i(0) + \sum_{i=\#\text{pos}+1}^{n} \partial g_{\text{neg}}^i(0), \quad (20)$$

where

$$\partial g_{\text{pos}}^i(0) = \begin{cases} 0 & \text{if } C_p(1 - y_i x_i w_{t-1}^c) < 0 \\ C_p y_i x_i (w_{t-1}^c - w_t) & \text{if } C_p(1 - y_i x_i w_{t-1}^c) > 0 \\ [0, C_p y_i x_i (w_{t-1}^c - w_t)] & \text{else } C_p(1 - y_i x_i w_{t-1}^c) = 0, \end{cases}$$

$$\partial g_{\text{neg}}^i(0) = \begin{cases} 0 & \text{if } C_n(1 - y_i x_i w_{t-1}^c) < 0 \\ C_n y_i x_i (w_{t-1}^c - w_t) & \text{if } C_n(1 - y_i x_i w_{t-1}^c) > 0 \\ [0, C_n y_i x_i (w_{t-1}^c - w_t)] & \text{else } C_n(1 - y_i x_i w_{t-1}^c) = 0. \end{cases} \quad (21)$$

Based on equalities (20) and (21), we can find that if $\max(\partial j(0)) \geq 0$, the minimum $j(l)$ is attained at the point equal to 0, which means $l^* = 0$. While if $\max(\partial j(0)) < 0$, the optimum $l^*$ is obtained by finding an intersection between $\partial j(l)$ and the $x$-axis (as Figure 1 shows). This can be done efficiently by sorting every step points. The whole algorithm is described as shown in Algorithm 2.

For Algorithm 2, the theorem below guarantees that it only requires $O(n \log n)$ time.

**Theorem 1.** *The total running time of Algorithm 2 is* $O(n \log n)$.

```
(1) **Input**: $w_t, w^c_{t-1}, A_0, B_0, Z_0, C_p, C_n, x_i, y_i$
(2) Compute $\max(\partial j(0))$ with formula (19) and (20)
(3) **if** $\max(\partial j(0)) \geq 0$ **then**
(4)     $l^* = 0$
(5) **else**
(6)     $L \leftarrow \{l_i \mid l_i = (1 - y_i x_i w^c_{t-1})/y_i\ x_i(w_t - w^c_{t-1}) > 0\}$ // find the step point sets with $l_i > 0$
(7)     Sort $L$ in ascending order $\rightarrow \{l_1, \ldots, l_K\}$ where $K = |L|$
(8)     $l_0 = 0$ // begin with zero point
(9)     **for** $k = 1, \ldots, K$ **do**
(10)        $\partial j(l_k) = \partial j(l_{k-1}) + A_0(l_k - l_{k-1})$
(11)        **if** $\partial j(l_k) > 0$ **then**
(12)            $l^* = (-\sum_{i=0}^{k-1} B_i)/A_0$ // *Case 1* $l^*$ lies in the slash area
                where $B_i = \begin{cases} B_0 & \text{if } i = 0 \\ C_p y_i x_i (w^c_{t-1} - w_t) & \text{if } i \neq 0,\ y_i = +1 \\ C_n y_i x_i (w^c_{t-1} - w_t) & \text{else } i \neq 0,\ y_i = -1 \end{cases}$
(13)        **else**
(14)            $\partial j(l_k) = \partial j(l_k) + B_k$
(15)            **if** $\partial j(l_k) \geq 0$ **then**
(16)                $l^* = l_k$ // *Case 2* $l^*$ lies in the vertical area
(17)            **end if**
(18)        **end if**
(19)    **end for**
(20) **end if**
(21) **Output**: $l^*$.
```

Algorithm 2: Efficient line search for solving OP2.

*Proof.* From the pseudo code of Algorithm 2, we can find that the time of step (1) to step (4) is $O(1)$, and step (6) needs at most $O(n)$ time. Step (7) is sorting, which can be implemented in $O(K \log K)$ time. Step (9) to step (19) take $O(K)$ time, and other steps all require $O(1)$ time. Therefore, Algorithm 2 has the complexity of $O(n + K \log K)$. Since in practice $K \leq n$, which means that even in the worst case the total running time of Algorithm 2 is $O(n \log n)$. □

## 4. Experiments

In this section, we will compare our classifier with other existing learners for maximizing $F$-measure and give details about results on the benchmark datasets.

*4.1. Baselines and Datasets.* We evaluate the performance of our algorithm (termed as BM-ls-CS) with SVMperf [11], SVM-CS [16], BM-nls-CS, and LR-ML$^E$ [8], which are all $F$-measure based learners. The first three baselines follow EUM approach, while the last one falls into DT approach. More specifically, the first one, SVMperf, uses direct method and is the most popular imbalanced classifier for EUM approach. It adopts the structured SVM to maximize $F$-measure and applies the cutting plane algorithm for inner optimization, which is similar to ours. The second one, SVM-CS, is a recently proposed classifier, and as mentioned before, it belongs to indirect method. Same as our BM-ls-CS, it is a cost-based algorithm. The main difference between SVM-CS and ours is the inner solver. The third comparison algorithm BM-nls-CS uses bundle method without line search as the optimizer. We include it in evaluations to see whether the line search technology we proposed can improve the speed of convergence. In addition, we also compare our algorithm with a decision-theoretic method named LR-ML$^E$, which is recently proposed by Ye et al. We select it as the fourth baseline, since it is an efficient algorithm and only needs $O(n^2)$ time for computing the optimal predictions. Finally, it should be noted that, in our experiments, we do not implement the $F$-measure based learners with "approximative solver" (such as SGD). Although these learners may have a low per-iteration cost and low total training time, their approximations to the optimal solution are crude and often fail to achieve a precise solution.

All the comparison algorithms adopt the same experimental setup and are carried out on a Linux machine with 3.4 GHz Intel Core and 8 GB of RAM. The penalty parameter $C$ for SVMperf is selected from set $\{2^{-4}, \ldots, 2^8\}$ by cross-validation, and the corresponding parameters for SVM-CS and LR-ML$^E$ are determined from set $\{n \cdot 2^{-4}, \ldots, n \cdot 2^8\}$ (the parameters $C$ in SVMperf and $C$ in SVM-CS and LR-ML$^E$ satisfy the following relation: $C_{\text{perf}} = n \cdot C_{\text{cs}} = n \cdot C_{\text{LR-ML}^E}$), while for BM-ls-CS and BM-nls-CS, the regularization parameter $\delta \in \{2^{-8}/100, \ldots, 2^4/100\}$ (the parameters $C$ in SVMperf and $\delta$ in bundle method satisfy the following relation: $C = 1/100\delta$) and the parameter $\lambda$ in formula (12) are selected from $\{0.1, 0.2, \ldots, 1\}$. For all the cost-sensitive algorithms (BM-ls-CS, SVM-CS, and BM-nls-CS), we set $C_p = (1 - t/2)$ and $C_n = t/2$. The proper $t$ is chosen from $t \in \{0.1, 0.2, \ldots, 0.9\}$, which is suggested by Proposition 6 in Parambath's paper.

Table 2: Characteristics of the experimental datasets.

| Dataset | #Examples | #Features | Min (%) |
|---|---|---|---|
| a3a | 32561 | 123 | 0.2396 |
| acoustic | 98528 | 50 | 0.2324 |
| ijcnn1 | 141691 | 22 | 0.0971 |
| letter | 20000 | 16 | 0.0396 |
| news20 | 19928 | 62061 | 0.0500 |
| satimage | 6435 | 36 | 0.2417 |

The approximation gap $\varepsilon$ is set as $\varepsilon = 10^{-3}$ for each EUM algorithm.

With the algorithms above (SVMperf, SVM-CS, BM-nls-CS,LR-ML$^E$, and BM-ls-CS), we perform experiments on six datasets, which are a3a, acoustic, ijcnn1, letter, news20, and satimage. We choose them as the experimental sets, since they are all imbalanced datasets with large sample sizes. These datasets can be downloaded from LIBSVM website (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/), and their characteristics are summarized in Table 2.

In Table 2, the "#Examples" and "#Features" denote the number of examples and features, respectively. "Min" represents the proportion of examples in the minority class. For each dataset, we use the same split as in LIBSVM repository and report the results from the following two aspects.

## *4.2. Experimental Results*

*4.2.1. The Performance Behaviors of Different Learners (F1 Value and Running Time).* Since the main goal of this paper is to produce an efficient $F$-measure learner, in the first part of experiments, we are concerned about the performance behaviors of different algorithms, and the comparison results are depicted in Table 3.

Note that, in Table 3, there are two values in each blank, the top one denotes the objective value measured by $F1$, and the bottom one is the running time in seconds. Higher value and lower time are better.

From Table 3, we can find that when measured by $F1$, the performances of EUM algorithms (SVMperf, SVM-CS, BM-nls-CS, and BM-ls-CS) and DT algorithm (LR-ML$^E$) vary from one dataset to another, and there is no one algorithm that can outperform other algorithms on all the datasets. For example, on news20 set, EUM algorithms are almost better than DT algorithm, while, on satimage set, DT algorithm is superior to EUM algorithms. This is coherent with the result of Ye et al., according to which both two approaches are effective, and it is difficult to say which one is better on the large datasets [8]. Meanwhile, statistics show that, for the EUM approach, three cost-sensitive algorithms (SVM-CS, BM-nls-CS, and BM-ls-CS) are better than (or comparable to) SVMperf, which once again indicate that we can produce a good $F$-measure based classifier by transforming it into a cost-sensitive problem.

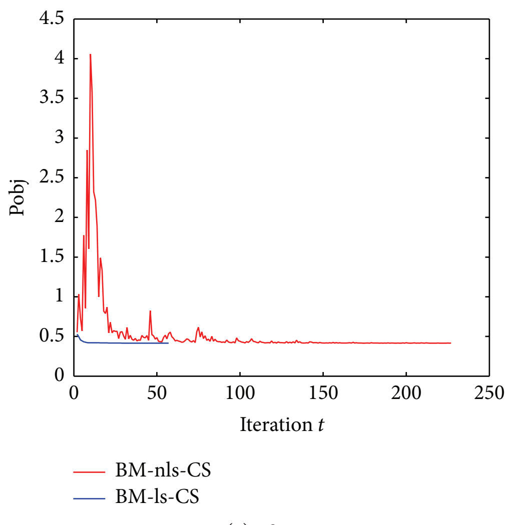
Moreover, Table 3 also shows that if measured by running time, our BM-ls-CS consistently outperforms SVMperf, BM-nls-CS, and LR-ML$^E$ on all benchmark datasets. For example, when compared with SVMperf, BM-ls-CS performs better in terms of both $F1$ value and CPU time. Especially for CPU time, BM-ls-CS can gain speedups of hundreds orders of magnitude over SVMperf on several experimental datasets. Similar comparison results appear with BM-nls-CS and LR-ML$^E$. Statistics show that our algorithm with line search is significantly faster than those two baselines, while obtaining better (or comparable) $F1$ values. However, for SVM-CS which is implemented by Liblinear, it is a bit different. Experimental results show that our algorithm is faster than SVM-CS on five out of the six datasets (a3a, acoustic, ijcnn1, letter, and satimage) and is only slower than SVM-CS on news20. Statistics demonstrate that BM-ls-CS performs better than SVM-CS in terms of $F1$ value (98.70 versus 81.67), while SVM-CS can achieve speedups of several orders of magnitude over BM-ls-CS (0.21 s versus 12.64 s). The reason maybe lies in their different inner optimizers. SVM-CS adopts Liblinear, which is specially designed for the dataset with large features, while the bundle method we use does not give special consideration of this situation (note that the cutting plane algorithm which SVMperf uses also does not consider this situation).

All the statistical data above proves that as an $F$-measure based learner, our BM-ls-CS is both efficient and effective, when compared with other existing baselines.

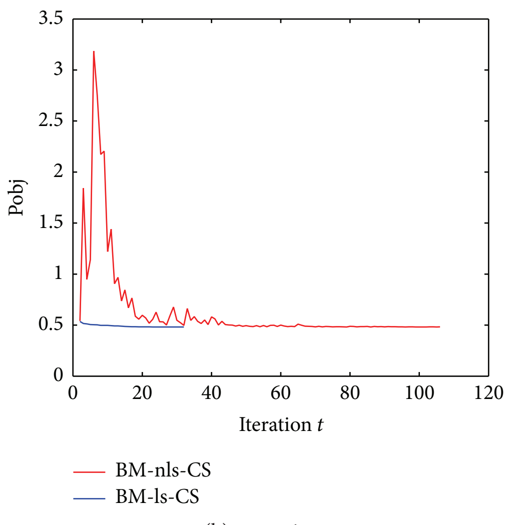
Finally, from Table 3 we can observe that, although BM-nls-CS and SVMperf solve the same equivalent problem with similar optimization technique, their performances are quite different. Experimental results show that BM-nls-CS is faster than SVMperf on all the datasets, and it is largely due to their different implementations (e.g., QP solver). Therefore, in order to see whether our line search technology can enhance the convergence speed, in the following we will only compare the BM-ls-CS with BM-nls-CS for fair play.

*4.2.2. The Comparison between BM-ls-CS and BM-nls-CS.* In the second part of experiments, we are interested in the convergence speed between our algorithm and BM-nls-CS, which uses the same inner solver without line search. Thus, we consider the number of iterations used in reducing the primal objective value, and Figure 2 gives the objective value as a function of training iterations for the two algorithms on various datasets.

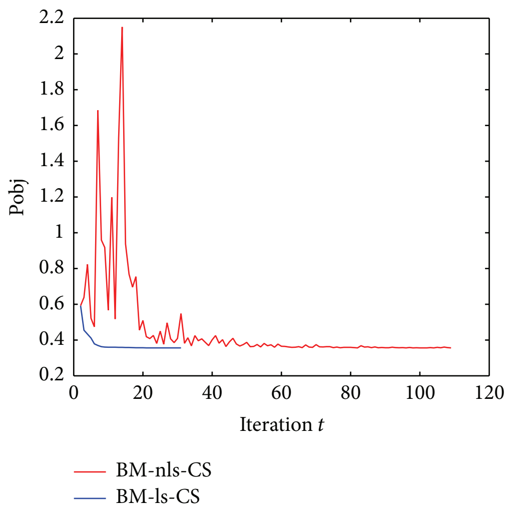
From Figure 2, we can find that even though the BM-nls-CS ultimately converges to the minimum, its values heavily fluctuate during the iterations. The reason for these fluctuations lies in the fact that, during the iterations, the cutting plane selected by BM-nls-CS only guarantees that dual value monotonically increases. However, there is no guarantee that such a cutting plane will lead the primal value to monotonically decrease, as the figure depicts that $F(w_{t+1}) > F(w_t)$ often occurs. On the contrary, it is clear from the figure that our algorithm enjoys progression with "strictly" decreasing objective values and achieves speedups of more than one order of magnitude over BM-nls-CS. This fact implies that our line search technology can help to avoid the "stalling" steps and accelerate the convergence speed of algorithm.



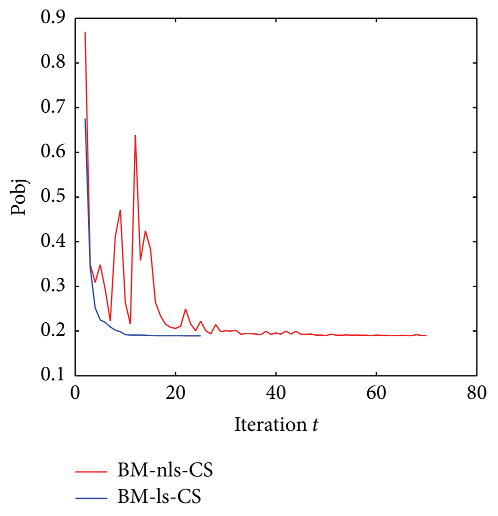
(a) a3a



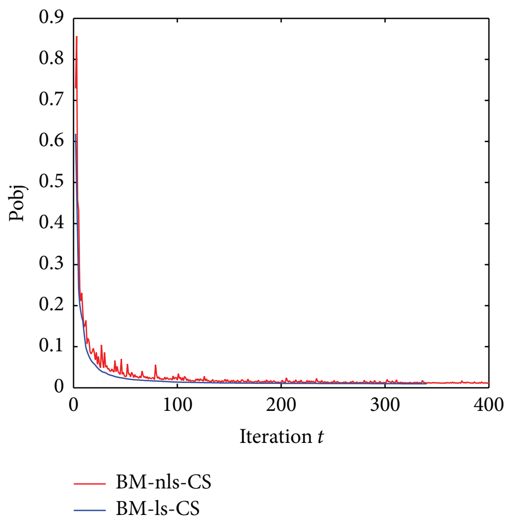
(b) acoustic



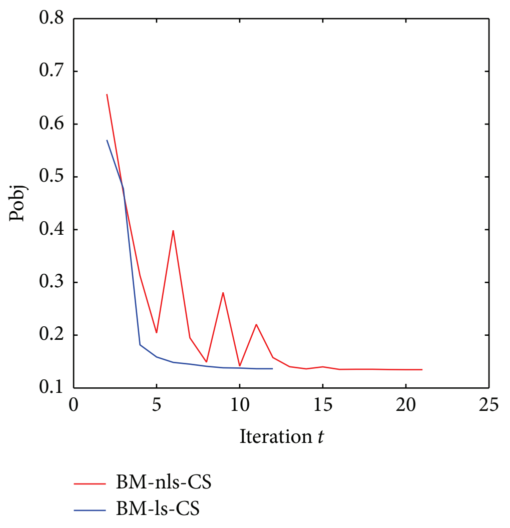
(c) ijcnn1



(d) letter



(e) news20



(f) satimage

FIGURE 2: Convergence behaviors of the BM-ls-CS and BM-nls-CS.

Table 3: The $F1$ value and running time of different algorithms.

| | a3a | acoustic | ijcnn1 | letter | news20 | satimage |
|---|---|---|---|---|---|---|
| SVMperf | 67.08 | 60.88 | 47.16 | 52.38 | 98.04 | 95.58 |
| | 81.81 s | 86.81 s | 75.89 s | 1.53 s | 71.95 s | 4.30 s |
| SVM-CS | 67.55 | 63.26 | 55.97 | 63.12 | 81.67 | 95.78 |
| | 3.22 s | 0.97 s | 0.84 s | 0.68 s | 0.21 s | 0.02 s |
| BM-nls-CS | 67.38 | 64.34 | 56.13 | 63.37 | 98.70 | 96.57 |
| | 0.64 s | 1.13 s | 0.26 s | 0.11 s | 41.04 s | 0.04 s |
| LR-ML$^E$ | 67.36 | 60.22 | 52.84 | 52.57 | 88.65 | 97.39 |
| | 6.97 s | 38.84 s | 5.12 s | 0.19 s | 18.12 s | 0.02 s |
| BM-ls-CS | 67.38 | 64.02 | 56.19 | 63.38 | 98.70 | 96.61 |
| | 0.19 s | 0.49 s | 0.14 s | 0.03 s | 12.64 s | 0.01 s |

## 5. Conclusion

In this paper, we have presented a novel cost-sensitive SVM algorithm that can optimize $F$-measure efficiently. We began our work with an explicit transformation from maximizing $F$-measure to cost-sensitive classification and then proposed to use the bundle method for the inner optimization, which had $O(1/\varepsilon)$ rate of convergence. For the problem where the existing bundle method only guaranteed that the dual objective increases monotonically and did not guarantee that the primal objective decreases monotonically, an efficient line search algorithm has been proposed, which can avoid this undesirable effect, and accelerated the practical convergence speed of our BM-ls-CS algorithm. Experiments on the benchmark datasets showed that when compared with other existing $F$-measure based learners, BM-ls-CS we proposed not only gave better generalization performance, but also provided significant speedups during the training. There are two issues that are worthy of further investigations in the future. The first topic is to extend our approach to other imbalanced measures such as PAUC [23] or SAUC [24] and design the efficient algorithms for optimizing these metrics. The second one is to solve our problem from the view of Multiobjective Optimization, since recent works on MOO [25, 26] show that a cost-sensitive problem can be regarded as a multiobjective problem. In the future, we plan to produce an efficient $F$-measure classifier through the Multiobjective Optimization.

## Competing Interests

The authors have declared that no conflict of interests exists.

## 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. This work is also supported by Co-Innovation Center for Information Supply & Assurance Technology, Anhui University.

## References

[1] C. van Rijsbergen, "Foundation of evaluation," *Journal of Documentation*, vol. 30, no. 4, pp. 365–373, 1974.

[2] H. Daumé III, J. Langford, and D. Marcu, "Search-based structured prediction," *Machine Learning*, vol. 75, no. 3, pp. 297–325, 2009.

[3] J. F. Díez-Pastor, J. J. Rodríguez, C. I. García-Osorio, and L. I. Kuncheva, "Diversity techniques improve the performance of the best imbalance learning ensembles," *Information Sciences*, vol. 32, no. 5, pp. 98–117, 2015.

[4] B. Zhao, J. Wang, M. Li, F.-X. Wu, and Y. Pan, "Detecting protein complexes based on uncertain graph model," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 486–497, 2014.

[5] F. Firoozbakht, I. Rezaeian, L. Porter, and L. Rueda, "Breast cancer subtype identification using machine learning techniques," in *Proceedings of the IEEE 4th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS '14)*, pp. 1–2, IEEE, Miami, Fla, USA, June 2014.

[6] T. D. Imler, J. Morea, C. Kahi et al., "Multi-center colonoscopy quality measurement utilizing natural language processing," *The American Journal of Gastroenterology*, vol. 110, no. 4, pp. 543–552, 2015.

[7] Z. Munkhjargal, G. Bella, A. Chagnaa et al., "Named entity recognition for Mongolian language," in *Text, Speech, and Dialogue*, pp. 243–251, Springer, 2015.

[8] N. Ye, K. M. A. Chai, W. S. Lee et al., "Optimizing F-measures: a tale of two approaches," in *Proceedings of the International Conference on Machine Learning (ICML '12)*, pp. 156–163, Edinburgh, UK, June 2012.

[9] D. R. Musicant, V. Kumar, and A. Ozgur, "Optimizing F-measure with support vector machines," in *Proceedings of the 16th IAIRSC*, pp. 356–360, 2003.

[10] Z. Liu, M. Tan, and F. Jiang, "Regularized F-measure maximization for feature selection and classification," *Journal of Biomedicine and Biotechnology*, vol. 2009, Article ID 617946, 8 pages, 2009.

[11] 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.

[12] J. Suzuki, E. McDermott, and H. Isozaki, "Training conditional random fields with multivariate evaluation measures," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL '06)*, pp. 217–224, Association for Computational Linguistics, 2006.

[13] W. Cheng, K. Dembczyński, E. Hüllermeier, A. Jaroszewicz, and W. Waegeman, "F-measure maximization in topical classification," in *Rough Sets and Current Trends in Computing: 8th International Conference, RSCTC 2012, Chengdu, China, August 17–20, 2012. Proceedings*, vol. 7413 of *Lecture Notes in Computer Science*, pp. 439–446, Springer, Berlin, Germany, 2012.

[14] P. M. Chinta, P. Balamurugan, S. Shevade, and M. N. Murty, "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, Dallas, Tex, USA, August 2013.

[15] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Huellermeier, "Optimizing the F-measure in multi-label classification: plug-in rule approach versus structured loss minimization," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 1130–1138, 2013.

[16] S. P. Parambath, N. Usunier, and Y. Grandvalet, "Optimizing F-measures by cost-sensitive classification," in *Advances in Neural Information Processing Systems*, pp. 2123–2131, Curran Associates, 2014.

[17] K. J. Dembczynski, W. Waegeman, W. Cheng et al., "An exact algorithm for F-measure maximization," in *Advances in Neural Information Processing Systems*, pp. 1404–1412, 2011.

[18] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," in *Machine Learning and Knowledge Discovery in Databases*, T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, Eds., vol. 8725 of *Lecture Notes in Computer Science*, pp. 225–239, Springer, Berlin, Germany, 2014.

[19] W. Waegeman, K. Dembczynski, A. Jachnik, W. Cheng, and E. Hüllermeier, "On the Bayes-optimality of F-measure maximizers," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3333–3388, 2014.

[20] N. Natarajan, O. Koyejo, P. Ravikumar, and I. S. Dhillon, "Optimal decision-theoretic classification using non-decomposable performance metrics," http://arxiv.org/abs/1505.01802.

[21] V. Franc and S. Sonnenburg, "Optimized cutting plane algorithm for large-scale risk minimization," *The Journal of Machine Learning Research*, vol. 10, pp. 2157–2192, 2009.

[22] C. H. Teo, S. V. N. Vishwanathan, A. J. Smola, and Q. V. Le, "Bundle methods for regularized risk minimization," *The Journal of Machine Learning Research*, vol. 11, pp. 311–365, 2010.

[23] H. Narasimhan and S. Agarwal, "A structural SVM based approach for optimizing partial AUC," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 516–524, Atlanta, Ga, USA, June 2013.

[24] S. Wu, P. Flach, and C. Ferri, "An improved model selection heuristic for AUC," in *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17–21, 2007. Proceedings*, vol. 4701 of *ecture Notes in Computer Science*, pp. 478–489, Springer, Berlin, Germany, 2007.

[25] X.-Y. Zhang, Y. Tian, and Y.-C. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[26] X.-Y. Zhang, Y. Tian, R. Cheng, and Y.-C. Jin, "An efficient approach to nondominated sorting for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.