Accelerating Model Selection with Safe Screening for L_1-Regularized L_2-SVM

Zheng Zhao, Jun Liu, and James Cox

SAS Institute Inc., 600 Research Drive, Cary, NC 27513, USA
{zheng.zhao,jun.liu,james.cox}@sas.com

Abstract. The L_1-regularized support vector machine (SVM) is a powerful predictive learning model that can generate sparse solutions. Compared to a dense solution, a sparse solution is usually more interoperable and more effective for removing noise and preserving signals. The L_1-regularized SVM has been successfully applied in numerous applications to solve problems from text mining, bioinformatics, and image processing. The regularization parameter has a significant impact on the performance of an L_1-regularized SVM model. Therefore, model selection needs to be performed to choose a good regularization parameter. In model selection, one needs to learn a solution path using a set of predefined parameter values. Therefore, many L_1-regularized SVM models need to be fitted, which is usually very time consuming. This paper proposes a novel safe screening technique to accelerate model selection for the L_1-regularized L_2-SVM, which can lead to much better efficiency in many scenarios. The technique can successfully identify most inactive features in an optimal solution of the L_1-regularized L_2-SVM model and remove them before training. To achieve safe screening, the technique solves a minimization problem for each feature on a convex set that is formed by the intersection of a tight n-dimensional hyperball and the upper half-space. An efficient algorithm is designed to solve the problem based on zero-finding. Every feature that is removed by the proposed technique is guaranteed to have zero weight in the optimal solution. Therefore, an L_1-regularized L_2-SVM solver achieves exactly the same result by using only the selected features as when it uses the full feature set. Empirical study on high-dimensional benchmark data sets produced promising results and demonstrated the effectiveness of the proposed technique.

Keywords: Screening, sparse support vector machine, model selection.

1 Introduction

Feature selection is an effective technique for dimensionality reduction and relevance detection [1]. The L_1-regularized support vector machine (SVM) is a powerful feature selection algorithm [3, 4, 5, 6] that is in the embedded model [2]. It can simultaneously fit a model by margin maximization and remove noisy features by soft-thresholding. It has been successfully applied to solve many problems in text mining, bioinformatics, and image processing. The L_1-regularized

T. Calders et al. (Eds.): ECML PKDD 2014, Part III, LNCS 8726, pp. 385–400, 2014.

SVM enjoys two major advantages compared to other variances of sparse SVM models [7, 8, 9]: first, it solves a convex problem; therefore, an optimal solution can always be obtained without any relaxation of the original problem. Second, it is efficient. A well-implemented L_1-regularized SVM solver can readily handle problems that have tens of millions samples and features [6].

The value of the regularization parameter λ has a significant impact on the performance of an L_1-regularized SVM model. Model selection can be used to select a good parameter value. During model selection, a series of L_1-regularized SVM models need to be fit for a set of predefined regularization parameter values. The best regularization parameter value can be chosen by using a prespecified criterion, such as the accuracy or the area under the curve (AUC) that is achieved by the resulting models on holdout samples. When data are huge, the computational cost of model selection can be prohibitive. Assume that k regularization parameter values, $\lambda_1 > \lambda_2 > \ldots > \lambda_k$, need to be tried in a model selection process. It is easy to see that this process can be greatly accelerated if the solution obtained for λ_i can be used to speed up the computation of the solution for λ_{i+1}. Based on this idea, highly efficient screening techniques are recently proposed for Lasso [10] to accelerate its model selection. The key idea is that, given a solution $\boldsymbol{w}_1^*$ for $\lambda = \lambda_1$, many features that have zero coefficients in $\boldsymbol{w}_2^*$ when $\lambda = \lambda_2$ can be identified. By removing these "inactive" features, the cost for computing $\boldsymbol{w}_2^*$ can be significantly reduced. Although effective screening algorithms have been designed for Lasso [11, 12, 13, 14, 15], research into screening for the L_1-regularized SVM is largely untouched.

In this paper, a novel screening technique is proposed to speed up model selection for an L_1-regularized L_2-SVM.[1] The technique makes use of the variational inequality [16] and the nonnegative constraint on the dual variables of the L_1-regularized L_2-SVM model for constructing a tight convex set, which can be used to compute bounds for screening features. A prescreening strategy and a fast zero-finding algorithm are designed and implemented to ensure the efficiency of the screening process. Features that are removed by the technique are guaranteed to be inactive in the optimal solution. Therefore, the screening technique is "safe," because an L_1-regularized L_2-SVM solver can achieve exactly the same result when it uses the features selected by the technique as when it uses the full feature set. To the best knowledge of the authors, this is the first screening technique that is proposed for accelerating the speed of model selection for the L_1-regularized L_2-SVM. Empirical study on five high-dimensional benchmark data sets produced promising results and demonstrated that the proposed screening technique can greatly speed up model selection for an L_1-regularized L_2-SVM by efficiently removing a large number of inactive features.

[1] Our ongoing work will extend the technique proposed in this paper to screen features for the L_1-regularized L_1-SVM.

2 L_1-Regularized L_2-SVM

Assume that $\mathbf{X} \in \mathrm{I\!R}^{m \times n}$ is a data set that contains n samples, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, and m features, $\mathbf{X} = \left(\mathbf{f}_1^\top, \ldots, \mathbf{f}_m^\top\right)^\top$. Assume also that $\mathbf{y} = (y_1, \ldots, y_n)$ contains n class labels, $y_i \in \{-1, +1\}$, $i = 1, \ldots, n$. Let $\mathbf{w} \in \mathrm{I\!R}^m$ be the m-dimensional weight vector, let $\xi_i \geq 0, i = 1, \ldots, n$ be the n slack variables, and let $b \in \mathrm{I\!R}$ and $\lambda \in \mathrm{I\!R}^+$ be the bias and the regularization parameter, respectively. The primal form of the L_1-regularized L_2-SVM is defined as:

$$\min_{\boldsymbol{\xi}, \mathbf{w}} \frac{1}{2} \sum_{i=1}^{n} \xi_i^2 + \lambda ||\mathbf{w}||_1, \tag{1}$$
$$s.t. \;\; y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

Eq. (1) specifies a convex problem that has a non-smooth L_1 regularizer, which enforces that the solution is sparse. Let $\boldsymbol{w}^\star(\lambda)$ be the optimal solution of Eq. (1) for a given λ. All the features that have nonzero values in $\boldsymbol{w}^\star(\lambda)$ are called active features, and the other features are called inactive. Let $\boldsymbol{\alpha} \in \mathrm{I\!R}^n$ be the n-dimensional dual variable. By applying the Lagrangian multiplier [17], the dual of the problem defined in Eq. (1) can be obtained as:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha} - \mathbf{1}\|_2^2 \qquad , \tag{2}$$
$$s.t. \;\; \|\hat{\mathbf{f}}_j^\top \boldsymbol{\alpha}\| \leq \lambda, \quad j = 1, \ldots, m, \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \;\; \boldsymbol{\alpha} \succcurlyeq \mathbf{0}.$$

Here, $\hat{\mathbf{f}} = \mathbf{Y}\mathbf{f}$, and $\mathbf{Y} = diag(\mathbf{y})$ is a diagonal matrix. By defining $\boldsymbol{\alpha} = \lambda \boldsymbol{\theta}$, Eq. (2) can be reformulated as:

$$\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - \frac{\mathbf{1}}{\lambda}||_2^2 \qquad , \tag{3}$$
$$s.t. \;\; \|\hat{\mathbf{f}}_j^\top \boldsymbol{\theta}\| \leq 1, \quad j = 1, \ldots, m, \quad \sum_{i=1}^{n} \theta_i y_i = 0, \;\; \boldsymbol{\theta} \succcurlyeq \mathbf{0}.$$

In the primal formulation for the L_1-regularized L_2-SVM, the primal variables are b, $\mathbf{w}$, and $\boldsymbol{\xi}$. And in the dual formulation, the dual variables are $\boldsymbol{\alpha}$ or $\boldsymbol{\theta}$. When b and $\mathbf{w}$ are known, $\boldsymbol{\xi}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\theta}$ can be obtained as:

$$\xi_i = \alpha_i = \lambda \theta_i = \max\left(0, 1 - y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right)\right). \tag{4}$$

The relation between $\boldsymbol{\theta}$ and $\mathbf{w}$ can be expressed as:

$$\boldsymbol{\theta}^\top \hat{\mathbf{f}}_j = \begin{cases} \operatorname{sign}(w_j), & \text{if } w_j \neq 0 \\ [-1, +1], & \text{if } w_j = 0 \end{cases}, \quad j = 1, \ldots, m. \tag{5}$$

$\lambda_{\max}$ is defined as the smallest λ value that leads to $\mathbf{w} = \mathbf{0}$ when it is used in Eq. (1). Given a data set $(\mathbf{X}, \mathbf{y})$, $\lambda_{\max}$ can be obtained in a closed form as:

$$\lambda_{\max} = \left\| \sum_{i=1}^{n} \left(y_i - \frac{n_+ - n_-}{n}\right) \mathbf{x}_i \right\|_\infty , \tag{6}$$

where n_+ and n_- denote the number of positive and negative samples, respectively. And when $\lambda \geq \lambda_{max}$, the optimal solution of the problem defined in Eq. (1) can be written as:

$$\mathbf{w}^\star = \mathbf{0}, \quad b^\star = \frac{(n_+ - n_-)}{n}. \tag{7}$$

Denote $\mathbf{m} = \sum_{i=1}^{n} \left(y_i - \frac{n_+ - n_-}{n} \right) \mathbf{x}_i$. The first feature to enter the model is the one that corresponds to the element that has the largest magnitude in $\mathbf{m}$.

3 Safe Screening for L_1-Regularized L_2-SVM

Eq. (5) shows that the necessary condition for a feature $\mathbf{f}$ to be active in an optimal solution is $|\boldsymbol{\theta}^\top \hat{\mathbf{f}}| = 1$. On the other hand, for any feature $\mathbf{f}$, if $|\boldsymbol{\theta}^\top \hat{\mathbf{f}}| < 1$, it must be inactive in the optimal solution. Given a λ value, this condition can be used to develop a rule for screening inactive features to speed up training for the L_1-regularized L_2-SVM. The key is to compute the upper bound of $|\boldsymbol{\theta}^\top \hat{\mathbf{f}}|$ for features. A feature can be safely removed if its upper bound value is less than 1. The cost of computing the upper bounds can be much lower than training L_1-regularized L_2-SVM. Therefore, screening can greatly lower the computational cost by removing many inactive features before training.

To bound the value of $|\boldsymbol{\theta}^\top \hat{\mathbf{f}}|$, it is necessary to construct a closed convex set $\mathbf{K}$ that contains $\boldsymbol{\theta}$. The upper bound value can be then computed by maximizing $|\boldsymbol{\theta}^\top \hat{\mathbf{f}}|$ over $\mathbf{K}$, which defines a convex problem with a unique solution.

3.1 Constructing the Convex Set K

Given $\lambda_1, \ldots, \lambda_k$, k models need to be trained for model selection. Let $\boldsymbol{\theta}_i$ be the solution that corresponds to λ_i, this section shows that $\boldsymbol{\theta}_i$ can be used to construct a convex set that contains $\boldsymbol{\theta}_{i+1}$ for bounding the value of $|\boldsymbol{\theta}_{i+1}^\top \hat{\mathbf{f}}|$. When λ_i is close to λ_{i+1}, this convex set can be very tight.

Assume that $\boldsymbol{\theta}^\star$ is the optimal solution of Eq. (3) and $t \geq 0$. It is easy to verify that $\boldsymbol{\theta}^\star$ is also the optimal solution of the following problem:

$$\min_{\boldsymbol{\theta}} \left\| \boldsymbol{\theta} - \left(t \frac{\mathbf{1}}{\lambda} + (1-t)\, \boldsymbol{\theta}^\star \right) \right\|_2^2 \qquad , \tag{8}$$
$$s.t. \;\; \|\hat{\mathbf{f}}_j^\top \boldsymbol{\theta}\| \leq 1, \;\; j = 1, \ldots, m, \;\; \sum_{i=1}^{n} \theta_i y_i = 0, \;\; \boldsymbol{\theta} \succcurlyeq \mathbf{0}.$$

In the following, Eq. (8) and the variational inequality [16] are used to construct a closed convex set $\mathbf{K}$ to bound $|\boldsymbol{\theta}^\top \hat{\mathbf{f}}|$. Proposition 1 introduces the variational inequality for a convex optimization problem.

Proposition 1. *Let $\boldsymbol{\theta}^\star$ be an optimal solution of a convex problem:*

$$\min g(\boldsymbol{\theta}), \quad s.t. \quad \boldsymbol{\theta} \in \mathbf{K},$$

where g is continuously differentiable and $\mathbf{K}$ is closed and convex. Then the following variational inequality holds:

$$\nabla g\left(\boldsymbol{\theta}^\star\right)^\top \left(\boldsymbol{\theta} - \boldsymbol{\theta}^\star\right) \geq 0, \quad \forall \boldsymbol{\theta} \in \mathbf{K}.$$

Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ be the optimal solutions of the problem defined in Eq. (3) and Eq. (8) for λ_1 and λ_2, respectively. Assume that $\lambda_1 > \lambda_2$ and that $\boldsymbol{\theta}_1$ is known[2]. The following results can be obtained by applying Proposition 1 to the problem defined in Eq. (8) for $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively:

$$\left(\boldsymbol{\theta}_1 - \left(t_1 \frac{\mathbf{1}}{\lambda_1} + (1 - t_1)\,\boldsymbol{\theta}_1\right)\right)^\top \left(\boldsymbol{\theta} - \boldsymbol{\theta}_1\right) \geq 0, \tag{9}$$

$$\left(\boldsymbol{\theta}_2 - \left(t_2 \frac{\mathbf{1}}{\lambda_2} + (1 - t_2)\,\boldsymbol{\theta}_2\right)\right)^\top \left(\boldsymbol{\theta} - \boldsymbol{\theta}_2\right) \geq 0. \tag{10}$$

Let $t = \frac{t_1}{t_2} \geq 0$. By substituting $\boldsymbol{\theta} = \boldsymbol{\theta}_2$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_1$ into Eq. (9) and Eq. (10), respectively, and then combining the two inequalities, it leads to:

$$\mathbf{B}_t = \left\{\boldsymbol{\theta}_2 : \left(\boldsymbol{\theta}_2 - \mathbf{c}\right)^\top \left(\boldsymbol{\theta}_2 - \mathbf{c}\right) \leq l^2\right\}, \tag{11}$$

$$\mathbf{c} = \frac{1}{2}\left(t\boldsymbol{\theta}_1 - t\frac{\mathbf{1}}{\lambda_1} + \frac{\mathbf{1}}{\lambda_2} + \boldsymbol{\theta}_1\right), l = \frac{1}{2}\left\|t\boldsymbol{\theta}_1 - t\frac{\mathbf{1}}{\lambda_1} + \frac{\mathbf{1}}{\lambda_2} - \boldsymbol{\theta}_1\right\|_2.$$

As the value of t changes from 0 to ∞, Eq. (11) generates a series of hyperballs that contains $\boldsymbol{\theta}_2$. The following theorem studies when the radius of the hyperball generated by Eq. (11) reaches its minimum:

Theorem 1. *Let $\mathbf{a} = \frac{\frac{\mathbf{1}}{\lambda_1} - \boldsymbol{\theta}_1}{\left\|\frac{\mathbf{1}}{\lambda_1} - \boldsymbol{\theta}_1\right\|_2}$. The radius of the hyperball generated by Eq. (11) reaches it minimum when*

$$t = 1 + \left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1}\right) \frac{\mathbf{a}^\top \mathbf{1}}{\left\|\frac{\mathbf{1}}{\lambda_1} - \boldsymbol{\theta}_1\right\|_2}. \tag{12}$$

Let $\mathbf{c}$ be the center of the ball and l be the radius. When the minimum is reached, they can be computed as:

$$\mathbf{c} = \frac{1}{2}\left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1}\right) P_{\mathbf{a}}\left(\mathbf{1}\right) + \boldsymbol{\theta}_1,\ l = \frac{1}{2}\left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1}\right) \left\|P_{\mathbf{a}}\left(\mathbf{1}\right)\right\|. \tag{13}$$

Here, $P_{\mathbf{u}}\left(\mathbf{v}\right) = \mathbf{v} - \frac{\mathbf{v}^\top \mathbf{u}}{\|\mathbf{u}\|_2^2}\mathbf{u}$ is an operator that projects $\mathbf{v}$ to the null-space of $\mathbf{u}$. Since $\|\mathbf{a}\|_2 = 1$, $P_{\mathbf{a}}\left(\mathbf{1}\right) = \mathbf{1} - \left(\mathbf{a}^\top \mathbf{1}\right)\mathbf{a}$.

[2] When $\lambda_1 = \lambda_{max}$, $\mathbf{w} = 0$ and $\boldsymbol{\theta}_1$ is given in Eq. (4).

Proof. The theorem can be proved by minimizing the r defined in Eq. (11).

□

Theorem 1 suggests that when $t = 1 + \left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1}\right) \mathbf{a}^\top \mathbf{1} \left\| \frac{\mathbf{1}}{\lambda_1} - \boldsymbol{\theta}_1 \right\|_2^{-1}$, the volume of $\mathbf{B}_t$ is minimized, which forms a good basis for constructing $\mathbf{K}$. The nonnegative constraint on the dual variable confines $\boldsymbol{\theta}$ in the upper half-space: $\boldsymbol{\theta} \succcurlyeq \mathbf{0}$, and can be used to further reduce the volume of $\mathbf{K}$:

$$\mathbf{K} = \left\{ \boldsymbol{\theta} : (\boldsymbol{\theta} - \mathbf{c})^\top (\boldsymbol{\theta} - \mathbf{c}) \leq l^2, \boldsymbol{\theta} \succcurlyeq \mathbf{0} \right\}, \tag{14}$$

$$\mathbf{c} = \frac{1}{2}\left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1}\right) P_{\mathbf{a}}(\mathbf{1}) + \boldsymbol{\theta}_1,\ l = \frac{1}{2}\left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1}\right) \| P_{\mathbf{a}}(\mathbf{1}) \|.$$

3.2 Computing the Upper Bound

Given the convex set $\mathbf{K}$ defined in Eq. (14), the maximum value of $\left|\boldsymbol{\theta}_2^\top \hat{\mathbf{f}}\right|$ can be computed by solving the problem:

$$\max \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right|,\ s.t.\ (\boldsymbol{\theta} - \mathbf{c})^\top (\boldsymbol{\theta} - \mathbf{c}) \leq l^2,\ \boldsymbol{\theta} \succcurlyeq \mathbf{0}. \tag{15}$$

Since the following equation holds:

$$\max |x| = \max \{-\min(x), \max(x)\} = \max \{-\min(x), -\min(-x)\}.$$

The computation of $\max \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right|$ can be decomposed to the following two subproblems: $m_1 = -\min \boldsymbol{\theta}^\top \hat{\mathbf{f}}$, $m_2 = -\min \boldsymbol{\theta}^\top (-\hat{\mathbf{f}})$. And $\max \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right| = \max (m_1, m_2)$. This suggests that the key to bound $\left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right|$ is to compute:

$$\min \boldsymbol{\theta}^\top \hat{\mathbf{f}},\ s.t.\ (\boldsymbol{\theta} - \mathbf{c})^\top (\boldsymbol{\theta} - \mathbf{c}) \leq l^2,\ \boldsymbol{\theta} \succcurlyeq \mathbf{0}. \tag{16}$$

Its Lagrangian $L(\boldsymbol{\theta}, \alpha, \boldsymbol{\nu})$ can be written as:

$$L(\boldsymbol{\theta}, \alpha, \boldsymbol{\nu}) = \boldsymbol{\theta}^\top \hat{\mathbf{f}} + \frac{1}{2}\alpha \left(\|\boldsymbol{\theta} - \mathbf{c}\|_2^2 - l^2 \right) + \boldsymbol{\nu}^\top \boldsymbol{\theta},\ \alpha \geq 0,\ \boldsymbol{\nu} \succcurlyeq \mathbf{0}. \tag{17}$$

Since $\|\boldsymbol{\theta} - \mathbf{c}\|_2^2 \leq l^2$, the problem specified in Eq. (16) is bounded from below by $-(\|\mathbf{c}\|_2 + l) \|\mathbf{f}\|_2$. Thus, $\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \alpha, \boldsymbol{\nu})$ is also bounded from below. Since the minimum achieves on the boundary, it must hold that $\alpha > 0$. It is also easy to verify that $\alpha = 0 \Rightarrow \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right| = 0$.

Setting the derivative of $L(\boldsymbol{\theta}, \alpha, \boldsymbol{\nu})$ to be zero leads to the equation:

$$\mathbf{f} + \alpha (\boldsymbol{\theta} - \mathbf{c}) - \boldsymbol{\nu} = 0 \Rightarrow \boldsymbol{\theta} = \frac{1}{\alpha}\boldsymbol{\nu} - \frac{1}{\alpha}\mathbf{f} + \mathbf{c}.$$

Therefore, $\theta_i = \frac{1}{\alpha}\nu_i - \frac{1}{\alpha}f_i + c_i,\ i = 1, \ldots, n$. According to the complementary slackness condition, $\boldsymbol{\nu}^\top \boldsymbol{\theta} = 0$. Also since $\boldsymbol{\nu} \succcurlyeq 0$ and $\boldsymbol{\theta} \succcurlyeq 0$. It must hold that $\nu_i\theta_i = 0,\ i = 1, \ldots, n$. These conditions lead to the following equations:

$$\theta_i = \max\left(c_i - \frac{1}{\alpha}f_i,\ 0\right). \tag{18}$$

This suggests that when α is know, $\boldsymbol{\theta}$ can be computed by using Eq. (18). In the following, it shows that the value of α can be efficiently computed by solving a zero finding problem through binary search.

Computing $\boldsymbol{\alpha}$ via zero finding According to the complementary slackness condition, $\alpha\left(\|\boldsymbol{\theta} - \mathbf{c}\|_2^2 - l^2\right) = 0$. Because $\alpha > 0$, it must hold that:

$$\|\boldsymbol{\theta} - \mathbf{c}\|_2^2 - l^2 = 0 \Rightarrow \boldsymbol{\theta}^\top\boldsymbol{\theta} - 2\mathbf{c}^\top\boldsymbol{\theta} - l^2 + \mathbf{c}^\top\mathbf{c} = 0. \tag{19}$$

Let $\mathcal{A} = \{i : \theta_i > 0\}$. The following equation can be obtained.

$$\boldsymbol{\theta}^\top\boldsymbol{\theta} - 2\mathbf{c}^\top\boldsymbol{\theta} - l^2 + \mathbf{c}^\top\mathbf{c} = \sum_{i\in\mathcal{A}}\theta_i^2 - 2\sum_{i\in\mathcal{A}}c_i\theta_i - l^2 + \mathbf{c}^\top\mathbf{c}. \tag{20}$$

By plugging Eq. (18) into Eq. (20). A function of α can be obtained as:

$$g\left(\frac{1}{\alpha}\right) = \frac{1}{\alpha^2}\sum_{i\in\mathcal{A}}f_i^2 - \sum_{i\in\mathcal{A}}c_i^2 - l^2 + \mathbf{c}^\top\mathbf{c}. \tag{21}$$

And the α value can be obtained by solving the zero finding problem:

$$g\left(\frac{1}{\alpha}\right) = 0 \tag{22}$$

The following theorem suggests that $g\left(\frac{1}{\alpha}\right)$ monotonically increases as $\frac{1}{\alpha}$ increases. Therefore this problem can be solved efficiently via binary search.

Theorem 2. *The function $g\left(\frac{1}{\alpha}\right)$ monotonically increases as $\frac{1}{\alpha}$ increases.*

Proof. Assume that $g_i\left(\frac{1}{\alpha}\right)$ is defined as:

$$g_i\left(\frac{1}{\alpha}\right) = \begin{cases} i \in \mathcal{A}, & \frac{1}{\alpha^2}f_i^2 - c_i^2 \\ i \notin \mathcal{A}, & 0 \end{cases}. \tag{23}$$

$g\left(\frac{1}{\alpha}\right)$ can be rewritten as:

$$g\left(\frac{1}{\alpha}\right) = \sum\nolimits_{i=1}^{n} g_i\left(\frac{1}{\alpha}\right) - l^2 + \mathbf{c}^\top\mathbf{c}.$$

The theorem can be proved by showing that for $\forall i \in \{1, \ldots, n\}$, $g_i\left(\frac{1}{\alpha}\right)$ either increases monotonically as $\frac{1}{\alpha}$ increases, or is a constant. Let $\epsilon > 0$, this can be proved by comparing $g_i\left(\frac{1}{\alpha}\right)$ to $g_i\left(\frac{1}{\alpha} + \epsilon\right)$ in the following four cases.

1. $c_i > 0,\ f_i \leq 0$: $c_i > 0,\ f_i \leq 0 \Rightarrow \theta_i = c_i - \frac{1}{\alpha} f_i,\ i \in \mathcal{A}$, for $\forall \frac{1}{a} \in \mathrm{I\!R}^+$. In this case $g_i\left(\frac{1}{\alpha}\right)$ can be written as:
$$g_i\left(\frac{1}{\alpha}\right) = \frac{1}{\alpha^2} f_i^2 - c_i^2. \tag{24}$$
And it can be verify that $g_i\left(\frac{1}{\alpha} + \epsilon\right) > g_i\left(\frac{1}{\alpha}\right)$ when $c_i > 0,\ f_i \leq 0$.
2. $c_i \leq 0,\ f_i > 0$: $c_i \leq 0,\ f_i > 0 \Rightarrow \theta_i = 0,\ i \notin \mathcal{A}$, for $\forall \frac{1}{a} \in \mathrm{I\!R}^+$. In this case $g_i\left(\frac{1}{\alpha}\right)$ can be written as:
$$g_i\left(\frac{1}{\alpha}\right) = 0. \tag{25}$$
Therefore, $g_i\left(\frac{1}{\alpha}\right)$ is a constant when $c_i \leq 0,\ f_i > 0$.
3. $c_i > 0,\ f_i > 0$: In this case $g_i\left(\frac{1}{\alpha}\right)$ can be written as:
$$g_i\left(\frac{1}{\alpha}\right) = \begin{cases} \frac{1}{\alpha} \in \left(0, \frac{c_i}{f_i}\right) \Rightarrow \theta_i = c_i - \frac{1}{\alpha} f_i, i \in \mathcal{A}, & \frac{1}{\alpha^2} f_i^2 - c_i^2 \\ \frac{1}{\alpha} \in \left[\frac{c_i}{f_i}, +\infty\right) \Rightarrow \theta_i = 0, i \notin \mathcal{A} \quad , & 0 \end{cases} \tag{26}$$
And it can be verify that $g_i\left(\frac{1}{\alpha} + \epsilon\right) > g_i\left(\frac{1}{\alpha}\right)$ when $c_i > 0, f_i > 0$.
4. $c_i < 0,\ f_i \leq 0$: In this case $g_i\left(\frac{1}{\alpha}\right)$ can be written as:
$$g_i\left(\frac{1}{\alpha}\right) = \begin{cases} \frac{1}{\alpha} \in \left(0, \frac{c_i}{f_i}\right] \Rightarrow \theta_i = 0, i \notin \mathcal{A} \quad , & 0 \\ \frac{1}{\alpha} \in \left(\frac{c_i}{f_i}, +\infty\right) \Rightarrow \theta_i = c_i - \frac{1}{\alpha} f_i, i \in \mathcal{A}, & \frac{1}{\alpha^2} f_i^2 - c_i^2 \end{cases} \tag{27}$$
It can also be verify that $g_i\left(\frac{1}{\alpha} + \epsilon\right) > g_i\left(\frac{1}{\alpha}\right)$ when $c_i < 0,\ f_i \leq 0$.

This finishes the proof of the theorem.

□

When a value is given to α, $\mathcal{A}$ can be determined via computing θ_i by using one of the four equations (Eq. (24) – Eq. (27)) provided in Theorem 2 according to the value of c_i and f_i. And the obtained $\mathcal{A}$ can be used to compute the value of $g\left(\frac{1}{\alpha}\right)$. When $\mathcal{A}$ is determined, solving Eq. (22) leads to the following equation:

$$\frac{1}{\alpha'} = \sqrt{\frac{l^2 - \mathbf{c}^\top \mathbf{c} + \sum\limits_{i \in \mathcal{A}} c_i^2}{\sum\limits_{i \in \mathcal{A}} f_i^2}} \tag{28}$$

Let an index set $\mathcal{B}$ is defined as $\mathcal{B} = \{i : (c_i > 0, f_i > 0) \text{ or } (c_i < 0, f_i \leq 0)\}$. Assume that $\mathcal{B}$ contains k members. A sorted index set $\mathcal{B}_{sorted} = \{i_1, \ldots, i_k\}$ can be obtained by sorting the value of $\frac{c_i}{f_i},\ i \in \mathcal{B}$. The following theorem provides the stopping condition for using binary search to solve the zero finding problem.

Theorem 3. *Let $\mathcal{T} = \left\{0, \frac{c_{i_1}}{f_{i_1}}, \ldots, \frac{c_{i_k}}{f_{i_k}}, +\infty\right\} = \{t_1, \ldots, t_{k+2}\}$, where $i_1, \ldots, i_k$ are the k sorted indices in $\mathcal{B}_{sorted}$. Given $\frac{1}{\alpha}$, assume that $t_j < \frac{1}{\alpha} \leq t_{j+1}$. The binary search stops when the $\frac{1}{\alpha'}$ computed by using Eq. (28) also satisfies that $t_j < \frac{1}{\alpha'} \leq t_{j+1}$. In this case, set $\frac{1}{\alpha} = \frac{1}{\alpha'}$ and it can verified that $g\left(\frac{1}{\alpha}\right) = 0$.*

Proof. The theorem can be proved by using the fact that when the value of $\frac{1}{\alpha}$ varies in $(t_j, t_{j+1}]$, $\mathcal{A}$ keeps unchanged.

□

Theorem 3 suggests that t_{k+1} can be used as the starting point for binary search. If $g\left(\frac{1}{\alpha}\right) > 0$, decrease $\frac{1}{\alpha}$. If $g\left(\frac{1}{\alpha}\right) < 0$, increase $\frac{1}{\alpha}$. The search stops when the condition specified in Theorem 3 is satisfied. And the obtained $\frac{1}{\alpha}$ and $\mathcal{A}$ can be used to compute $\boldsymbol{\theta}^\top \hat{\mathbf{f}}$ by using the following equation:

$$\boldsymbol{\theta}^\top \hat{\mathbf{f}} = \sum_{i \in \mathcal{A}} c_i f_i - \frac{1}{\alpha} \sum_{i \in \mathcal{A}} f_i^2. \tag{29}$$

3.3 Computing the Upper Bound without Using $\boldsymbol{\theta} \succcurlyeq \mathbf{0}$

When $\boldsymbol{\theta} \succcurlyeq 0$ is not used to construct $\mathbf{K}$, $\max \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right|$ has a closed form solution on the hyper-ball defined in Theorem 1.

Theorem 4. *The optimization problem:*

$$\max \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right|, \ s.t. \ (\boldsymbol{\theta} - \mathbf{c})^\top (\boldsymbol{\theta} - \mathbf{c}) \leq l^2, \tag{30}$$

has a closed form solution:

$$\max \left|\boldsymbol{\theta}^\top \hat{\mathbf{f}}\right| = \left|\mathbf{c}^\top \hat{\mathbf{f}}\right| + l \left\|\hat{\mathbf{f}}\right\|_2. \tag{31}$$

Proof. The theorem can be proved by using the method of Lagrange multipliers.

□

Let m be the bound computed by solving Eq. (15), and m' be the bound computed by solving Eq. (30). It is easy to see that $m < m'$, since the $\mathbf{K}$ used in Eq. (15) is tighter. However, since m' can be computed in closed form, its computational cost is low . Therefore, it can be used for pre-screening features. More specifically, If $m' < 1$, there is no need to compute m by solving Eq. (15), since $m < m' < 1$. Computing m requires to solve a zero finding problem using binary search which is usually more expensive than computing Eq. (31).

3.4 The Screening Algorithm

Algorithm 1 shows the procedure of screening features for L_1-regularized L_2-SVM. Given λ_1, λ_2, and $\boldsymbol{\theta}_1$, the algorithm returns a list $\mathcal{L}$, which contains the indices of the features that are potentially active in the optimal solution that corresponds to λ_2. The algorithm first weights a feature using $\mathbf{Y}$ in Line 3. It then computes a bound for $|\hat{\mathbf{f}}^\top \boldsymbol{\theta}|$ using Eq. (31) in Line 4. If this bound is less than 1, the algorithm goes to test the next feature. This is the pre-screening step for improving algorithm's efficiency by using a bound that is cheaper to compute. If a feature passes the pre-screening, the algorithm computes a tighter bound for the feature in Line 8 and Line 9. If the bound is larger than 1, it adds the index of the feature to $\mathcal{L}$ in Line 11. The function $\mathsf{neg\_min}(\hat{\mathbf{f}})$ computes $-\min \boldsymbol{\theta}_2^\top \hat{\mathbf{f}}$. It first solves a zero finding problem for $\hat{\mathbf{f}}$ in Line 17, then uses the obtained $\frac{1}{\alpha}$ and $\mathcal{A}$ to compute $\min \boldsymbol{\theta}_2^\top \hat{\mathbf{f}}$ in Line 18. It returns $-\min \boldsymbol{\theta}_2^\top \hat{\mathbf{f}}$ in Line 19. The function $\mathsf{zero\_finding}(\hat{\mathbf{f}})$ solves the zero finding problem. This function first uses $max\left(\frac{c_j}{f_j}, j \in \mathcal{B}\right)$ as the starting value for $\frac{1}{\alpha}$. If $g\left(\frac{1}{\alpha}\right) < 0$, it must hold that $\frac{1}{\alpha} \leq \frac{1}{\alpha'} < \infty$. Therefore the stopping condition specified in Theorem 3 is satisfied. The algorithm returns $\frac{1}{\alpha}$ and $\mathcal{A}$ in Line 28. Otherwise it setups the *low* and *high* variables for binary search. The binary search is performed in Line 32 to Line 45. The stopping condition is tested in Line 36. If this condition is satisfied, the function stops searching and returns $\frac{1}{\alpha}$ and $\mathcal{A}$.

The algorithm needs to be implemented carefully to ensure efficiency. First, each step of the computation needs to be decomposed to many small substeps, so that the intermediate results obtained in the preceding substeps can be used by the following substeps to accelerate computation. Second, the substeps need to be organized and ordered properly so that no redundant computation is performed. It turns out the procedure listed in Algorithm 1 can be very efficient.

The pre-screening step requires to compute $\mathbf{Yf}$, $\mathbf{f}^\top\mathbf{y}$, and $\mathbf{f}^\top\mathbf{f}$. Since these computations are independent of $\boldsymbol{\theta}_1$, λ_1, and λ_2. Therefore, they can been pre-computed before training[3], and the cost is $O(mp)$ for m features. Here p is the average feature length[4]. The pre-screening step also requires to compute $\boldsymbol{\theta}_1^\top \mathbf{1}$ and $\boldsymbol{\theta}_1^\top \boldsymbol{\theta}_1$. They are shared by all the features. So they can be computed at the beginning of screening, and the cost is $O(n)$. For each feature, the pre-screening step requires to compute $\boldsymbol{\theta}_1^\top \mathbf{f}$, and its cost is $O(mp)$ for m features. However, when a solver fits a L_1-regularized L_2-SVM model, it might have already computed $\hat{\mathbf{f}}^\top \boldsymbol{\theta}_1$ as an intermediate result for all the features. In this case, $\hat{\mathbf{f}}^\top \boldsymbol{\theta}_1$ can be obtained from the solver for screening features at no cost. Given these intermediate results, the bound in the pre-screening step can be obtained in $O(m)$ for m feature. Therefore, the total computational cost for pre-screening m features is $O(mp)$. And if $\hat{\mathbf{f}}^\top \boldsymbol{\theta}_1$ can be obtained from the intermediate results generated by the L_1-regularized L_2-SVM solver and $\mathbf{Yf}$, $\mathbf{f}^\top\mathbf{y}$, and $\mathbf{f}^\top\mathbf{f}$ are precomputed before training, the total cost can decrease to just $O(m+n)$.

Assume that q features passed the pre-screening[5]. To compute the tighter bounds for these features, the algorithm requires to compute $\mathbf{c}$ and l. The cost is $O(n)$. For each feature, it can be verified that the algorithm takes at most $O(\log(p))$ steps to solve the zero finding problem. In each step, it takes $O(p)$ to determine $\mathcal{A}$ and compute $g\left(\frac{1}{\alpha}\right)$. Thus, cost for solving the zero-finding problem is $O(p\log(p))$. In the process of solving the zero-finding problem, $\sum_{i\in\mathcal{A}} c_i f_i$ and $\sum_{i\in\mathcal{A}} f_i^2$ are computed as the intermediate results. Given them as well as the $\frac{1}{\alpha}$

[3] They can also be used by the L_1-regularized L_2-SVM solver.

[4] For dense data $p = n$, for sparse data usually $p \ll n$.

[5] Usually, $q \ll m$.

```
Input: $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{y} \in \mathbb{R}^n$, $\lambda_1$, $\lambda_2$, $\boldsymbol{\theta}_1 \in \mathbb{R}^n$.
Output: $\mathcal{L}$, the retained feature list.
$\mathbb{L} = \emptyset$, $i = 1$, $\mathbf{Y} = diag\,(\mathbf{y})$;
for $i \leq m$ do
    $\hat{\mathbf{f}} = \mathbf{Y}\mathbf{f}_i$;
    $m{=}\left|\mathbf{c}^\top \hat{\mathbf{f}}\right| + l\left\|\hat{\mathbf{f}}\right\|_2$;
    if $m < 1$ then
        continue;
    end
    $m_1$=neg_min($\hat{\mathbf{f}}$), $m_2$=neg_min($-\hat{\mathbf{f}}$);
    $m = \max\{m_1, m_2\}$;
    if $m \geq 1$ then
        $\mathcal{L} = \mathcal{L} \cup \{i\}$;
    end
    $i = i + 1$;
end
return $\mathcal{L}$;

Function neg_min($\hat{\mathbf{f}}$)
    $\left\{\frac{1}{\alpha}, \mathcal{A}\right\}$ = zero_finding($\hat{\mathbf{f}}$);
    $m = \sum\limits_{i\in\mathcal{A}} c_i f_i - \frac{1}{\alpha} \sum\limits_{i\in\mathcal{A}} f_i^2$;
    return $-m$;
end

Function zero_finding($\hat{\mathbf{f}}$)
    $\mathcal{B} = \{i : (c_i > 0, f_i > 0)$ or $(c_i < 0, f_i \leq 0)\}$;
    $search = true$, $\frac{1}{\alpha} = max\left(\frac{c_j}{f_j}, j \in \mathcal{B}\right)$;
    compute $\mathcal{A}$ and $g\left(\frac{1}{\alpha}\right)$;
    if $g\left(\frac{1}{\alpha}\right) < 0$ then
        compute $\frac{1}{\alpha'}$ using Eq. (28);
        $\frac{1}{\alpha} = \frac{1}{\alpha'}$;
        return $\left\{\frac{1}{\alpha}, \mathcal{A}\right\}$;
    else
        $low = 0$, $high = \frac{1}{\alpha}$;
    end
    while $search$ do
        $\frac{1}{\alpha} = \frac{1}{2}\,(low + high)$;
        compute $\mathcal{A}$ and $g\left(\frac{1}{\alpha}\right)$, compute $\frac{1}{\alpha'}$ using Eq. (28);
        if the condition specified in Theroem 3 is satisfied then
            $\frac{1}{\alpha} = \frac{1}{\alpha'}$, $search = false$;
        else
            if $g\left(\frac{1}{\alpha}\right) < 0$ then
                $low = t_{j+1}$, $t_{j+1}$ is as defied in Theroem 3;
            else
                $high = t_j$, $t_j$ is as defied in Theroem 3;
            end
        end
    end
    return $\left\{\frac{1}{\alpha}, \mathcal{A}\right\}$;
end
```

Algorithm 1. Screening for L_1-regularized L_2-SVM

determined by zero finding, $\boldsymbol{\theta}^\top \hat{\mathbf{f}}$ can be computed in $O(1)$. Therefore, the total cost for computing the tighter bounds for q features is $O\big(n + qp \log(p)\big)$.

In summary, in the worst case of the proposed procedure, the total computational cost for screening a data set that has m features is $O\big(mp + qp \log(p)\big)$. And if $\hat{\mathbf{f}}^\top \boldsymbol{\theta}_1$ can be obtained from the intermediate results generated by the L_1-regularized L_2-SVM solver and $\mathbf{Yf}$, $\mathbf{f}^\top \mathbf{y}$, and $\mathbf{f}^\top \mathbf{f}$ are precomputed before training, the total cost can decrease to just $O\big(m + n + qp \log(p)\big)$.

4 Empirical Study

The screening approach presented in Algorithm 1 was implemented in the C language. This section evaluates its power for accelerating model selection for L_1-regularized L_2-SVM. Experiments are performed on a Windows Server 2008 R2 with two Intel Xeon® L5530 (2.40GHz) CPUs and 72GB memory.

4.1 Experiment Setup

Five benchmark data sets are used in the experiment. One is a microarray data set: gli_85. Three are text data sets: rcv1.binary(rcv1b), real-sim, and news20.binary (news20b). And one is a educational data mining data set: kdd2010 bridge-to-algebra (kddb). The gli_85 data set is downloaded from Gene Expression Omnibus,[6] and the other four data sets are downloaded from the LIBSVM data repository.[7] According to the feature-to-sample ratio (m/n), the five data sets fall into three groups: (1) the $m \gg n$ group, including the gli_85 and news20b data sets; (2) the $m \approx n$ group, including the rcv1b and kddb data sets; and (3) the $m \ll n$ group, including the real-sim data set. Table 1 shows detailed information about the five benchmark data sets.

Table 1. Summary of the benchmark data sets

Data Set	sample (n)	feature (m)	m/n
gli_85	85	22,283	262.15
rcv1b	20,242	47,236	2.33
real-sim	72,309	20,958	0.29
news20b	19,996	1,355,191	67.77
kddb	19,264,097	29,890,095	1.55

A L_1-regularized L_2-SVM solver based on the coordinate gradient descent (cgd) algorithm [18] is implemented in the C language for training the L_1-regularized L_2-SVM model. A similar solver is also implemented in the liblinear

[6] www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4412

[7] www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

package [6]. The difference is that in liblinear, the bias term b is also penalized by the L_1 regularizer and is inactive in most cases. In contrast, the solver that is implemented for this paper solves the problem specified in Eq. (1) exactly. Therefore, the bias term is not penalized and is alway active.

For each benchmark data set, the L_1-regularized L_2-SVM solver is used to fit model along a sequence of 20 λ values: $\left\{\lambda_k = \frac{1}{k}\lambda_{max} - \epsilon, k = 1, \ldots, 20, \epsilon = 10^{-8}\right\}$. When $\lambda = \lambda_{max} - \epsilon$, only one feature is active. Denote n_+ and n_- as the number of positive and negative samples, respectively. And let $\mathbf{m} = \sum_{i=1}^{n}\left(y_i - \frac{n_+ - n_-}{n}\right)\mathbf{x}_i$. This feature corresponds to the largest element in $\mathbf{m}$.

For each given benchmark data set, the L_1-regularized L_2-SVM solver runs in four different configurations: (1) In **org**, the solver runs without any accelerating technique. (2) In **warm**, the solver runs with warm-start. In the kth iteration, the $\mathbf{w}_{k-1}$ obtained in the $(k-1)$th iteration is used as the initial $\mathbf{w}_k$ for fitting model. When λ_k and λ_{k-1} are close, warm-start can effectively speed up training by reducing the number of iterations for the solver to converge. (3) In **scr**, the solver runs with the screening technique. (4) In **warm_scr**, the solver runs with both warm-start and the screening technique. Both warm-start and screening can be used to speed up model selection. The main purpose of running the L_1-regularized L_2-SVM solver with different configurations is not only to compare screening with warm-start, but also to provide a sensitivity study to explore that whether better performance can be achieved by combining two techniques.

Table 2. Total run time (in sec.) of the L_1-regularized L_2-SVM solver when different combinations of accelerating techniques are used to speed up model selection.

Alg.	gli_85	rcv1b	real-sim	news20b	kddb
org	284.08	19.04	20.73	1040.22	9071.73
warm	259.20	11.54	14.06	786.44	5770.12
scr	1.89	4.09	8.53	25.97	947.01
warm_scr	**1.83**	**2.70**	**5.90**	**18.22**	**643.34**

Table 3. Total number of iterations for the L_1-regularized L_2-SVM solver to converge when different combinations of accelerating techniques are used

Alg.	gli_85	rcv1_trainb	real-sim	news20b	kddb
org	16,176	1004	548	2,501	737
warm	14,772	578	361	1,908	483
scr	16,028	995	591	2,857	809
warm_scr	15,227	606	369	2,035	499

4.2 Results

Table 2 and Table 3 show the results of the total run time and the total number of iterations for the L_1-regularized L_2-SVM solver to converge when different combinations of accelerating techniques are used. The total run time and total number of iterations are obtained by aggregating the time and number of iterations used by the L_1-regularized L_2-SVM solver when it fits models using different regularization parameters. In terms of total running time, screening with warm-start (**warm_scr**) provides the best performance. Compared to **org**, for the $m \gg n$ group, the speed-up ratio is about 155.5 for the gli_85 data and 57.1 for the news20b data. For the $m \approx n$ group, the speed-up ratio is about 7.1 for the rcv1b data and 14.1 for the kddb data. And for the $m \ll n$ group, the speed-up ratio is about 3.5 for the real-sim data. The result shows that **warm_scr** is more effective when the number of features is larger than the number of samples. A similar trend is observed on **scr**. In terms of the total iteration number, the best performance is achieved by **warm** and **warm_scr**. This suggests that warm-start can effectively speed up convergence by providing a good start point for optimization. When **org** is compared to **scr**, the result suggests that the proposed screening technique can significantly improve the efficiency of the L_1-regularized L_2-SVM solver. This justifies that screening can effectively reduce the computational cost of training by removing most inactive features.

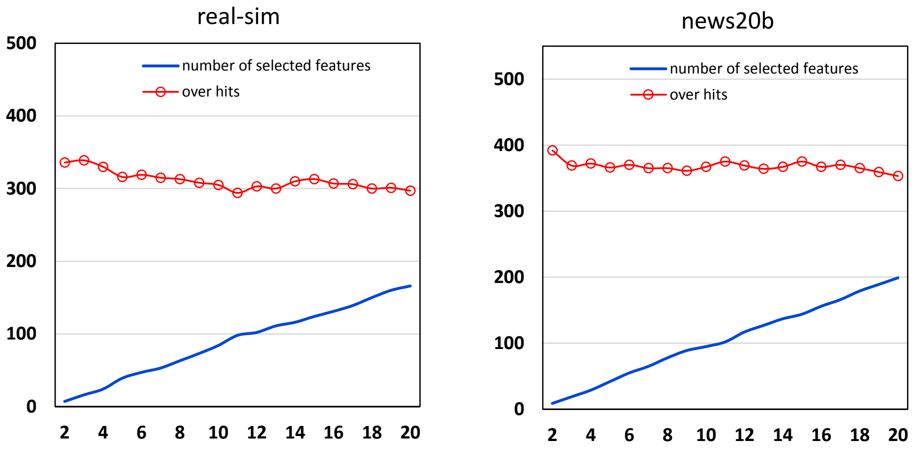

Fig. 1. Detailed information about the "over hits" on two benchmark data sets when λ decreases from λ_{max} to $\frac{1}{20}\lambda_{max}$. "Over hits" is the number of inactive features that are not removed by screening. The results show that the number of leftover inactive features is stable, and is small when compared to the size of the original feature set.

Figure 1 shows detailed information about the number of leftover inactive features on the real-sim and news20b data sets when λ decreases from λ_{max} to $\frac{1}{20}\lambda_{max}$. The result shows that this number is very stable during model selection. Let k be the number of active features. The proposed screening technique keeps to retain about $k + 400$ features for training the L_1-regularized L_2-SVM model. This number is much smaller than the dimensionality of the original data sets. Similar trends are also observed on other data sets and are not presented in this paper because of the space limit. Table 4 compares the time used by screening to the time used by training. Compared to training time, the screening time is marginal, especially when $m \gg n$. Notice that for training, the solver uses only the features that are selected by screening. The training time can be much longer if screening is not used to eliminate inactive features.

Table 4. Comparison of screening to training time. For training, the solver uses only the features that are selected by the proposed screening technique. The training time can be much longer if screening is not used to eliminate inactive features.

Tech.	gli_85	rcv1b	real-sim	news20b	kddb
scr					
scr	0.01	0.73	1.79	1.29	35.29
tr	1.89	3.35	6.74	24.68	911.72
ratio	0.01	0.22	0.27	0.05	0.04
warm_scr					
scr	0.03	0.75	1.75	1.31	34.93
tr	1.79	1.95	4.15	16.91	608.41
ratio	0.02	0.38	0.42	0.08	0.06

The results indicate that the proposed screening technqiue is effective for removing inactive features. And with warm-start they form a powerful combination for accelerating model selection for the L_1-regularized L_2-SVM.

5 Conclusion

Screening is an effective technique for accelerating model selection for L_1-regularized sparse learning model by eliminating features that are guaranteed to be inactive. This paper proposes a novel technique to screen features for L_1-regularized L_2-SVM by bounding $|\hat{\mathbf{f}}^\top \boldsymbol{\theta}|$ on a tight convex set formed by the interaction of an n-dimensional hyper-ball and the upper half-space. An efficient binary search algorithm is designed and implemented to compute this bound for features. Empirical study shows that the proposed technique can greatly improve model selection efficiency by stably eliminating a large portion of the inactive features. Our ongoing work will extend the technique to screen features for the L_1-regularized L_1-SVM model and provide support for distributed computing in a massively parallel processing (MPP) environment.

Acknowledgments. The authors would like to thank Anne Baxter, Russell Albright, and the anonymous reviewers for their valuable suggestions to improve this paper.

References

[1] Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Boston (1998)

[2] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. JMLR 3, 1157–1182 (2003)

[3] Bradley, P.S., Mangasarian, L.O.: Feature selection via concave minimization and support vector machines. In: ICML (1998)

[4] Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. In: NIPS (2003)

[5] Bi, J., Embrechts, M., Breneman, C.M., Song, M.: Dimensionality reduction via sparse support vector machines. JMLR 3, 1229–1243 (2003)

[6] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. JMLR 9, 1871–1874 (2008)

[7] Weston, J., Elisseff, A., Schoelkopf, B., Tipping, M.: Use of the zero norm with linear models and kernel methods. JMLR 3, 1439–1461 (2003)

[8] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)

[9] Li Wang, M.T., Tsang, I.W.: Learning sparse svm for feature selection on very high dimensional datasets. In: ICML (2010)

[10] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B 58, 267–288 (1996)

[11] Ghaoui, L., Viallon, V., Rabbani, T.: Safe feature elimination in sparse supervised learning. Pacific Journal of Optimization 8, 667–698 (2012)

[12] Wang, J., Lin, B., Gong, P., Wonka, P., Ye, J.: Lasso screening rules via dual polytope projection. In: NIPS (2013)

[13] Zhen, J.X., Hao, X., Peter, J.R.: Learning sparse representations of high dimensional data on large scale dictionaries. In: NIPS (2011)

[14] Liu, J., Zhao, Z., Wang, J., Ye, J.: Safe screening with variational inequalities and its applicaiton to lasso. arXiv:1307.7577 (2013)

[15] Tibshirani, R., Bien, J., Friedman, J.H., Hastie, T., Simon, N., Taylor, J., Tibshirani, R.J.: Strong rules for discarding predictors in lasso-type problems. Journal of the Royal Statistical Society: Series B 74, 245–266 (2012)

[16] Lions, J.L., Stampacchia, G.: Variational inequalities. Communications on Pure and Applied Mathematics 20(3), 493–519 (1967)

[17] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)

[18] Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Mathematical Programming 117, 387–423 (2009)