Minimax Sparse Logistic Regression for Very High-Dimensional Feature Selection

Mingkui Tan, Ivor W. Tsang, and Li Wang

***Abstract*— Because of the strong convexity and probabilistic underpinnings, logistic regression (LR) is widely used in many real-world applications. However, in many problems, such as bioinformatics, choosing a small subset of features with the most discriminative power are desirable for interpreting the prediction model, robust predictions or deeper analysis. To achieve a sparse solution with respect to input features, many sparse LR models are proposed. However, it is still challenging for them to efficiently obtain unbiased sparse solutions to very high-dimensional problems (e.g., identifying the most discriminative subset from millions of features). In this paper, we propose a new minimax sparse LR model for very high-dimensional feature selections, which can be efficiently solved by a cutting plane algorithm. To solve the resultant nonsmooth minimax subproblems, a smoothing coordinate descent method is presented. Numerical issues and convergence rate of this method are carefully studied. Experimental results on several synthetic and real-world datasets show that the proposed method can obtain better prediction accuracy with the same number of selected features and has better or competitive scalability on very high-dimensional problems compared with the baseline methods, including the ℓ_1-regularized LR.**

***Index Terms*— Feature selection, minimax problem, single-nucleotide polymorphism (SNP) detection, smoothing method, sparse logistic regression.**

I. Introduction

BECAUSE of the strong convexity and probabilistic underpinnings [7], logistic regression (LR) is widely studied and used in many applications [8], [14], [17], [39]. Compared with support vector machine (SVM), the advantages of LR are its posterior model for model selection and its probabilistic output for uncertainty prediction, which can be used for comparing classifier outputs, especially for multiclass prediction. In addition, the logistic loss is twice-differentiable and strongly convex, which is good for faster optimizations [15], [33]. Given a set of labeled examples $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^m$ is the input and $y_i \in \{\pm 1\}$ is the output, in LR, a linear decision function $f(x) = \mathbf{w}'\mathbf{x}$ is learned to minimize a regularized negative log-likelihood [33]

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \log\left(1 + e^{-y_i\mathbf{w}'\mathbf{x}_i}\right) \qquad (1)$$

where $C > 0$ is a tradeoff parameter.

In the standard LR, the ℓ_2-norm regularizer can only induce dense solutions. However, great attention has been drawn to achieving sparse LR that has several attractive properties, such as robustness to noise [26], [35]. In addition, in data mining and machine learning applications, a sparse decision rule with respect to input features, which is also known as feature selection [6], [25], [34], [37], is desirable for faster prediction or better pattern interpretation. For example, in the microarray data analysis, the number of genes can exceed several thousands. However, a compact gene subset of tens of features can provide better prediction ability than that with all genes [11], [40]. Similar observation can also be found in the single-nucleotide polymorphism (SNP) detection [3]. With the fast development of biotechnologies, SNP becomes a viable tool for bio-studies. An SNP measures the DNA sequence variation when a single nucleotide (A, T, C, or G) in the genome differs between members of a biological species or paired chromosomes. Usually, the number of SNPs can exceed 100 000, but only a small number of SNPs are highly related to a given disease [3]. Considering that the large number of irrelevant SNPs may seriously deteriorate the prediction ability, the detection of the most informative SNPs is very important for nucleotide-level disease diagnosis.

To select the most informative features regarding the output $\mathbf{y}$, a direct way is to impose an ℓ_0-constraint to (1). Specifically, suppose r features are expected to be selected, we can impose an ℓ_0-constraint $\|\mathbf{w}\|_0 \leq r$ to (1), resulting in the following ℓ_0-constrained problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \log(1 + e^{-y_i\mathbf{w}'\mathbf{x}_i}): \text{ s.t. } \|\mathbf{w}\|_0 \leq r. \quad (2)$$

The nonconvexity of the ℓ_0 constraint makes the above problem challenging to solve. Instead, many researchers resort to solving its relaxations. Among them, the ℓ_1-norm convex relaxation is widely studied. The ℓ_1-regularized LR (ℓ_1-LR) solves the following minimization problem:

$$\min_{\mathbf{w}} \|\mathbf{w}\|_1 + C\sum_{i=1}^n \log(1 + e^{-y_i\mathbf{w}'\mathbf{x}_i}). \qquad (3)$$

Manuscript received July 12, 2012; revised March 13, 2013; accepted May 5, 2013. Date of publication June 17, 2013; date of current version September 27, 2013. This work was supported in part by the Singapore NTU A*SERC under Grant 112 172 0013.

M. Tan and I. W. Tsang are with the School of Computer Engineering, Nanyang Technological University, 639798 Singapore (e-mail: tanm0097@e.ntu.edu.sg; IvorTsang@ntu.edu.sg).

L. Wang is with the Department of Mathematics, University of California, San Diego, CA 92093 USA (e-mail: liw022@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2263427

By changing C, ℓ_1-LR can induce sparse solutions of different levels [36]. Recent studies about ℓ_1-LR focus on the improvement of the training efficiency [26]. Thorough comparisons of several state-of-the-art ℓ_1-LR algorithms can be referred in [36].

Although widely used, ℓ_1-LR has several limitations. For example, as $\|\mathbf{w}\|_1$ is to sum up the absolute values of all w_i, it is sensitive to their scales and may lead to biased or suboptimal solution [37]. To overcome this drawback, several nonconvex relaxation methods, such as weighted ℓ_1-regularization [4], [37], are proposed. Recent studies from [16] and [37] show that they can achieve lower predictive risk. Unfortunately, since these methods involve many times of ℓ_1-LR training, they are very expensive in computation. To select the most informative features, recently, many researchers proposed to use greedy methods, which iteratively include one feature into a feature subset and conduct the optimization with the selected features [18], [29]. A group orthogonal matching pursuit (GOMP) for LR was proposed in [18], which can be generalized to single-feature case with group size of one. In [29], a more general greedy scheme was presented, which includes GOMP as a special case. Since only one feature (or one group) is selected in each iteration, the major drawback of greedy methods is their high computational cost when selecting a relatively large number of features. In addition, as there is no regularizer used in the objective function, the over-fitting problem may happen [18], [29]. Recently, a new convex relaxation for SVM, called the Feature Generating Machine (FGM), was proposed in [27]. However, several issues of this method are not addressed. For example, the tightness of the convex relaxation was unknown. The training efficiency of FGM is also limited due to the inefficient solver [27].

Regarding the above issues, this paper focuses attention on the linear feature selection of very high-dimensional problems. To reduce the high-computational cost and the bias problem of the existing feature selection methods, this paper present a new minimax sparse LR (MSLR) model. In summary, the contributions of this paper are listed as follows.

1) A new weight scaling scheme is proposed for sparse LR, which can be further transformed as a minimax optimization problem.
2) A smoothing coordinate descent method is proposed to solve the nonsmooth minimax problems, which significantly improves the training efficiency.
3) The weakly linear convergence rate of the smoothing coordinate descent algorithm is verified.

The rest of this paper is organized as follows. In Section II, a new sparse minimax LR model is presented. The smoothing coordinate descent method is introduced in Section III. The experimental results and conclusion remarks are given in Sections IV and V, respectively.

II. Minimax Sparse LR

A. Notations Definitions

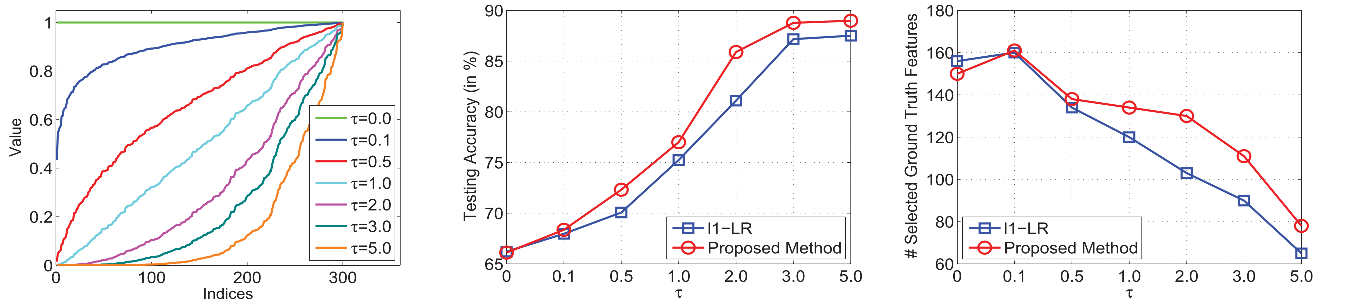
Throughout this paper, we denote the transpose of vector/matrix by the superscript $'$, all-one vector by $\mathbf{1} \in \mathbb{R}^n$, the element wise product between two matrices $\mathbf{A}$ and $\mathbf{B}$ by $\mathbf{A} \odot \mathbf{B}$, the ℓ_2-norm of any vector $\mathbf{x}$ by $||\mathbf{x}|| = \sqrt{\mathbf{x}'\mathbf{x}}$, the ℓ_p-norm of any vector $\mathbf{x}$ by $||\mathbf{x}||_p$ and the matrix norm of any matrix $\mathbf{A}$ by $||\mathbf{A}|| = \max_{||\mathbf{x}||=1} ||\mathbf{A}\mathbf{x}||$. In addition, let $||\mathbf{x}||_0$ denote the zero norm of a vector $\mathbf{x}$ which counts the number of nonzero elements in $\mathbf{x}$, and diag($\mathbf{d}$) denote a diagonal matrix with the elements of $\mathbf{d}$ on the diagonal.

B. Minimax Sparse LR

For the standard linear LR in (1), the ℓ_2-regularizer $\|\mathbf{w}\|^2$ is used to avoid the over-fitting problem but cannot induce sparse solutions. To achieve the sparsity, one can impose the ℓ_1-constraint $\|\mathbf{w}\|_1 \leq r$ on the logistic loss model [36]. However, there are several deficiencies regarding the ℓ_1-regularizer. At first, the performance of ℓ_1-regularized methods may face the bias risks brought by the scale variation of $\mathbf{w}$ [16], [37]. Specifically, if $|w_i| \in \{0, 1\}$, we have $\|\mathbf{w}\|_1 = \|\mathbf{w}\|_0$, and ℓ_1-regularized methods can perform well. However, in practice, $|w_i|$ can follow any distribution, making $\|\mathbf{w}\|_1$ very far away from $\|\mathbf{w}\|_0$, which leads to bias in ℓ_1-regularization [37].

To show the influences brought by scale variations, we conduct a synthetic experiment on several toy problems of size 4096×8192, where half of the instances of $\mathbf{X}$ are used for training and the rest for testing. To generate the ground-truth informative features, we manually generate a sparse vector $\mathbf{w} \in \mathbb{R}^m$ and then produce the output label by $\mathbf{y} = \text{sign}(\mathbf{X}\mathbf{w})$. Since only those features with nonzero w_is contribute to the output, they are deemed as the ground-truth informative features. Then the task of feature selection is to recover these features from the output label $\mathbf{y}$. Several types of $\mathbf{w}$ with different scale variations are studied, which is detailed in the caption of Fig. 1. In Fig. 1(a), we showed the distribution of the nonzero $|w_j|$ of different $\mathbf{w}$s, the prediction accuracy and the number of recovered ground-truth features for ℓ_1-LR. It shows that when $|w_j| \in \{0, 1\}$, ℓ_1-LR has similar performance with the proposed method of this paper. But when the scale variation becomes larger, where $\mathbf{w}$ has a long tail, ℓ_1-LR shows poorer performance than the proposed method in terms of prediction accuracy and number of recovered ground-truth features. In other words, the bias problem for ℓ_1-LR happens. Also due to the scale variation of $\mathbf{w}$, it is hard to control the number of features. However, in many real applications, a controllable number of selected features is desired. For example, in biology study, biologists prefer to select hundreds of SNPs or genes for further studies. The final issue for ℓ_1-LR is that it is expensive to be solved for very high-dimensional problems. Recently, a very fast ℓ_1 solver are developed based on the coordinate descent [36]. However, the efficiency is still limited on high-dimensional dense datasets.

To address the above issues, we propose a novel weight scaling scheme for sparse LR as follows. To be more specific, we introduce a weight scaling vector $\mathbf{d} \in [0, 1]^m$ to the regressor $\mathbf{w} \in \mathbb{R}^m$, and then impose an additional ℓ_1 constraint $||\mathbf{d}||_1 \leq r$ to control the sparsity. Given $\mathbf{d}$, a feature j is not selected if and only if $d_j = 0$; otherwise, the associate feature j will be selected. For simplicity, let $\mathcal{D} = \big\{\mathbf{d}\big|\|\mathbf{d}\|_1 = \sum d_j \leq r, d_j \in [0, 1], \; j = 1, \ldots, m\}$ be the domain of $\mathbf{d}$. By taking these elements into consideration, we propose to solve the



(a) Distribution of the nonzero ground truth $|w_i|$. (b) Testing accuracy with different τ. (c) # recovered ground-truth features.

Fig. 1. Demonstration of the influence brought by the scale variations. To generate synthetic problems of different scale variations, we first generate a sparse vector $\mathbf{v}$ of 300 nonzero entries sampled from the Uniform distribution $\mathcal{U}(0,1)$, and then produce $\mathbf{w}$ by $w_i = v_i^\tau$ of randomly assigned signs, where τ is chosen from $\{0, 0.1, 0.5, 1.0, 2.0, 3.0, 5.0\}$. For simplicity, all the nonzero w_is are normalized to $[-1, 1]$. The value of τ determines the scale variations for nonzero w_is. To be more specific, when $\tau = 0$, we have $|w_i| \in \{0, 1\}$ and $\|\mathbf{w}\|_1 = \|\mathbf{w}\|_0$. However, when τ becomes large, w_i will appear as a long tail, as shown in Fig. 1(a), where the nonzero entries of $\mathbf{w}$ is normalized into $[-1, 1]$ and sorted by $|w_i|$ in ascending order. (a) Distribution of the nonzero ground truth $|w_i|$. (b) Testing accuracy with different τ. (c) No. of recovered ground-truth features.

following joint minimization problem regarding $\mathbf{d}$ and $\mathbf{w}$ with $\mathbf{d} \in \mathcal{D}$:

$$\min_{\mathbf{d}\in\mathcal{D}} \min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{m} d_j w_j^2 + Cl(\mathbf{w}\odot\mathbf{d}) \qquad (4)$$

where d_j can be considered as a scaling factor on w_j. Apparently, at the optimality of the above problem, the value of w_j will be 0 if $d_j = 0$. It is worth mentioning that, the scalar r in (4) has practical meanings. Particularly, it can be considered as a conservative estimation to k. Specifically, suppose about k features are expected to be chosen, we constrain that $1 \le r < k$. In practice, r can be several times smaller than k.

The above model has several advantages. At first, as d_j is in $[0, 1]$, the influence of the scale variation of w_j can be alleviated by setting a relatively large tradeoff parameter C. Secondly, an efficient optimization scheme can be introduced to solve it. Last but not least, it is much easier to control the number of selected features through the parameter r, which will be shown later.

Problem (4) can be directly solved with $2m$ optimization variables. However, it is very expensive for large m. To address this issue, we first make the following transformations.

Proposition 1: Let $\mathbf{Q} = (\mathbf{X}'\text{diag}(\mathbf{d})\mathbf{X}) \odot (\mathbf{y}\mathbf{y})'$, where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, $\mathbf{y} = [y_1, \ldots, y_n]'$, by introducing the dual form of the inner minimization problem regarding $\mathbf{w}$, (4) can be transformed as a minimax optimization problem

$$\min_{\mathbf{d}\in\mathcal{D}} \max_{\boldsymbol{\alpha}} -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha} - G(\boldsymbol{\alpha}) : \text{ s.t } 0 \le \alpha_i \le C \quad \forall i \qquad (5)$$

where $\boldsymbol{\alpha}$ is the dual variable and $G(\boldsymbol{\alpha}) = \sum_{i:\alpha_i>0} \alpha_i \log(\alpha_i) + \sum_{i:\alpha_i<C}(C-\alpha_i)\log(C-\alpha_i)$.

Proof: By defining $0\log(0) = 0$, we first derive the dual form of the inner minimization problem with fixed $\mathbf{d} \in \mathcal{D}$. Let $\xi_i = -y_i(\mathbf{w}\odot\mathbf{d})'\mathbf{x}_i = -y_i\mathbf{w}'(\mathbf{x}_i\odot\mathbf{d})$ and $g(\xi_i) = \log(1+e^{\xi_i}) = \log(1+e^{-y_i\mathbf{w}'(\mathbf{x}_i\odot\mathbf{d})})$, given $\mathbf{d}$, the inner problem of the ℓ_1 LR becomes

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{j=1}^{m} d_j w_j^2 + C\sum_{i=1}^{n} g(\xi_i), \text{ s.t. } \xi_i = -y_i\mathbf{w}'(\mathbf{x}_i\odot\mathbf{d}).$$

By introducing dual variables $\boldsymbol{\alpha}$ to the constraints, the Lagrangian function of the above problem is $\mathcal{L}(\mathbf{w},\boldsymbol{\alpha}) = \frac{1}{2}\sum_{j=1}^{m} d_j w_j^2 + C\sum_{i=1}^{n} g(\xi_i) + \sum_{i=1}^{n}\alpha_i(-\xi_i - y_i\mathbf{w}'(\mathbf{x}_i \odot d))$. The KKT condition can be obtained by: $\nabla_{w_j}\mathcal{L} = d_j w_j - \sum_{i=1}^{n}\alpha_i y_i(x_{ij}d_j) = 0 \Rightarrow d_j w_j = d_j\sum_{i=1}^{n}\alpha_i y_i x_{ij}$; $\nabla_{\xi_i}\mathcal{L} = Cg'(\xi_i) - \alpha_i = 0 \Rightarrow \xi_i = g'^{-1}(\frac{\alpha_i}{C})$. In addition, we have $g'^{-1}(z) = \log\frac{z}{1-z}$, hence $\xi_i = \log\frac{\alpha_i}{C-\alpha_i}$ and $g(\xi_i) = \log\frac{C}{C-\alpha_i}$. Substitute all the equations into the Lagrange function, we can obtain: $\mathcal{L}(\mathbf{w},\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{j=1}^{m} d_j w_j^2 + nC\log C - \sum_{i=1}^{n}(C-\alpha_i)\log(C-\alpha_i) - \sum_{i=1}^{n}\alpha_i\log\alpha_i$. With the definition of $g(\xi_i) = \log\frac{C}{C-\alpha_i} = \log(1+e^{\xi_i})$, we have the constraint for $\boldsymbol{\alpha}$: $\boldsymbol{\alpha} \in [0, C]^n$ [13]. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $\mathbf{Q} = (\mathbf{X}'\text{diag}(\mathbf{d})\mathbf{X}) \odot (\mathbf{y}\mathbf{y})'$, the Lagrangian dual is to maximize $\mathcal{L}(\mathbf{w},\boldsymbol{\alpha})$ regarding the dual variable $\boldsymbol{\alpha}$, which can be simplified as: $\max_{\boldsymbol{\alpha}} \; -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha} - G(\boldsymbol{\alpha}) \;:\; \text{s.t. } \boldsymbol{\alpha} \in [0, C]^n$, where $G(\boldsymbol{\alpha}) = \sum_{i=1}^{n}(C-\alpha_i)\log(C-\alpha_i) + \sum_{i=1}^{n}\alpha_i\log\alpha_i$ and $\sum_{j=1}^{m} d_j w_j^2 = \boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha}$. In order to select the most informative features, we have to find the best $\mathbf{d}$ that minimizes the logistic loss. Easily, this is to solve the minimax problem (5). This completes the proof. ■

For simplicity, we define the domain of $\boldsymbol{\alpha}$ as $\mathcal{A} = \{\boldsymbol{\alpha} | 0 \le \alpha_i \le C, i = 1, \ldots, n\}$. Let $c_j = \sum_{i=1}^{n}\alpha_i y_i x_{ij}$, and we have $f(\boldsymbol{\alpha},\mathbf{d}) = \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha} + G(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{j=1}^{m} c_j^2 d_j + G(\boldsymbol{\alpha})$. Apparently, $f(\boldsymbol{\alpha},\mathbf{d})$ is convex in $\boldsymbol{\alpha}$ and linear in $\mathbf{d}$. Now since both $\mathcal{A}$ and $\mathcal{D}$ are compact, the following relation holds from the minimax theorem [28].

Theorem 1: Given the above definition of $\mathcal{A}$ and $\mathcal{D}$ for $f(\boldsymbol{\alpha},\mathbf{d})$, we have the following:

$$\min_{\mathbf{d}\in\mathcal{D}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} -f(\boldsymbol{\alpha},\mathbf{d}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\mathbf{d}\in\mathcal{D}} -f(\boldsymbol{\alpha},\mathbf{d}). \qquad (6)$$

With the above equivalence, the proposed weight scaling LR problem can be addressed by solving $\max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\mathbf{d}\in\mathcal{D}} -f(\boldsymbol{\alpha},\mathbf{d})$. It can be further transformed as a nonlinear constrained optimization problem [23]

$$\min_{\boldsymbol{\alpha}\in\mathcal{A},\theta} \quad \theta : \text{ s.t. } f(\boldsymbol{\alpha},\mathbf{d}) - \theta \le 0 \;\; \forall\, \mathbf{d} \in \mathcal{D}, \;\; \theta \in \mathbb{R}. \qquad (7)$$

The above problem is a convex programming problem. Therefore, it can be solved globally. In addition, as the compact domain $\mathcal{D}$ contains infinite number of constraints, it is a semi-infinite programming (SIP) problem.

Algorithm 1 Cutting Plane Algorithm for Solving (7)

```
Initialization: set $\boldsymbol{\alpha} = \mathbf{1}$, $\mathcal{C} = \emptyset$ and $t = 1$.
0: Find the most-active-constraint $\mathbf{d}^t$ by solving (8), and set $\mathcal{C} = \mathcal{C} \bigcup \{\mathbf{d}^t\}$.
1: Given the reduced constant set $\mathcal{C}$, solve the subproblem (9).
2: Let $t = t + 1$. Repeat step 0-2 until convergence.
```

Algorithm 2 General Smoothing Method

```
Initialization: set $\boldsymbol{\alpha}^0 \in (0, C)^n$, $q_0 > 0$ and $k = 0$.
0: If the stopping criterion is achieved, go to Step 3.
1: Given $q_k$, update $\boldsymbol{\alpha}^k$ to $\boldsymbol{\alpha}^{k+1}$ to make a sufficient decrease of $f(\boldsymbol{\alpha}, q)$.
2: Update $q_{k+1}$. Set $k = k + 1$ and go to Step 0.
3: Stop and output $\boldsymbol{\alpha}^*$.
```

C. Optimization Strategies

The SIP problem (7) in general is difficult to solve as there are infinite number of constraints involved. However, in the feature selection task, there should be a few active constraints. Based on this fact, the cutting plane algorithm can be adopted to efficiently solve it [20]. The general scheme is simplified in Algorithm 1. The basic idea of the cutting plane algorithm is to iteratively add one or several active constraints to the active set and then solve a reduced subproblem with selected constraints. Accordingly, the computational cost can be greatly reduced.

In Algorithm 1, a critical problem is to find the most-active-constraint from the infinite number of candidates. Typically, we need to solve $\max_{\mathbf{d} \in \mathcal{D}} \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}\boldsymbol{\alpha}$, which is equivalent to

$$\max_{\mathbf{d} \in \mathcal{D}} \quad \frac{1}{2}\sum_{j=1}^{m} c_j^2 d_j \tag{8}$$

where $c_j = \sum_{i=1}^{n} \alpha_i y_i x_{ij}$ is the importance of the jth feature. To obtain the most-active $\mathbf{d}$, we can first find the r largest c_j^2, and then assigning those d_j to 1 and the rest to 0. Once an active $\mathbf{d}$ is found, we can update $\mathcal{C}$ by $\mathcal{C} = \mathcal{C} \cup \{\mathbf{d}\}$. Let $|\mathcal{C}| = T$, the remaining problem is to solve a reduced problem of (7) with the active constraints defined by $\mathcal{C}$

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}, \theta} \quad \theta : \text{ s.t. } f(\boldsymbol{\alpha}, \mathbf{d}^t) - \theta \le 0, \; \mathbf{d}^t \in \mathcal{C}, \; t = 1, \ldots, T. \tag{9}$$

This minimax subproblem can be solved by the reduced-gradient method [24]. Actually, by applying Lagrangian theory, one can arrive at the dual form of (9) as follows:

$$\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2}\boldsymbol{\alpha}'\Big(\sum \mu_t \mathbf{Q}_t\Big)\boldsymbol{\alpha} - G(\boldsymbol{\alpha}) : \text{ s.t. } \boldsymbol{\mu}'\mathbf{1} = 1, \mu_t \ge 0$$

where $\mathbf{Q}_t = (\mathbf{X}'\text{diag}(\mathbf{d}^t)\mathbf{X}) \odot (\mathbf{y}\mathbf{y}')$ and $\boldsymbol{\mu}$ is the dual variable for the quadratic constraints. Let $\boldsymbol{\mu}^*$ be the optimal solution to (10), the optimal kernel matrix can be obtained by $\sum \mu_t^* \mathbf{Q}_t = (\mathbf{X}'(\sum \mu_t^* \text{diag}(\mathbf{d}^t))\mathbf{X}) \odot (\mathbf{y}\mathbf{y}')$. Therefore, the optimal $\mathbf{d}^*$ can be obtained by $\sum_t \mu_t^* \mathbf{d}^t$. Notice that (10) is essentially the SimpleMKL problem and can be addressed by a subgradient method [21], [24]. However, the efficiency of this method is limited since it needs many times of classier (SVM or LR) training to compute the subgradient [24]. In addition, due to numerical issues, it is difficult to obtain a high precision solution for classier training, thus the subgradient may not be accurate enough to make it converge quickly.

III. Smoothing Coordinate MSLR

Now we focus on solving the minimax problem (9): $\min_{\boldsymbol{\alpha} \in \mathcal{A}} \max_{\mathbf{d}^t \in \mathcal{C}} f(\boldsymbol{\alpha}, \mathbf{d}^t)$. Without loss of generality, suppose $T = |\mathcal{C}|$, (9) can be equivalently written as follows:

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\alpha}) : \qquad f(\boldsymbol{\alpha}) = \max_{t=1,\ldots,T} \; f_t(\boldsymbol{\alpha}), \; \mathbf{d}^t \in \mathcal{C} \tag{10}$$

where $f_t(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}_t\boldsymbol{\alpha} + G(\boldsymbol{\alpha})$ and $f(\boldsymbol{\alpha})$ are a nonsmooth function. Notice that, for any constraint defined by $\mathcal{C}$, the resultant problem can be considered a standard LR problem. Therefore, based on the [33, Th. 1], the optimal solution to (10) lies exactly in $(0, C)^n$. In this sense, hereafter we can constrain $\boldsymbol{\alpha}$ in domain $\mathcal{A} = \{\boldsymbol{\alpha} \big| 0 < \alpha_i < C, i = 1, \ldots, n\}$.

A. Smoothing Method for Solving Minimax Problem

Smoothing methods have shown promising results on solving nonsmooth minimax problems [23], [32]. Its basic idea is to solve a sequence of smoothing approximations to $f(\boldsymbol{\alpha})$. Let $q > 0$ be a smoothing parameter, the following smooth function is introduced to approximate $f(\boldsymbol{\alpha})$:

$$f(\boldsymbol{\alpha}, q) = \frac{1}{q} \ln \sum_{t=1}^{T} \exp(q f_t(\boldsymbol{\alpha})). \tag{11}$$

Proposition 2: Let $f(\boldsymbol{\alpha}, q)$ be defined in (11), then: 1) $f(\boldsymbol{\alpha}, q)$ is monotonically decreasing w.r.t. q, and $f(\boldsymbol{\alpha}) \le f(\boldsymbol{\alpha}, q) \le f(\boldsymbol{\alpha}) + \frac{\ln T}{q}$, and 2) $f(\boldsymbol{\alpha}, q)$ is twice continuous differentiable. Let $\widehat{\lambda}_t = \exp(q f_t(\boldsymbol{\alpha}))/\sum_{t=1}^{T} \exp(q f_t(\boldsymbol{\alpha}))$, we have: $\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, q) = \sum_{t=1}^{T} \widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha})$, $\nabla_{\boldsymbol{\alpha}}^2 f(\boldsymbol{\alpha}, q) = \sum_{t=1}^{T} \widehat{\lambda}_t \nabla^2 f_t(\boldsymbol{\alpha}) + \sum_{t=1}^{T}(q\widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha}) \nabla f_t(\boldsymbol{\alpha})') - q(\sum_{t=1}^{T} \widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha}))(\sum_{t=1}^{T} \widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha}))'$ [32].

The smoothing method solves (10) by minimizing a sequence of (11) with gradually increased q. The basic scheme is shown as in Algorithm 2, where one can use a Newton update $\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k - s\nabla^2 f(\boldsymbol{\alpha}, q)^{-1}\nabla f(\boldsymbol{\alpha}, q)$ or a quasi-Newton update to make a sufficient decrease of (11) with fixed q [23]. s is the step size obtained by Armijo's line search. The major difference of smoothing methods from standard optimization methods is that a nondecreasing sequence $\{q_k\}$ should be maintained to make $f(\boldsymbol{\alpha}, q_k)$ gradually approach to $f(\boldsymbol{\alpha})$. For example, the simple geometric increment of q can be used, i.e., $q_{k+1} = \beta q_k$ with $\beta \in (1, +\infty)$ [32].

Because of the low computational cost per iteration, the smoothing method has shown competitive run-time complexity with other methods [23]. However, the standard smoothing method requires that the domain of the variable $\boldsymbol{\alpha}$ is in $\mathbb{R}^n$ [32]. Fortunately, the following proposition shows that the smoothing method is applicable to solve (10) if we keep $\{\boldsymbol{\alpha}^k\}$ always in $\in (0, C)^n$ for all k.

Proposition 3: Algorithm 2 generates a sequence $\{\boldsymbol{\alpha}^k\}$ in $(0, C)^n$ and the limit point $\boldsymbol{\alpha}^*$ is the optimal solution to (9).[1] The proof can be found in Appendix A, which parallelizes the results in [32]. In step 1 of Algorithm 2, usually the Newton or

[1] It is worth mentioning that the smoothing method cannot be applied to FGM with squared hinge loss, where $\boldsymbol{\alpha}$ is in a closed domain $\{\boldsymbol{\alpha}| \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \ge 0, \forall i\}$ [27]. In such case, the convergence of smoothing method cannot be guaranteed anymore.

quasi-Newton update is adopted to make a sufficient decrease of $f(\boldsymbol{\alpha}, q)$ [23]. However, there are several challenges for the traditional smoothing method to solve (10) with direct Newton or quasi-Newton update. At first, the construction of the search direction in Newton or quasi-Newton is very time and memory consuming for large-scale problems. Secondly, the line search of the open domain problem is difficult. That is to say, it is difficult to find a step size s to make sufficient decrease of $f(\boldsymbol{\alpha}, q)$ while keep $\boldsymbol{\alpha}^{k+1} \in (0, C)^n$.

B. Coordinate Descent Update

To address the challenges of the traditional smoothing method, we propose a coordinate descent update. Basically, it does the updating with each component α_i in $\boldsymbol{\alpha}$ iteratively [35], which have three advantages. Firstly, with one dimension updating, it is easy to make the sequence $\{\boldsymbol{\alpha}^k\}$ be always in $(0, C)^n$. Secondly, it is also easy to conduct the line search w.r.t. single variable in $(0, C)$. Finally, the coordinate descent update is scalable for large-scale problems.

Recall that, for a given q, in each iteration of the smoothing method, essentially it solves the following problem:

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}} \quad f(\boldsymbol{\alpha}, q) = \frac{1}{q} \ln \sum_{t=1}^{T} \exp(q f_t(\boldsymbol{\alpha})). \tag{12}$$

Suppose we have $\boldsymbol{\alpha}^k$ at the iteration k, and we need to update the ith component $\boldsymbol{\alpha}^{k,i}$ to obtain $\boldsymbol{\alpha}^{k+1,i}$, then the basic update needs to solve an one variable minimization problem

$$\begin{aligned} &\min_z f(\boldsymbol{\alpha}^k + z e_i, q) = \min_z \frac{1}{q} \ln \sum_{t=1}^{T} \exp(q f_t(\boldsymbol{\alpha}^k + z e_i)) \\ &\text{s.t.} \ -c_1 \leq z \leq c_2 \end{aligned} \tag{13}$$

where $c_1 = \alpha_i^k$ and $c_2 = C - \alpha_i^k$. For each $f_t(\boldsymbol{\alpha})$, let $g(z) = (c_1 + z)\log(c_1 + z) + (c_2 - z)\log(c_2 - z)$ and $h_t(z) = \frac{a_t}{2} z^2 + b_t z$, where $a_t = (\mathbf{Q}^t)_{ii}$, $b_t = (\mathbf{Q}^t \alpha^k)_i$. Furthermore, let $\gamma_t = \exp(\frac{q}{2} \boldsymbol{\alpha}' \mathbf{Q}^t \boldsymbol{\alpha})$ and $p(z) = g(z) + \frac{1}{q} \ln \Big(\sum_{t=1}^{T} \gamma_t \exp(q h_t(z)) \Big)$, then (13) can be written as $\min\limits_z \ \ p(z) : \ s.t. \ -c_1 \leq z \leq c_2$. The following result holds for $p(z)$.

Lemma 1: $p(z)$ attains a unique minimizer $z^* \in (-c_1, c_2)$, where $\nabla p(z^*) = 0$.

Proof: At first, we have

$$\nabla p(z) = \nabla g(z) + \Big(\sum_{t=1}^{T} \lambda_t (a_t z + b_t) \Big)$$

and

$$\begin{aligned} \nabla^2 p(z) = \nabla^2 g(z) + \sum_{t=1}^{T} \Big(\lambda_t \nabla^2 h_t(z) \Big) + q \sum_{t=1}^{T} \Big(\lambda_t \nabla h_t(z)^2 \Big) \\ -q \left(\sum_{t=1}^{T} \lambda_t \nabla h_t(z) \right)^2 \end{aligned}$$

where

$$\begin{aligned} \nabla h_t(z) &= a_t z + b_t \\ \lambda_t &= \frac{\gamma_t \exp(q h_t(z))}{\sum_{t=1}^{T} \gamma_t \exp(q h_t(z))} \\ \nabla^2 h_t(z) &= a_t \\ \nabla g(z) &= \log \frac{c_1 + z}{c_2 - z} \\ \nabla^2 g(z) &= \frac{c_1 + c_2}{(c_1 + z)(c_2 - z)}. \end{aligned}$$

It is easy to verify: $\lim\limits_{z \to -c_1} \nabla p(z) = -\infty$ and $\lim\limits_{z \to c_2} \nabla p(z) = +\infty$. In addition, $\nabla^2 p(z) > 0$, hence $p(z)$ monotonically increases within range $(-c_1, c_2)$. Accordingly, $\nabla p(z)$ has an unique zero point in $(-c_1, c_2)$ and $p(z)$ has an unique minimizer in this range. ■

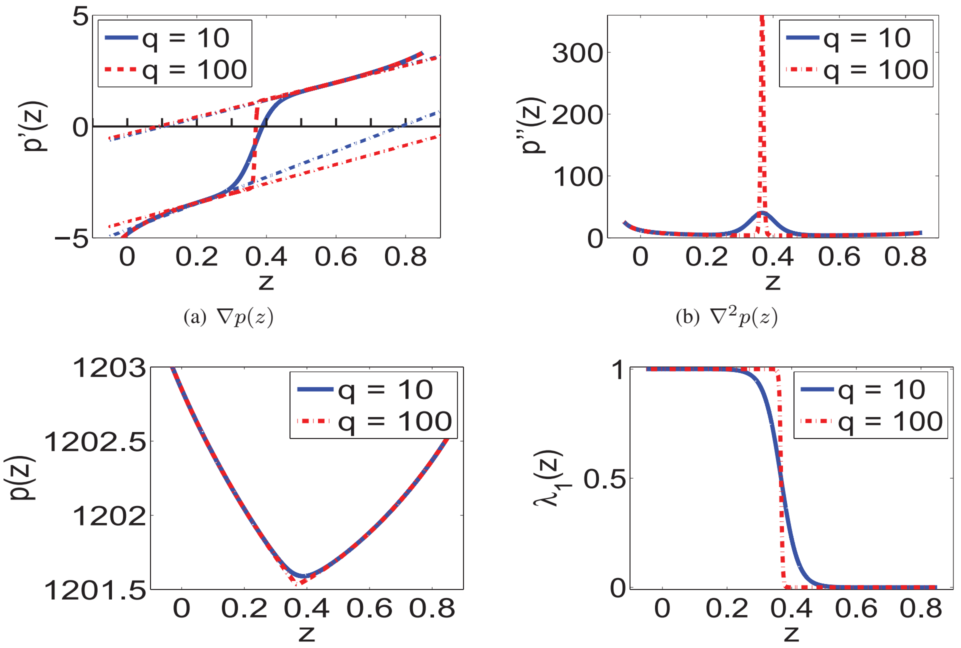
Algorithm 3 Newton-Bisection Search Method

```
Given $a_t$, $b_t$, $c_1$, $c_2$ and $\epsilon$, where $t = 1, ..., T$; set $Z_l = 0$, $Z_u = s$, the iteration index $k = 0$.
0: $o = \begin{cases} 1 \text{ if } \nabla p(z_m) \geq 0 \\ 2 \text{ if } \nabla p(z_m) < 0 \end{cases}$, $Z_u = \begin{cases} z_m + c_1 \text{ if } o = 1 \\ c_2 - z_m \text{ if } o = 2 \end{cases}$.
1: Initialize $Z_o^0 \in (0, s)$
2: If $|\nabla \widehat{p}_o(Z_o^k)| < \epsilon$, go to Step 7.
3: Let $d = -\frac{\nabla \widehat{p}_o(Z_o^k)}{\nabla^2 \widehat{p}_o(Z_o^k)}$;
4: Update $Z_l$ and $Z_u$ by $\begin{cases} Z_l = \max(Z_o^k, Z_l) \text{ if } d > 0 \\ Z_u = \min(Z_o^k, Z_u) \text{ if } d < 0 \end{cases}$;
5: $Z_o^{k+1} = \begin{cases} Z_o^k + d \text{ if } Z_o^k + d \in (Z_l, Z_u) \\ \frac{(Z_l + Z_u)}{2} \quad \text{Otherwise} \end{cases}$;
6: Let $k = k + 1$, go to Step 2.
7: Stop, set $\begin{cases} Z_2^k = s - Z_1^k \text{ if } o = 1 \\ Z_1^k = s - Z_2^k \text{ if } o = 2 \end{cases}$ and output $(Z_1^k, Z_2^k)$.
```

To find the minimizer of $p(z)$, it is equivalent to reach the zero point of $\nabla p(z)$ by using Newton search. However, the traditional unconstrained Newton search method may fail for open-constrained problem. The first numerical issue comes from the logarithmic computation $\log(z_1 - z_2)$ when z_1 is close to z_2 [33]. To address it, we can use a variable transform as in [33]. Specifically, let $s = c_1 + c_2$, and if $z \to -c_1$, let $Z_1 = c_1 + z$; if $z \to c_2$, let $Z_2 = c_2 - z$. Rather than directly optimizing variable z, we now do optimization on Z_1 or Z_2. Let $o = 1, 2$ denote the routine we choose, $\widehat{h}_{1t}(z) = \frac{a_t}{2} z^2 + b_t z$ and $\widehat{h}_{2t}(z) = \frac{a_t}{2} z^2 - b_t z$, we can uniformly minimize a transformed problem

$$\begin{aligned} &\min_{Z_o} \widehat{p_o}(Z_o) = Z_o \log(Z_o) + (s - Z_o) \log(s - Z_o) \\ &\qquad + 1/q \ln(\sum_{t=1}^{T} \gamma_t \exp(q \widehat{h}_{ot}(Z_o - c_o))) \ s.t \ 0 \leq Z_o \leq s. \end{aligned} \tag{14}$$

In addition, the smoothing function $f(\boldsymbol{\alpha}, q)$ becomes increasingly ill-conditioned when q increases. Then the Newton search may diverge from the desired root. To show this issue, we give a specific example in Fig. 2. From Fig. 2(b), the second derivative has a sharp change around the root and the smooth condition for $q = 100$ becomes bad. Correspondingly,



(a) $\nabla p(z)$

(b) $\nabla^2 p(z)$

(c) $p(z)$

(d) $\lambda_1(z)$

Fig. 2. Example of the ill-conditioned problem when q becomes large. (The example comes from the experiment on news20 dataset for $q = 10$ and $q = 100$ with two subfunctions, where $c_1 = 0.09$, $c_2 = 0.99$, $a_1 = 0.015$, $a_2 = 0.040$, $b_1 = -2.433$, $b_2 = 1.380$, $||\mathbf{w}_1||^2 = 2406.148$, and $||\mathbf{w}_2||^2 = 2403.345$. The dashed-dotted lines in Fig. 2(a) denote the tangent lines at $z = 0.25$ and 0.55, respectively.) (a) $\nabla p(z)$. (b) $\nabla^2 p(z)$. (c) $p(z)$. (d) $\lambda_1(z)$.

the Newton search works well when $q = 10$ but cannot converge when $q = 100$. To solve the possibly ill-conditioned problem, we proposed to use a Newton-Bisection method, which is also known as the safeguarding rooting finding method [12]. Let Z_l and Z_u be the lower bound and upper bound of the transformed variable Z_o, the Newton-Bisection search is performed as in Algorithm 3. If the Newton search overshoot the search range, the Bisection search is used instead.

C. Convergence

Theorem 2: Suppose the sequences $\{\boldsymbol{\alpha}^k\}$ and $\{q_k\}$ are generated by Algorithm 2 with the proposed coordinate descent update, $\{\boldsymbol{\alpha}^k\}$ converges to the global solution of (12).

The proof can be found in Appendix B. In addition, given a monotonically nondecreasing sequence $\{q_k\}$, the proposed smoothing coordinate descent method attains a sublinear convergence rate.

Theorem 3: There exists a k_0 and a $\kappa_{q_k} \in (0, 1)$, for $\forall k \geq k_0$, $f(\boldsymbol{\alpha}^k) - f(\boldsymbol{\alpha}^*) \leq \kappa_{q_k}^{(k-k_0)}(f(\boldsymbol{\alpha}^{k_0}) - f(\boldsymbol{\alpha}^*)) + (2\log(T)/q_k)$.

The proof can be found in Appendix C. From Theorem 3, the error brought by a fixed q is bounded by $((2\log(T)/q) + \epsilon)$, where ϵ is the numerical error [32]. Hence, with a properly large q_k, the smoothing algorithm can achieve a very accurate solution. The following theorem states that Algorithm 1 can globally converge.

Theorem 4: Given that in each iteration of Algorithm 1, the reduced minimax subproblem (9) and the most-active-constraint selection problem can be globally solved, Algorithm 1 can achieve a global solution to (7).

The proof can be adapted from [27].

D. Discussions of Related Work

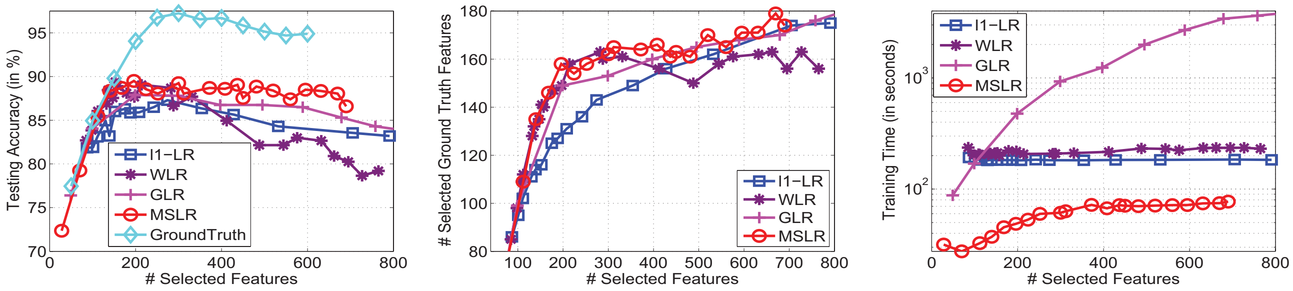
A typical feature selection method that is close to our model is the feature scaling scheme studied in [9] and [10], where a feature scaling vector $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_m]' \geq \mathbf{0}$ is introduced to the dual of SVM to weight the importance of each feature [10]. Specifically, its optimization problem is written as follows:

$$\min_{\boldsymbol{\delta}} \max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k_{\delta}(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i}^{n}\alpha_i,$$
$$\text{s.t. } 0 \leq \alpha_i \leq C, \sum y_i\alpha_i = 0, ||\boldsymbol{\delta}||_p = \delta_0 \qquad (15)$$

where $p \geq 1$, δ_0 is a parameter to control the desired sparsity, $\boldsymbol{\alpha}$ is the vector of SVM dual variables and $k_{\delta}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{m}\delta_k^2 x_i^k x_j^k$ is a weighted linear kernel [10, eq. (5.18)]. In [9], an alternated optimization scheme was proposed to solve (15). However, it is inefficient for tackling high-dimensional problems. The nonsmooth subproblem (10) can also be solved through subgradient methods [21], [24]. In addition, some other methods are also available, such as the sequential quadratic programming (SQP) method [23] and proximal gradient methods [22], [30]. However, SQP has much higher running complexity per iteration [23] and the proximal gradient methods have difficulties of the line search w.r.t. $\boldsymbol{\alpha}$ on $(0, C)^n$. On the contrary, the line search of the proposed smoothing coordinate descent method can be easily solved through Algorithm 3.

E. Computational Complexity

The liblinear ℓ_1 solver has linear convergence rate and scales $O(nm)$ [35]. The proposed smoothing coordinate



(a) Prediction accuracy (b) # recovered ground-truth features (c) Training time

Fig. 3. Experimental results with number of selected features on $\mathbf{X} \in \mathbb{R}^{5000\times 100\,000}$ with 300 ground-truth features. (a) Prediction accuracy. (b) No. of recovered ground-truth features. (c) Training time.

descent attains a linear convergence rate. Therefore, it converges to the optimum within finite iterations. In summary, the training complexity of the proposed MSLR includes two parts: the smoothing coordinate descent which takes $O(nrT_m)$ and the most-active-constraint selection which takes $O(nmT_m)$ complexity, where T_m is the maximum number of the cutting plane iterations. To compare, MSLR can be more efficient than ℓ_1 solvers on high-dimensional dense datasets since it only needs to take T_m times on the most-active-constraint selection, while ℓ_1 solver may need many times to converge. In FGM, the subgradient method (namely, SimpleMKL) is used to solve the minimax problem, which is very computationally expensive for large-scale datasets.

IV. Experiments

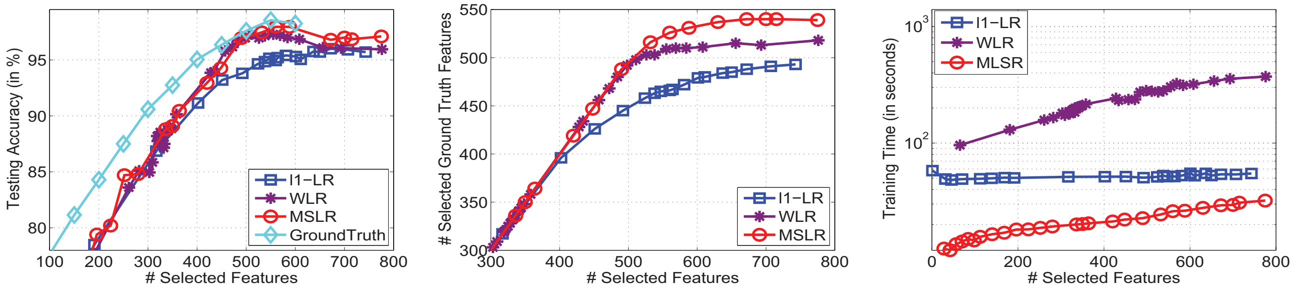
We evaluate the performance of MSLR on several synthetic and real-world datasets. The ℓ_1-LR (ℓ_1-LR) [35], weighted ℓ_1-logistic regression (WLR) [4], [37], greedy LR (GLR) [18], and FGM [27] are adopted as baseline methods. For simplicity, we denote by FGM-LR and FGM-SVM for FGM with SimpleMKL solver regarding LR and SVM, respectively [1]. From [35], LIBlinear solver for ℓ_1-LR provides the state-of-the-art performance [2]. We also adapt it to implement WLR [4], [37]. In the experiments, we vary the parameter C for ℓ_1-LR and WLR to achieve different sparsity. For MSLR, we set $\beta = 5/3$ and $q_0 = 1/100$ for the smoothing algorithm. In addition, we fix $C = 10$ and $T_m = 10$, and vary r from 2 to 40 to select different number of features. All the methods are written in C++ and all experiments are conducted on Intel Core i7 CPU (2.80 GHz) with 64-b operating system.

A. Synthetic Experiment

In the first synthetic experiment, we generated a $5000 \times 100\,000$ Gaussian random matrix as $\mathbf{X}$. To generate the ground-truth informative features, we generated a sparse vector $\mathbf{w}$ with 300 nonzero entries sampled from Gaussian distribution $\mathcal{N}(0, 2)$, where each nonzero $|w_i|$ denoted an informative feature. Notice that in this example, the number of instances is much smaller than the number of features, which happens in many feature selection tasks, such as gene selection [11] and effective SNP detection (in our experiments). Finally, we produced the output by $\mathbf{y} = \text{sign}(\mathbf{Xw})$. Similarly, we generated the testing dataset $\mathbf{X}_{\text{test}}$ and $\mathbf{y}_{\text{test}} = \text{sign}(\mathbf{X}_{\text{test}}\mathbf{w})$. The number of testing points is set to 2000. We only study the performance of ℓ_1-LR, WLR, GLR, and MSLR. For fair comparisons, after obtaining the feature subset, we did retraining with the selected features using standard linear LR with $C = 5$. For ℓ_1-LR and WLR, we change the regularization parameter $C \in [0.001, 4]$ to select different number of features. For MSLR, with fixed C, we can select different number features by changing r or T_m.

The prediction accuracy, the number of recovered ground-truth features w.r.t. the number of selected features and the training time are recorded in Fig. 3. We also reported the accuracy obtained by LR with the k most informative ground-truth features, denoted by GroundTruth. From Fig. 3(a) and (b), MSLR achieves the best prediction accuracy and recovers the largest number of ground-truth features with more than 300 selected features. In other words, MSLR can be more effective to select the predictive and informative features. Particularly, MSLR shows much better performance than ℓ_1-LR on a wide range of selected features. Two reasons cause this. Firstly, in MSLR, the regularizer can help to avoid the over-fitting problem. Secondly, a suitable C can be set to reduce the bias. For ℓ_1-LR, both the number of selected features and model complexity are controlled by C, where a larger C can reduce the risk of bias, but will lead to more features; a smaller C can select fewer features but may produce bias. Therefore, it is hard to trade off between two parts. As expected, WLR, which is designed to alleviate the bias problem, shows better results than ℓ_1-LR when the number of selected features is smaller than 300. GLR also shows consistently better performance than LR. However, both of them show poorer performance than MSLR. For GLR, since there is no regularizer, the bias problem can be alleviated to some extent, but the over-fitting problem may deteriorate the performance. Another interesting observation is that, all methods, including the GroundTruth, show decreasing performance trend when the number of selected features is over 400, which verifies the importance of the feature selection. On the issue of training efficiency, from Fig. 3(c), MSLR shows better training efficiency than others. Typically, GLR is the most expensive one when the number of selected features is larger than 100.

In the second experiment, we generated another dataset with relatively large number of instances and smaller dimensions, namely $\mathbf{X} \in \mathbb{R}^{20\,000\times 10\,000}$. In addition, we generated a sparse vector $\mathbf{w}$ of 600 nonzero entries sampled from $\mathcal{N}(0, 10)$ as the



(a) Prediction accuracy (b) # recovered ground-truth features (c) Training time

Fig. 4. Experimental results with number of selected features on $\mathbf{X} \in \mathbb{R}^{20\,000\times 10\,000}$ with 600 ground-truth features. (a) Prediction accuracy. (b) No. of recovered ground-truth features. (c) Training time.

TABLE I

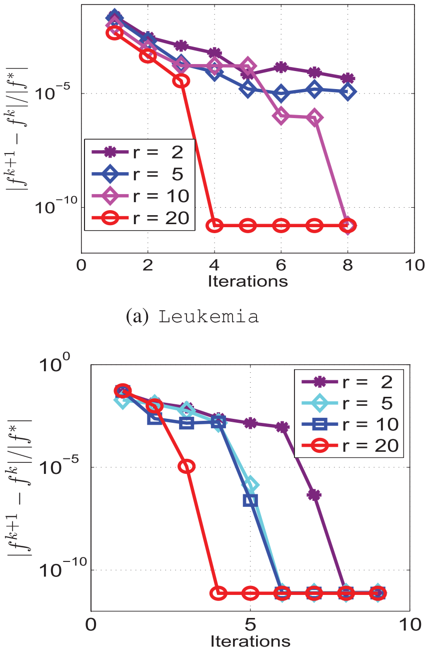
Datasets Used in the Experiments

Datasets	# Features	# Training	# Testing	Data Type
Leukemia	7,029	38	34	dense
Arxiv astro-ph	99,757	62,369	32,487	sparse
news20	1,355,191	9,996	10,000	sparse
RCV1	47,236	20,242	677,399	sparse
real-sim	20,958	72,309	-	sparse
mil-image	10,800	1,200	800	dense
SNP	393,321	2,060	2,841	dense
epsilon	2,000	400,000	100,00	dense

ground-truth. With this generation, most of the features with nonzero w_i are significant to the output $\mathbf{y}$. The experimental results are shown in Fig. 4, where the GLR is not considered due to its high computational cost. We can observe that MSLR generally outperforms the two competitors and obtains the closest results to GroundTruth. More importantly, with more training examples, all the methods can select more ground-truth features, but MSLR selects the most among them and obtains the best prediction accuracy when the number of selected features is more than 400.

B. Real-World Dataset Experiments

The real-world datasets used in our experiments are listed in Table I. Among them, Leukemia, news20.binary, RCV1, and real-sim can be downloaded from [2], where Leukemia is a microarray gene dataset. Arxiv astro-ph can be downloaded from [1]. The last three datasets are with dense features. The mil-image is with medium dimensions from [38] for multi-instance learning. The original data package contains images of five scenes and only the first scene is studied in our experiments. We also apply MSLR on SNP detection. The SNP dataset collected in our experiment contains two groups with 2938 controls and 1963 cases for Autoimmune Disease study. In these experiments, we manually split mil-image and SNP into training set and testing set, respectively. Finally, epsilon is from [2] with a very large number of training instances.



(a) Leukemia

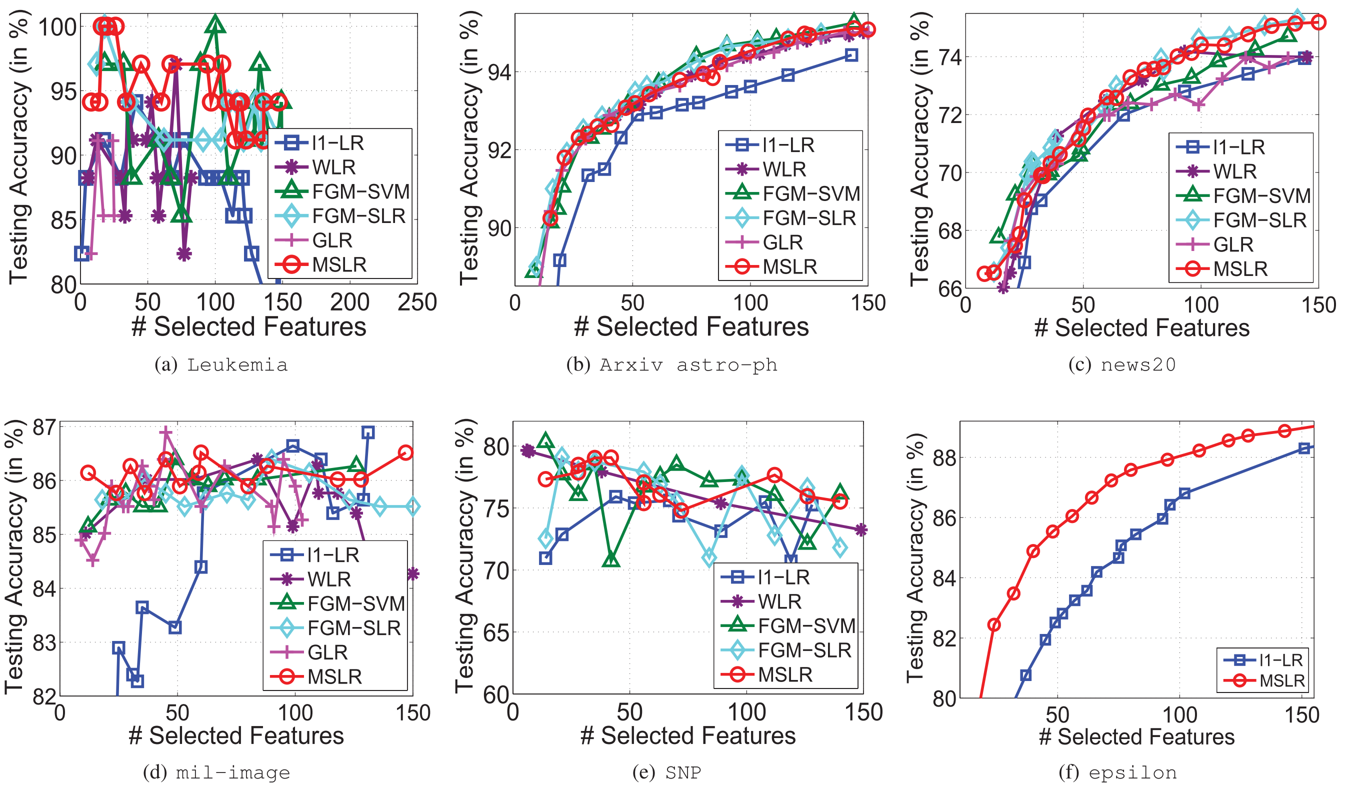
(b) news20

Fig. 5. Convergence of MSLR on (a) Leukemia and (b) news20 dataset.

In the first experiment, we study the convergence behavior of MSLR. We recorded the relative function value difference $|f^{k+1} - f^k|/|f^*|$ w.r.t. outer iterations for Algorithm 1 in Fig. 5 on Leukemia and news20 datasets, where f^* is the global function value. On both datasets, we can observe that the relative function value difference decreases sharply within several iterations, which demonstrates the nice convergence behavior of MSLR.

In the second experiment, we study the effectiveness and efficiency of various methods on real-world datasets. Similar in the synthetic experiments, we reported the prediction accuracy obtained by retraining with linear LR. Specifically, the prediction accuracy and the training time versus the number of selected features are recorded in Figs. 6 and 7, respectively. The recorded time is in logarithm scale. From Fig. 6, in general, MSLR can obtain better prediction accuracy than ℓ_1-LR and slightly better than WLR and GLR with the same number of selected features. This observation demonstrates that MSLR can select more informative features. Particularly, MSLR can obtain 100% prediction accuracy with eight features on Leukemia dataset, which is very useful for medical diagnosis.

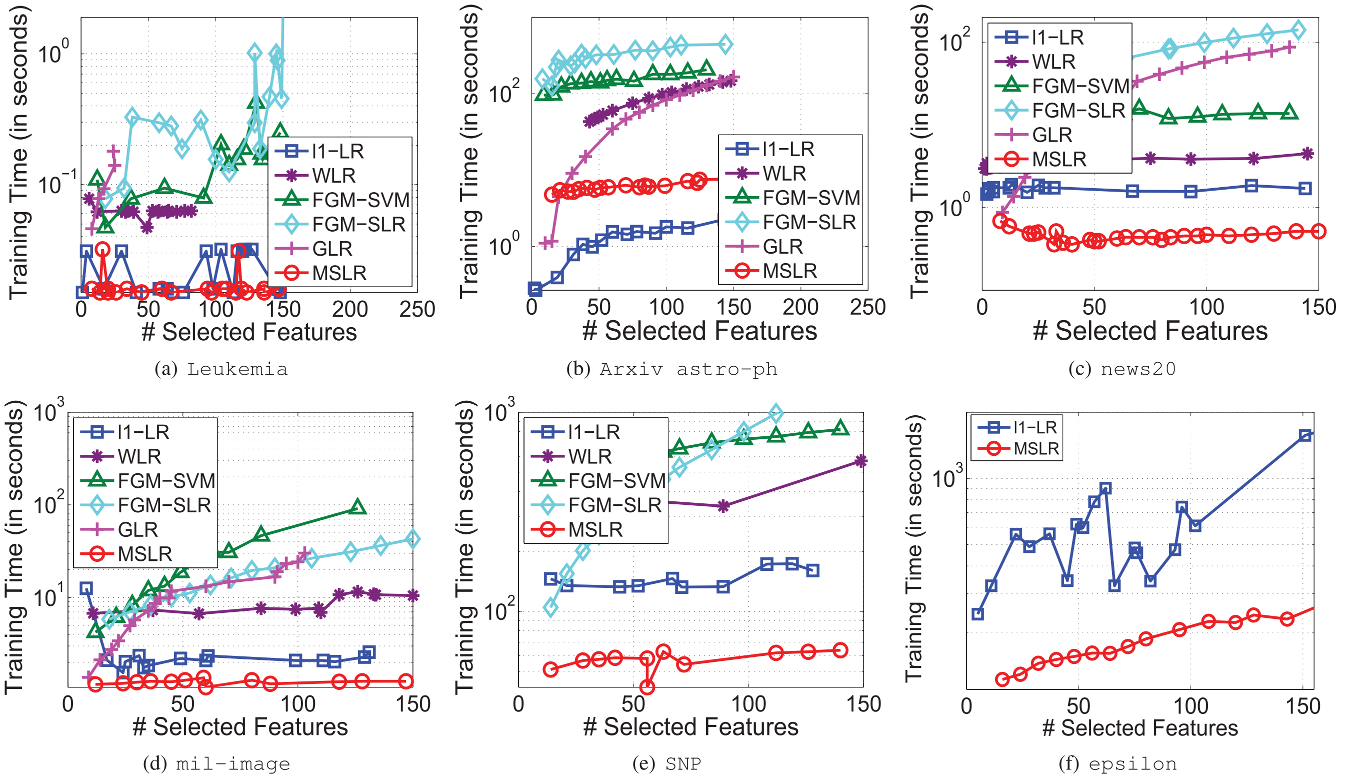
Now we come to the efficiency comparison. From Fig. 7, ℓ_1-LR method shows the best efficiency on Arxiv astro-ph,



(a) Leukemia (b) Arxiv astro-ph (c) news20

(d) mil-image (e) SNP (f) epsilon

Fig. 6. Testing accuracy on various datasets with different number of features. (a) Leukemia. (b) Arxiv astro-ph. (c) news20. (d) mil-image. (e) SNP. (f) epsilon.

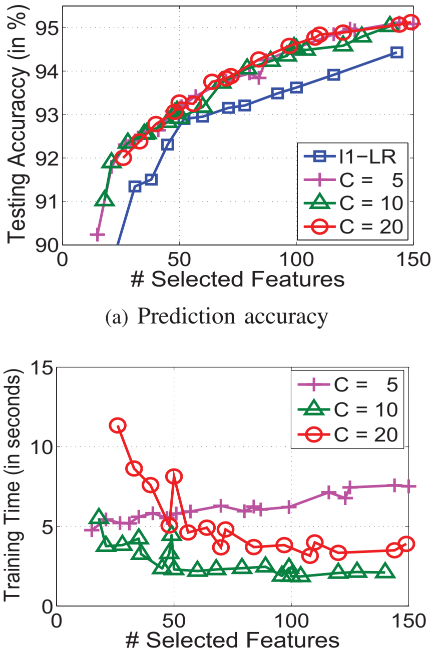


(a) Leukemia (b) Arxiv astro-ph (c) news20

(d) mil-image (e) SNP (f) epsilon

Fig. 7. Training time on various datasets (in logarithm scale) with different number of features. (a) Leukemia. (b) Arxiv astro-ph. (c) news20. (d) mil-image. (e) SNP. (f) epsilon.

and competitive efficiency on news20.binary dataset. Notice that these two datasets are with very sparse entries. Therefore, the liblinear solver for ℓ_1-LR gains very fast training speed. However, MSLR can obtain the best training efficiency among all methods on mil-image, epsilon and SNP datasets with competitive or better prediction accuracy. MSLR is also much



(a) Prediction accuracy

(b) Training time

Fig. 8. Sensitivity of MLSR to parameter C. (a) Prediction accuracy. (b) Training time.

faster than WLR and GLR in most cases, which verifies its good scalability. Particularly, MSLR can be much faster on the extremely large-scale epsilon dataset, where the results of WLR, FGM-SVM, FGM-LR, and GLR are not reported due to their high computational cost. Actually, in MSLR, it only needs to scan all the features at most T times but solves much reduced subproblems of complexity $O(nTB)$. Therefore, MSLR is more efficient with extremely large number of features. Because of the same reason, MSLR can be more efficient on dense datasets. Although MSLR, FGM-SVM and FGM-LR obtain very close prediction accuracy under the same sparsity, MSLR is much more efficient than them. To be more specific, MSLR can be 10–500 times faster than FGM-SVM and FGM-LR. In addition, the training time of MSLR is much less sensitive to the increment of the parameter r. For example, when r becomes relatively large ($r > 30$), MSLR is surely several hundred times faster than FGM-SVM and FGM-LR.

Finally, the number of selected features for MSLR can be easily controlled by r. In addition, we can also fix C and r, and change T_m to obtain different number of features. By comparison, it is much more difficult for ℓ_1-LR and WLR to obtain the desired number of features by tuning C. For example, in our experiments, on Leukemia dataset, we use $C = 0.08$ to obtain four features and $C = 70$ to obtain 101 features; while for WLR, we use $C = 50$ to obtain five features but have to set $C = 20\,000\,000$ to obtain 95 features. From the above experiment, the performance of ℓ_1-LR is sensitive to the parameter C.

In the third experiment, we conducted the sensitivity study on the parameter C for MSLR. Fig. 8 recorded the prediction accuracy and training time of MSLR on Arxiv astro-ph dataset with different C. ℓ_1-LR is adopted as the baseline. We chose different C from {5, 10, 20}. From the figures, MSLR is relatively insensitive to the regularization parameter in terms of prediction accuracy and training time.

TABLE II

PREDICTION ACCURACY ON RCV1 (IN %)

# Features	65	172	202	264	332	381
L1-LR	88.84 ±0.12	92.79 ±0.1	93.45 ±0.08	94.33 ±0.06	94.82 ±0.07	95.17 ±0.04
MSLR	91.11 ±0.09	94.42 ±0.06	94.86 ±0.04	95.41 ±0.04	95.87 ±0.06	96.09 ±0.03
Wilcoxon	1	1	1	1	1	1
P-value	1.82E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4

TABLE III

PREDICTION ACCURACY ON REAL-SIM (IN %)

# Features	101	159	195	242	266	318
L1-LR	89.42 ±0.18	91.68 ±0.17	92.29 ±0.12	92.89 ±0.14	93.26 ±0.13	93.65 ±0.11
MSLR	91.38 ±0.27	92.65 ±0.16	93.16 ±0.19	93.74 ±0.20	94.12 ±0.10	94.54 ±0.09
Wilcoxon	1	1	1	1	1	1
P-value	1.81E-4	1.83E-4	1.83E-4	1.83E-4	1.82E-4	1.81E-4

Finally, we conducted an additional experiment to compare the performance of MSLR and ℓ_1-LR on RCV1 and real-sim datasets. We did not include the results of WLR and GLR because of their high computational cost. We use the averaged cross-validation accuracies as the comparison criterion. Specifically, we randomly partitioned the datasets into ten folds, and used seven folds as the training set and the rest as the testing set. We independently repeated the procedure for ten times and recorded the average testing accuracies and the standard variations. For RCV1, since it was partitioned into training and testing set, we merged them to form a larger dataset of 697 641 instances. We vary C for ℓ_1-LR to obtain different number of features (denoted by #) and then set $r = \#/T_m$ for MSLR such that they obtain similar number of features. To show the statistic difference between two methods, we conducted the Wilcoxon test on the averaged testing accuracies with 5% significance level. The averaged testing accuracies as well as the p-values are listed in Tables II and III, where 1 shows the significant difference under the Wilcoxon test. From the two tables, we can observe that MSLR indeed significantly outperforms ℓ_1-LR with the same number of selected features.

V. CONCLUSION

In this paper, we proposed a new weight scaling scheme to achieve the sparse LR by introducing a scaling vector $\mathbf{d} \in [0, 1]^m$ into the LR model. To induce the sparsity, we imposed an ℓ_1-constraint $||\mathbf{d}||_1 \leq r$ on $\mathbf{d}$, where r was a conservative estimation to the desired number of features. We further transformed the model as a SIP problem and then solved it through an efficient cutting plane method. To solve the resultant nonsmooth minimax subproblem of the cutting plane method, we presented a smoothing coordinate descent algorithm that could attain a weakly linear convergence rate. In each iteration of the cutting plane algorithm, at most r features were selected. Therefore, the number of selected features can be controlled by changing the maximum number of cutting plane iterations T_m or r. Extensive experiments on several synthetic and real-world datasets verified the efficiency and effectiveness of the proposed method.

In this paper, we studied only the problem of high-dimensional linear feature selections. However, in many real-world problems, the features might have complex nonlinear structures like the Corral (or XOR) problem [34]. The detection of these nonlinear features of complex structures was desirable for many applications. However, it was still a challenging problem and cannot be handled by the proposed method of this paper. To address the nonlinear feature selection problem, one possible way was to explicitly map the nonlinear features to high-dimensional linear feature space using nonlinear feature mappings. Then the nonlinear feature selection problem can be transformed as a linear feature selection problem in the feature space [5]. For example, we can address the nonlinear feature selection of the XOR Corral problem by employing the polynomial kernel feature mappings [5]. We leave this nonlinear feature selection issue as our future study.

Appendix A

Proof of Proposition 2

The smoothing algorithm presented in [32] was originally proposed to solve unconstrained problems. To prove that it can be applied to solve the minimax problem (6), principally, we should verify that $f_t(\boldsymbol{\alpha}) = 1/2\boldsymbol{\alpha}'\mathbf{Q}^t\boldsymbol{\alpha} + G(\boldsymbol{\alpha})$ satisfies the necessary conditions mentioned in [32] and Algorithm 2 was well defined in the open domain $(0, C)^n$. The proof includes several Propositions and is concluded in Lemma 2. To begin with, we first present some basic properties of $f_t(\boldsymbol{\alpha})$ and the smoothing problem $f(\boldsymbol{\alpha})$ in (11).

Definition 1: Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable and let σ be a positive scalar. If f satisfies the condition, $(\nabla f(\mathbf{x}) - \nabla f(\mathbf{z}))'(\mathbf{x} - \mathbf{z}) \geq \sigma||\mathbf{x} - \mathbf{z}||^2, \ \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^n$, then f is strongly convex.

Proposition 4:

1) $f_t(\boldsymbol{\alpha})$ is strongly convex with $\sigma \geq \frac{4}{C}$ and twice continuous differentiable function within an open domain $\boldsymbol{\alpha} \in (0, C)^n$.
2) $f_t(\boldsymbol{\alpha})$ attains a minimum $\boldsymbol{\alpha}^*$ within $(0, C)^n$.
3) For any subset $\mathcal{S} \subset (0, C)^n$, there exists a $K_t \in (0, +\infty)$, such that $\max\{|f_t(\boldsymbol{\alpha})|, ||\nabla f_t(\boldsymbol{\alpha})||, ||\nabla f_t^2(\boldsymbol{\alpha})||\} \leq K_t$ for $\forall \boldsymbol{\alpha} \in \mathcal{S}$.
4) For any $\mathbf{x} \in \mathcal{S}$, $\mathbf{x}'\nabla^2 f_t(\boldsymbol{\alpha})\mathbf{x} \leq K_t||\mathbf{x}||^2$.

Proof:

1) It is easy to verify that $f_t(\boldsymbol{\alpha})$ is convex. In addition, we can verify that $\nabla f_t(\boldsymbol{\alpha}) = \mathbf{Q}^t\boldsymbol{\alpha} + \widehat{\mathbf{p}}$ and $\nabla^2 f_t(\boldsymbol{\alpha}) = \mathbf{Q}^t + \widehat{\mathbf{Q}}$, where $\widehat{\mathbf{p}} = [\log(\alpha_1/(C - \alpha_1)), \ldots, \log(\alpha_n/(C - \alpha_n))]$ and $\widehat{\mathbf{Q}} = \text{diag}[(1/\alpha_1) + (1/(C - \alpha_1)), \ldots, (1/\alpha_n) + (1/(C - \alpha_n))]$. To prove $f_t(\boldsymbol{\alpha})$ is strongly convex, we only need to prove that $\nabla^2 f_t(\boldsymbol{\alpha}) - \sigma\mathbf{I} \succeq 0$, where $\mathbf{I}$ is the identity matrix. Because $\mathbf{Q}^t_{ii} = (1/\alpha_i) + (1/(C - \alpha_i)) \geq 4/C$, with $\mathbf{Q}^t \succeq 0$, we know $f_t(\boldsymbol{\alpha})$ is strongly convex and $\sigma \geq 4/C$.
2) From [33, Th. 1], $f_t(\boldsymbol{\alpha})$ attains a minimum $\boldsymbol{\alpha}^*$ within $(0, C)^n$.
3) Firstly, $|\frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}^t\boldsymbol{\alpha}|, ||\mathbf{Q}^t\boldsymbol{\alpha}||$ and $||\mathbf{Q}^t||$ are bounded for $\forall \boldsymbol{\alpha} \in S$. $\boldsymbol{\alpha} \in \mathcal{S}$ means that any dimension of $\boldsymbol{\alpha}$ cannot reach to 0 and C. For $G(\alpha)$, because it is a convex function, its maximum value should lie on the boundary. In addition, with $\lim\limits_{\alpha \rightarrow 0} \alpha \log(\alpha) = 0$, we know $f_t(\boldsymbol{\alpha})$ is upper bounded within $\mathcal{S}$. Also $f_t(\boldsymbol{\alpha})$ attains a optimal solution in $(0, C)^n$. Then we can conclude $|f_t(\boldsymbol{\alpha})|$ is bounded within $\mathcal{S}$. Obviously, we also have $||\hat{\mathbf{p}}|| < +\infty$ and $||\widehat{\mathbf{Q}}|| < +\infty$. Note that $||\widehat{\mathbf{p}}|| = \infty$ and $||\widehat{\mathbf{Q}}|| = +\infty$ only when there is at least one dimension of $\boldsymbol{\alpha}$ lies on the boundary, which, however, will never happen because $\boldsymbol{\alpha} \in (0, C)^n$. Finally, we can obtain that for any subset $\mathcal{S} \subset (0, C)^n$, there exists a $K_t \in (0, +\infty)$, such that $\max\{|f_t(\boldsymbol{\alpha})|, ||\nabla f_t(\boldsymbol{\alpha})||, ||\nabla f_t^2(\boldsymbol{\alpha})||\} \leq K_t$ holds for $\forall \boldsymbol{\alpha} \in \mathcal{S}$.
4) For $\forall \mathbf{x} \in \mathcal{S}$, We have $\mathbf{x}'\nabla^2 f_t(\boldsymbol{\alpha})\mathbf{x} \leq ||\mathbf{x}||^2||\nabla^2 f_t(\boldsymbol{\alpha})|| \leq K_t||\mathbf{x}||^2$. This completes the proof. ■

Now we present the corresponding results regarding the smoothing function $f(\boldsymbol{\alpha}, q)$.

Proposition 5: Let $K = \max_{t=1}^T K_t$ and $L = K + 2K^2$, there exist constant $\tau_1 = \frac{4}{C}$ and $\tau_2 = qL$, such that $\tau_1||\mathbf{x}||^2 \leq \mathbf{x}'\nabla^2_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, q)\mathbf{x} \leq \tau_2||\mathbf{x}||^2$, for all $\boldsymbol{\alpha} \in \mathcal{S}$, $\mathbf{x} \in (0, 1)^n$ and $q > 1$.

Proof: We first prove the left side. For the first term of the Hessian matrix $\nabla^2_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, q)$, we have $\mathbf{x}'\big(\sum_{t=1}^T \widehat{\lambda}_t \nabla^2 f_t(\boldsymbol{\alpha})\big)\mathbf{x} = \sum_{t=1}^T \widehat{\lambda}_t \mathbf{x}'\big(\nabla^2 f_t(\boldsymbol{\alpha})\big)\mathbf{x} \geq \sum_{t=1}^T \widehat{\lambda}_t \frac{4}{C}||\mathbf{x}||^2 = \frac{4}{C}||\mathbf{x}||^2$ using Proposition 4. From [32, Lemma 2.1], $\sum_{t=1}^T \big(q\widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha})\nabla f_t(\boldsymbol{\alpha})'\big) - q\big(\sum_{t=1}^T \widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha})\big)\big(\sum_{t=1}^T \widehat{\lambda}_t \nabla f_t(\boldsymbol{\alpha})\big)' \succeq 0$. Accordingly, we can set $\tau_1 = \frac{4}{C}$. For the right side, with $K = \max_{t=1}^T K_t$, by adapting the [23, Lemma 3.2] and Proposition 4, we have $\mathbf{x}'\nabla^2_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, q)\mathbf{x} \leq qL||\mathbf{x}||^2$, where $L = K + 2K^2$. Hence we can set $\tau_2 = qL$. ■

Lemma 2: Based on Proposition (2), Algorithm 2 is well defined given that $\{\boldsymbol{\alpha}^k\}$ is always in $(0, C)^n$.

Proof: Let $\boldsymbol{\alpha}^*_q$ be the global optimum of $f(\boldsymbol{\alpha}, q)$ regarding q, from Proposition (2), $\nabla^2_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}, q) \succ 0$, hence for any $\boldsymbol{\alpha}^k \in (0, C)^n$, we have the Taylor expansion as follows: $f(\boldsymbol{\alpha}^*_q, q) = f(\boldsymbol{\alpha}^k, q) + \nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^k, q)'(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^k) + \frac{1}{2}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^k)'\nabla^2_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^k + \theta(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^k), q)(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^k)$, where $\theta \in (0, 1)$. If $\boldsymbol{\alpha}^*_q$ is the global optimum of $f(\boldsymbol{\alpha}, q)$, then $f(\boldsymbol{\alpha}^*_q, q) \leq f(\boldsymbol{\alpha}^k, q)$. Then we have $\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^k, q)'(\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^k) < 0$, which indicates that there exists at least a descent direction $\mathbf{h}^k$ at $\boldsymbol{\alpha}^k$ regarding q. Furthermore, regarding $\mathbf{h}^k$, we can always find a suitable step size s to make some decrease of $f(\boldsymbol{\alpha}, q)$ by using line search method. ■

Appendix B

Proof of Theorem 2

To prove the presented algorithm can solve the minimax problem, based on the above global optimality condition, we should prove that the sequence $\{\boldsymbol{\alpha}^k\}$ generated by the algorithm follows: 1) $\{\boldsymbol{\alpha}^k\} \in (0, C)^n$ for any $k > 0$, which will be stated in Lemma 3 and 2) $\{\boldsymbol{\alpha}^k\}$ has a limit point $\widehat{\boldsymbol{\alpha}} \in (0, C)^n$ that is the stationary point of the minimax problem, which will be summarized in Theorem 5. To begin with, we first present the optimality condition for $f(\boldsymbol{\alpha})$, which is shown as follows.

Proposition 6 [23], [32]:

1) Given $f_t(\boldsymbol{\alpha})$ defined by $\mathbf{d}^t$ with $t = 1, \ldots, T$, $\boldsymbol{\alpha}^*$ is stationary point to the minimax problem in (6), if there exists a vector $\boldsymbol{\zeta}^* = (\zeta_1^*, \ldots, \zeta_T^*)$ such that $\sum_{t=1}^{T} \zeta_t^* \nabla f_t(\boldsymbol{\alpha}^*) = 0, \zeta_t^* \geq 0, \sum_{t=1}^{T} \zeta_t^* = 1, \zeta_t^* = 0$, if $f_t(\boldsymbol{\alpha}^*) < \max\{f_1^*(\boldsymbol{\alpha}^*), \ldots, f_t^*(\boldsymbol{\alpha}^*)\}$.
2) If $\boldsymbol{\alpha}^*$ is a local minimum to (6), then $\boldsymbol{\alpha}^*$ is a stationary point satisfying (6).

Conversely, if $f(\boldsymbol{\alpha})$ is convex, then $\boldsymbol{\alpha}^*$ is a global minimum to (6) if $\boldsymbol{\alpha}^*$ is a stationary point. ■

Lemma 3: Suppose $\{\boldsymbol{\alpha}^k\}$ be the sequence of the smoothing dual coordinate descent for the Step 1 in Algorithm 2, then $\{\boldsymbol{\alpha}^k\} \in (0, C)^n$ for any $k > 0$.

Proof: When fixing q, we solve the problem by using coordinate descent algorithm such that $\boldsymbol{\alpha} \in (0, C)^n$. In addition, it holds for $\forall q > 0$. Then the lemma holds. ■

We further need the following Lemmas and Propositions.

Lemma 4: Let $\boldsymbol{\alpha}^k$ be the sequence generated by the outer iteration of Algorithm 2, then: 1) the sequence $f(\boldsymbol{\alpha}^k, q_k)$ is monotonically decreasing; 2) the sequence $f(\boldsymbol{\alpha}^{k+1}, q_k)$ and $f(\boldsymbol{\alpha}^k, q_k)$ are both convergent and have the same limit; and 3) the sequence $\boldsymbol{\alpha}^k$ is bounded.

Proof: The proof can be easily obtained by adapting the proof of [32, Proposition 3.1]. ■

Proposition 7: Regarding $p(z)$, we have: 1) $p(z)$ is convex and twice continuous differentiable; and 2) its derivative is Lipschitz continuous. That is, $\exists L > 0$ such that

$$|\nabla p(x) - \nabla p(y)| \leq L|x - y| \quad \forall x, y. \tag{16}$$

The following lemma measures the improvement of the exact line search regarding a fixed q.

Lemma 5: With the the exact line search method, $f(\boldsymbol{\alpha}_i^k, q_k) - f(\boldsymbol{\alpha}_i^{k+1}, q_k) \geq (\nabla p(0)^2)/(2L_i^{q_k})$, where $L_i^{q_k}$ is the Lipschitz constant regarding q_k for the ith variable.

Proof: Without loss of generality, we drop the superscript q_k for $L_i^{q_k}$. In the coordinate descent update, suppose the cyclic rule is used. Hence, there are two iterations for the smoothing algorithm. Specifically, we denote by k for outer iteration and i for the ith variable $\alpha^{k,i}$ of the inner iteration to be optimized, where $i = 1, \ldots, n$.

For the ith dimension of the kth iteration, we use an exact line search to find the exact solution with an accuracy ε to the 1-D problem. It is hard to measure the improvement of the objective function with such 1-D update. Alternatively, we consider another line search method which gives a lower bound to the improvement. For the ith dimension in the kth iteration, with a fixed $v_{k,i}$ where $v_{k,i}$ is the update direction, we consider to use an exact line search to find the exact step size. Let $f(\boldsymbol{\alpha}_i^k, q_k)$ be the function regarding α_i and $f(\boldsymbol{\alpha}_i^{k+1}, q_k)$ be the optimal value obtained by such line search. Similar to the 1-D search in Section III-B, we can do the optimization on $p(z)$ from $z = 0$ rather than on f. Let s_k be the optimal step size, we have $\nabla p(0 + s_k v_{k,i}) v_{k,i} = 0$. From Proposition 7, we have $s_k L_i v_{k,i}^2 \geq |\nabla p(0 + s_k v_{k,i}) - \nabla p(0)| \times |v_{k,i}| = (\nabla p(0 + s_k v_{k,i}) - \nabla p(0)) \times v_{k,i} = -\nabla p(0) v_{k,i}$. Then, we have $s_k \geq -(\nabla p(0)/L_i v_{k,i})$. Now we measure the improvement of the above line search method. Let $\boldsymbol{\Omega}_s = \{z | p(z) \leq p(0)\}$. Obviously, $\boldsymbol{\Omega}_s$ is bounded. Then, $\forall s$ such that $z = 0 + s v_{k,i} \in \boldsymbol{\Omega}_s$, we have $|p(t s v_{k,i}) - p(0)| \leq t s L_i |v_{k,i}|$ with $t \in [0, 1]$. From mean value theorem, we further have $p(s v_{k,i}) - p(0) = s \int_0^1 \nabla p(t s v_{k,i}) v_{k,i} dt \leq s \nabla p(0) v_{k,i} + \frac{1}{2} s^2 L_i |v_{k,i}|^2$. Hence we have: $f(\boldsymbol{\alpha}_i^{k+1}, q_k) - f(\boldsymbol{\alpha}_i^k, q_k) \leq p(s_k v_{k,i}) - p(0) \leq s_k \nabla p(0) v_{k,i} + (1/2) s_k^2 L_i v_{k,i}^2 = -(\nabla p(0)^2/(2L_i))$. Obviously, with updated gradient in the line search, we can obtain better solution than the above search method. That is to say, $f(\boldsymbol{\alpha}_i^k, q_k) \geq f(\boldsymbol{\alpha}_i^{k+1}, q_k) + (\nabla p(0)^2)/(2L_i)$ still holds. This completes the proof. ■

Theorem 5: Suppose the sequences $\{\boldsymbol{\alpha}^k\}$ and $\{q^k\}$ are generated by Algorithm 2: 1) $\{\boldsymbol{\alpha}^k\}$ has a limit point $\widehat{\boldsymbol{\alpha}}$; 2) Every limit point is a stationary point of the minimax problem; and 3) $\widehat{\boldsymbol{\alpha}}$ is the solution to (12).

Proof:

1) By Lemma 4, the sequence $\{\boldsymbol{\alpha}^k\}$ is bounded. Then a limit point exists.
2) Let $\nabla p_i(0) = \nabla f(\boldsymbol{\alpha}^k, q^k)_i$, i.e., $\nabla p_i(0)$ is the gradient for ith dimension of $\boldsymbol{\alpha}$. In addition, let $\boldsymbol{\alpha}^*$ be the limit point. When $k \to +\infty$, $\nabla f(\boldsymbol{\alpha}^*, +\infty)_i = \nabla p_i(0) \to 0, i = 1, \ldots, n$, which can be proved by contradiction. Suppose that there exists an $\epsilon > 0$ such that $\nabla p_i(0) \geq \epsilon$, for $\forall k > k_0 > 1$. From Lemma 5, $f(\boldsymbol{\alpha}^{k+1}, q^k)$ cannot converge to $f(\boldsymbol{\alpha}^k, q^k)$ as $f(\boldsymbol{\alpha}_i^k, q^k) - f(\boldsymbol{\alpha}_i^{k+1}, q^k) \geq \frac{\nabla p_i(0)^2}{2L_i} \geq \frac{\epsilon^2}{2L_i}$, which contradicts the results in Lemma 4. Hence, $\nabla f(\boldsymbol{\alpha}^*, \infty)_i = 0$ for $i = 1, \ldots, n$. Finally, we obtain $\nabla f(\boldsymbol{\alpha}^*, \infty) = \mathbf{0}$. Then $\boldsymbol{\alpha}^*$ is a stationary point of the minimax problem.
3) With Proposition 6, $\boldsymbol{\alpha}^*$ is the global solution. Then we can complete the proof. ■

Appendix C
Proof of Theorem 3

Proposition 8: With τ_1 and τ_2 defined in Proposition 5, we have $\left(\nabla_{\mathbf{x}} f(\mathbf{x}, q) - \nabla_{\mathbf{y}} f(\mathbf{y}, q)\right)' (\mathbf{x} - \mathbf{y}) \geq \tau_1 \|\mathbf{x} - \mathbf{y}\|^2$ $\|\nabla_{\mathbf{x}} f(\mathbf{x}, q) - \nabla_{\mathbf{y}} f(\mathbf{y}, q)\| \leq \tau_2 \|\mathbf{x} - \mathbf{y}\|$.

Proof: With Proposition 5, the above result holds with $\tau_1 = \frac{4}{C}$ and $\tau_2 = qL$. ■

The following theorem shows that with fixed q, the coordinate descent updating attains linear convergence rate.

Lemma 6: Given $q > 0$, the dual coordinate descent method can attain at least linear convergence rate to the global optimum. In other words, let $\{\boldsymbol{\alpha}_q^k\}$ be the sequence of the dual coordinate descent algorithm and $\boldsymbol{\alpha}_q^*$ be the global solution regarding q, there exist $\kappa_q = (1 - 1/\eta_q) \in (0, 1)$ and an iteration k_0 such that $\forall k \geq k_0$, $f(\boldsymbol{\alpha}_q^{k+1}, q) - f(\boldsymbol{\alpha}_q^*, q) \leq \kappa_q (f(\boldsymbol{\alpha}_q^k, q) - f(\boldsymbol{\alpha}_q^*, q))$, where η is a constant.

Proof: We consider the following bounded problem:

$$\min_{\boldsymbol{\alpha}} \quad g(\mathbf{E}\boldsymbol{\alpha}) + \mathbf{b}'\boldsymbol{\alpha} \quad : \quad L_i \leq \alpha_i \leq U_i.$$

Now we construct the matrix $\mathbf{E}$ as follows:

$$\mathbf{E} = \begin{bmatrix} y_1 \mathbf{x}_1^1, & \ldots, & y_l \mathbf{x}_l^1 \\ \vdots & \vdots & \vdots \\ y_1 \mathbf{x}_1^t, & \ldots, & y_l \mathbf{x}_l^t \\ & \mathbf{I}_l & \end{bmatrix} \quad \mathbf{E}\boldsymbol{\alpha} = \begin{bmatrix} \mathbf{w}^1 \\ \vdots \\ \mathbf{w}^t \\ \boldsymbol{\alpha} \end{bmatrix}. \tag{17}$$

Define $g(\mathbf{E}\boldsymbol{\alpha}) = \frac{1}{q}\ln\sum_{t=1}^{T}\exp(q(\frac{1}{2}||\mathbf{w}^t||^2 + G(\boldsymbol{\alpha})))$. With $\mathbf{b} = 0, L_i = 0, U_i = C, \forall\ i$. In addition, as indicated before, its Hessian matrix $\nabla^2 g(\mathbf{E}\boldsymbol{\alpha})$ is positive definite for $q > 0$. All the conditions hold and the linear convergence is attained [19]. In addition, with Almost Cyclic Rule for choosing variable to optimize, and with Proposition 8, we have $\eta_q = (\tau_2\omega^2 n/\tau_1 \min_j ||\mathbf{E}_j||^2)$, where $\omega = (2 + ||\mathbf{E}||^2\tau_2)$, $\mathbf{E}_j$ is the jth column of $\mathbf{E}$ and n is the dimension of $\boldsymbol{\alpha}$. Finally, we have $\eta_q = (qL(||\mathbf{E}||^2 qL + 2)^2 n)/(\tau_1 \min_j ||\mathbf{E}_j||^2)$. More details can be seen in [19]. ■

Finally, from Proposition 2, let $\kappa_{q_k} = 1 - (1/\eta_{q_k})$, where η_{q_k} is defined in Lemma (6), k is the iteration index of the cutting plane algorithm and q_k is the corresponding smoothing parameter at the kth iteration, we have the following:

$$f(\boldsymbol{\alpha}^k) - f(\boldsymbol{\alpha}^*) \le \kappa_{q_k}^{(k-k_0)}(f(\boldsymbol{\alpha}^{k_0}) - f(\boldsymbol{\alpha}^*)) + \frac{(1 + \kappa_{q_k}^{(k-k_0)})\log(T)}{q_k}.$$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions, which have greatly improved this paper.

REFERENCES

[1] *Liblinear*. (2008) [Online]. Available: http://c2inet.sce.ntu.edu.sg/mingkui/fgm.htm

[2] *FGM*. (2010) [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[3] K. L. Ayers and H. J. Cordell, "SNP selection in genome-wide and candidate gene studies via penalized logistic regression," *Genet. Epidemiol.*, vol. 34, no. 8, pp. 879–891, 2010.

[4] E. J. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, 2007.

[5] Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear SVM," *J. Mach. Learn. Res.*, vol. 11, pp. 1471–1490, Apr. 2010.

[6] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney, "Feature selection methods for text classification," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 230–239.

[7] N. Ding and S. Vishwanathan, "T-logistic regression," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2010.

[8] H. Fei and J. Huan, "Boosting with structure information in the functional space: An application to graph classification," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, 643–652.

[9] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2002.

[10] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, *Feature Extraction, Foundations and Applications*. New York, NY, USA: Springer-Verlag, 2006.

[11] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[12] M. Heath, *Scientific Computing-An Introductory Survey*. New York, NY, USA: McGraw-Hill, 1997.

[13] S. S. Keerthi, K. Duan, S. K. Shevade, and A. Poo, "A fast dual algorithm for kernel logistic regression," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 151–165, 2005.

[14] J. Liao and K.-V. Chin, "Logistic regression for disease classification using microarray data: Model selection in a large p and small n case," *Bioinformatics*, vol. 23, no. 15, pp. 1945–1951, 2007.

[15] C.-J. Lin, R. C. Weng, and S. S. Keerthi, "Trust region Newton method for large-scale logistic regression," *J. Mach. Learn. Res.*, vol. 9, pp. 627–650, Jun. 2008.

[16] D. Lin, D. P. Foster, and L. H. Ungar, "A risk ratio comparison of L_0 and L_1 penalized regressions," Dept. Stat., Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep., 2010.

[17] Z. Liu, F. Jiang, G. Tian, S. Wang, F. Sato, S. J. Meltzer, and M. Tan, "Sparse logistic regression with l_p penalty for biomarker identification," *Stat. Appl. Genet. Molecular Biol.*, vol. 6, no. 1, pp. 1544–6115, 2007.

[18] A. C. Lozano, G. Swirszcz, and N. Abe, "Group orthogonal matching pursuit for logistic regression," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 452–460.

[19] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *J. Optim. Theory Appl.*, vol. 72, no. 1, pp. 7–35, 1992.

[20] A. Mutapcic and S. Boyd, "Cutting-set methods for robust convex optimization with pessimizing oracles," *Optim. Methods Softw.*, vol. 24, no. 3, pp. 381–406, 2009.

[21] A. Nedic and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J. Optim. Theory Appl.*, vol. 142, no. 1, pp. 205–228, 2009.

[22] A. Nemirovski, "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM J. Optim.*, vol. 15, no. 1, pp. 229–251, 2005.

[23] E. Y. Pee and J. O. Royset, "On solving large-scale finite minimax problems using exponential smoothing," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 390–421, 2010.

[24] A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.

[25] K.-Q. Shen, C.-J. Ong, X.-P. Li, and E. P. Wilder-Smith, "Feature selection via sensitivity analysis of SVM probabilistic outputs," *Mach. Learn.*, vol. 70, no. 1, pp. 1–20, 2008.

[26] J. Shi, W. Yin, S. Osher, and P. Sajda, "A fast hybrid algorithm for large-scale l_1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 11, pp. 713–741, Feb. 2010.

[27] M. Tan, I. Tsang, and L. Wang, "Learning sparse svm for feature selection on very high dimensional datasets," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1047–1054.

[28] M. Sion, "On general minimax theorems," *Pacific J. Math.*, vol. 8, no. 1, pp. 171–176, 1958.

[29] A. Tewari, P. Ravikumar, and I. S. Dhillon, "Greedy algorithms for structurally constrained high dimensional problems," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2011.

[30] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," Dept. Math., Univ. Washington, Seattle, WA, USA, Tech. Rep., 2008.

[31] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 1065–1072.

[32] S. Xu, "Smoothing method for minimax problems," *J. Optim. Theory App.*, vol. 20, no. 3, pp. 267–279, 2001.

[33] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Mach. Learn.*, vol. 85, nos. 1–2, pp. 41–75, 2010.

[34] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, Dec. 2004.

[35] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "A comparison of optimization methods and software for large-scale L_1-regularized linear classification," *J. Mach. Learn. Res.*, vol. 11, pp. 3183–3234, Jan. 2010.

[36] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, "An improved GLMNET for L1-regularized logistic regression and support vector machines," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2011.

[37] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *J. Mach. Learn. Res.*, vol. 11, pp. 1081–1107, Mar. 2010.

[38] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2007.

[39] J. Zhu and T. Hastie, "Classification of gene microarrays by penalized logistic regression," *Biostatistics*, vol. 5, no. 3, pp. 427–443, 2004.

[40] Z. Zhu, Y. S. Ong, and M. Dash, "Markov blanket-embedded genetic algorithm for gene selection," *Pattern Recognit.*, vol. 40, no. 11, pp. 3236–3248, 2007.

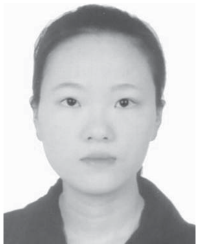
Mingkui Tan received the Bachelors degree in environmental science and engineering and the Masters degree in control science and engineering from Hunan University, Changsha, China, in 2006 and 2009, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore.

His current research interests include compressive sensing, machine learning, and large-scale convex optimization.

Ivor W. Tsang received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2007.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He is the Deputy Director of the Center for Computational Intelligence, NTU.

Dr. Tsang was a recipient of the Natural Science Award (Class II) in 2008, China, which recognized his contributions to kernel methods. He received the prestigious IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2004 Paper Award in 2006, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR in 2010, the Best Paper Award at ICTAI in 2011, the Best Poster Award Honorable Mention at ACML in 2012, and the Best Paper Award from the IEEE Hong Kong Chapter of Signal Processing Postgraduate Forum in 2006. He was a recipient of the Microsoft Fellowship in 2005, and ECCV in 2012 Outstanding Reviewer Award.



Li Wang received the Bachelors degree in information and computing science from the China University of Mining and Technology, Jiangsu, China, in 2006, and the Masters degree in computational mathematics from Xi’an Jiaotong University, Shaanxi, China, in 2009. She is currently pursuing the Ph.D. degree with the Department of Mathematics, University of California, San Diego, CA, USA.

Her current research interests include large scale polynomial optimization, semi-infinite polynomial programming, and machine learning.