Safe and Efficient Screening For Sparse Support Vector Machine

Zheng Zhao, Jun Liu, James Cox
SAS Institute Inc. 600 Research Drive, Cary, NC 27513
{zheng.zhao, jun.liu, james.cox}@sas.com

ABSTRACT
Sparse support vector machine (SVM) is a robust predictive model that can effectively remove noise and preserve signals. Like Lasso, it can efficiently learn a solution path based on a set of predefined parameters and therefore provides strong support for model selection. Sparse SVM has been successfully applied in a variety of data mining applications including text mining, bioinformatics, and image processing. The emergence of big-data analysis poses new challenges for model selection with large-scale datasets that consist of tens of millions samples and features. In this paper, a novel screening technique is proposed to accelerate model selection for millions of samples and features. This technique can precisely identify inactive features in the optimal solution and thereby provide strong support for model selection. Sparse SVM has been successfully applied in a variety of data mining applications including text mining, bioinformatics, and image processing. The key idea is that, given a series of regularized parameters, \(\lambda > 0\), the corresponding solutions, \(w_1, w_2, \ldots, w_k\), can be obtained and the best solution 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. Big-data analysis requires a higher standard of efficiency for predictive modeling. When data are huge, the computational cost of model selection can be prohibitive. An intuitive question is to ask whether the solution obtained in the \(k\)th step of model selection can be used in the \((k+1)\)th step to speed up computation. For Lasso [11], the answer leads to the state-of-the-art screening techniques to accelerate model selection [17, 7, 14, 12, 10]. The key idea is that, given a solution \(w_1^*\) for \(\lambda = \lambda_1\), one can identify many features that are guaranteed to have zero coefficients in \(w_2^*\) when \(\lambda = \lambda_2\). By removing a large number of these inactive features, the cost for computing \(w_2^*\) can be greatly reduced.

1. INTRODUCTION
Sparse predictive modeling algorithms provide powerful tools to analyze high-dimensional data and generate results that have high-degree of interpretability and robustness [5, 11]. In general, an \(\ell_1\)-regularized sparse predictive modeling algorithm can be formulated as \(\min_{w} \text{loss}(w) + \lambda \|w\|_1\). Here \(w \in \mathbb{R}^m\) contains the model coefficients, \(\text{loss}(w)\) is a loss function, and \(\lambda \geq 0\) is the regularization parameter that balances between the loss function and the regularizer. When the hinge loss or its square form is used as the loss function, the resulting sparse model is the \(\ell_1\)-regularized support vector machine (SVM) [4, 18, 2, 6, 16]. An \(\ell_1\)-regularized SVM model can simultaneously perform model fitting by margin maximization and remove noisy features by soft-thresholding. It has been successfully applied in a variety of data mining applications that include text mining, bioinformatics, and image processing. Compared to other variances of sparse SVM model [15, 8, 1], \(\ell_1\)-regularized SVM enjoys two major advantages. First, it defines a convex problem; therefore, an optimal solution can always be achieved without any relaxation of the original problem. Second, its optimization is simple, and a well implemented \(\ell_1\)-regularized SVM solver can readily handle large-scale problems with tens of millions samples and features [6].

The value of the regularization parameter \(\lambda\) is crucial to the performance of an \(\ell_1\)-regularized SVM. To achieve good performance, model selection is often used to help choose an appropriate \(\lambda\) value. For example, given a series of regularization parameters, \(\lambda_1 > \lambda_2 > \ldots > \lambda_k\), the corresponding solutions, \(w_1, w_2, \ldots, w_k\), can be obtained and the best solution 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. Big-data analysis requires a higher standard of efficiency for predictive modeling. When data are huge, the computational cost of model selection can be prohibitive. An intuitive question is to ask whether the solution obtained in the \(k\)th step of model selection can be used in the \((k+1)\)th step to speed up computation. For Lasso [11], the answer leads to the state-of-the-art screening techniques to accelerate model selection [17, 7, 14, 12, 10]. The key idea is that, given a solution \(w_1^*\) for \(\lambda = \lambda_1\), one can identify many features that are guaranteed to have zero coefficients in \(w_2^*\) when \(\lambda = \lambda_2\). By removing a large number of these inactive features, the cost for computing \(w_2^*\) can be greatly reduced.

Although screening algorithms have been designed for Lasso, very little research has been done for screening for \(\ell_1\)-regularized SVMs except in [7], which proposes a safe screening tech-
nique for $\ell_1$-regularized $\ell_2$-SVM. This paper presents a novel screening technique that is designed to speed up model selection for $\ell_1$-regularized $\ell_2$-SVM. The technique makes use of the variational inequality [9] for constructing a tight convex set, which can be used to compute bounds for screening features. Features that are removed by this technique are guaranteed to be inactive in the optimal solution. Therefore, the screening is “safe.” Experimental results on five high-dimensional benchmark data sets demonstrate that the proposed screening technique can dramatically speed up model selection for $\ell_1$-regularized $\ell_2$-SVM by efficiently removing a large number of inactive features.

2. $\ell_1$-REGULARIZED $\ell_2$-SVM

Assume that $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a data set that contains $n$ samples, $\mathbf{X} = (x_1, \ldots, x_n)$, and $m$ features, $\mathbf{X} = (f^1, \ldots, f^m)$ . Assume also that $y = (y_1, \ldots, y_n)$ contains $n$ class labels, $y_i \in \{-1,+1\}$, $i = 1, \ldots, n$. Let $\mathbf{w} \in \mathbb{R}^m$ be the $m$-dimensional weight vector, let $\xi_i \geq 0$, $i = 1, \ldots, n$ be the $n$ slack variables, and let $b \in \mathbb{R}$ and $\lambda \in \mathbb{R}^+$ be the bias and the regularization parameter, respectively. The primal form of the $\ell_1$-regularized $\ell_2$-SVM is defined as:

$$\min_{\mathbf{w}} \frac{1}{2} \sum_{i=1}^{n} \xi_i^2 + \lambda ||\mathbf{w}||_1,$$

s.t. $y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$.

Eq. (1) specifies a convex problem that has a non-smooth $\ell_1$ regularizer, which enforces that the solution is sparse. Let $\mathbf{w}^*(\lambda)$ be the optimal solution of Eq. (1) for a given $\lambda$. All the features that have nonzero values in $\mathbf{w}^*(\lambda)$ are called active features, and the other features are called inactive. Let $\alpha \in \mathbb{R}^n$ be the $n$-dimensional dual variable. By applying the Lagrangian multiplier [3], the dual of the problem defined in Eq. (1) can be obtained as:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \alpha_i y_i^2,$$

s.t. $||\mathbf{f}^\top \alpha|| \leq \lambda$, $j = 1, \ldots, m,$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$, $\alpha \succeq 0$.

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

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{n} \theta_i^2,$$

s.t. $||\mathbf{f}^\top \theta|| \leq 1$, $j = 1, \ldots, m,$

$$\sum_{i=1}^{n} \theta_i y_i = 0$, $\theta \succeq 0$.

In the primal formulation for the $\ell_1$-regularized $\ell_2$-SVM, the primal variables are $b$, $\mathbf{w}$, and $\xi$. And in the dual formulation, the dual variables are $\alpha$ or $\theta$. When $b$ and $\mathbf{w}$ are known, $\alpha$, $\xi$, and $\theta$ can be obtained as:

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

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

$$\alpha^\top \mathbf{f}_j = \begin{cases} \text{sign}(w_j) \lambda, & \text{if } w_j \neq 0, \\ [-\lambda, +\lambda], & \text{if } w_j = 0 \end{cases}, \; j = 1, \ldots, m.$$ (5)

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

$$\theta^\top \mathbf{f}_j = \begin{cases} \text{sign}(w_j) \lambda, & \text{if } w_j \neq 0, \\ [-\lambda, +\lambda], & \text{if } w_j = 0 \end{cases}, \; j = 1, \ldots, m.$$ (6)

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

$$\lambda_{\text{max}} = \frac{1}{||y_i - \frac{n_+ - n_-}{n} x_i||_\infty},$$ (7)

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

$$\mathbf{w} = 0, \; b^* = \frac{(n_+ - n_-)}{n}.$$ (8)

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. EFFICIENT AND SAFE SCREENING FOR $\ell_1$-REGULARIZED $\ell_2$-SVM

Eq. (6) shows that the necessary condition for a feature $f$ to be active in an optimal solution is $|\theta^\top f| = 1$. On the other hand, for any feature $f$, if $|\theta^\top f| < 1$, it must be inactive in the optimal solution. Given a $\lambda$ value, this condition can be used to develop a screening rule for removing inactive features to speed up training for the $\ell_1$-regularized $\ell_2$-SVM. The key is to compute the upper bound of $|\theta^\top f|$ for features. A feature can be safely removed if its upper bound value is less than 1. The cost of computing the upper bound can be much lower than training $\ell_1$-regularized $\ell_2$-SVM. Therefore, screening can greatly lower the computational cost by removing many inactive features before training. To bound the value of $|\theta^\top f|$, it is necessary to construct a closed convex set $\mathbf{K}$ that contains $\theta$. The upper bound value can be then computed by maximizing $|\theta^\top f|$ over $\mathbf{K}$.

3.1 Constructing the Convex Set $\mathbf{K}$

In the following, Eq. (3) and the variational inequality [9] are used to construct a closed convex set $\mathbf{K}$ to bound $|\theta^\top f|$. Proposition 3.1 introduces the variational inequality for a convex optimization problem.

**PROPOSITION 3.1.** Let $\theta^*$ be an optimal solution of the following convex optimization problem:

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

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

$$\nabla g(\theta^*)^\top (\theta - \theta^*) \geq 0, \; \forall \theta \in \mathbf{K}.$$
Let $\theta_1$ and $\theta_2$ be the optimal solutions of the problem defined in Eq. (3) for $\lambda_1$ and $\lambda_2$, respectively. Assume that $\lambda_1 > \lambda_2$ and that $\theta_1$ is known. The following results can be obtained by applying Proposition 3.1 to the convex problem defined in Eq. (3) for $\theta_1$ and $\theta_2$, respectively:

\[
\begin{align*}
(\theta_1 - \frac{1}{\lambda_1})^\top (\theta - \theta_1) & \geq 0, \\
(\theta_2 - \frac{1}{\lambda_2})^\top (\theta - \theta_2) & \geq 0.
\end{align*}
\]

By substituting $\theta = \theta_2$ into Eq. (9), and $\theta = \theta_1$ into Eq. (10), the following equations can be obtained:

\[
\begin{align*}
(\theta_1 - \frac{1}{\lambda_1})^\top (\theta_2 - \theta_1) & \geq 0, \\
(\theta_2 - \frac{1}{\lambda_2})^\top (\theta_2 - \theta_1) & \leq 0.
\end{align*}
\]

In the preceding equations, $\theta_1$, $\lambda_1$, and $\lambda_2$ are known. Therefore, Eq. (11) defines an $n$-dimensional halfspace and Eq. (12) defines an $n$-dimensional hyperball. Because $\theta_2$ needs to satisfy both equations, it must reside in the intersection of the halfspace and the hyperball. Obviously, this region is a closed convex set, and it can be used as $\mathbf{K}$ to bound $\theta_2$. Fig. 1 shows an example of the $\mathbf{K}$ in a two-dimensional space. In the figure, Eq. (11) defines the area below the blue line, Eq. (12) defines the area in the red circle, and $\mathbf{K}$ is indicated by the shaded area.

Besides the $n$-dimensional hyperball defined in Eq. (12), it is possible to derive a series of hyperballs by combining Eq. (11) and Eq. (12). Assume that $\theta^*$ is the optimal solution of Eq. (3) and $t \geq 0$. It is easy to verify that $\theta^*$ is also the optimal solution of the following problem:

\[
\min_{\theta} \left\| \theta - \left( \frac{1}{\lambda} + (1 - t) \theta^* \right) \right\|_2^2,
\]

\[\text{s.t. } |\hat{y}_j^\top \theta| \leq 1, \ j = 1, \ldots, m,
\]

\[
\sum_{i=1}^n \theta_i y_i = 0, \ \theta \neq 0.
\]

By applying Proposition 3.1 to the problem defined in Eq. (13) for $\theta_1$, and $\theta_2$, the following results can be obtained:

\[
\begin{align*}
(\theta_1 - \left( t_1 \frac{1}{\lambda_1} + (1 - t_1) \theta_1 \right))^\top (\theta - \theta_1) & \geq 0, \\
(\theta_2 - \left( t_2 \frac{1}{\lambda_2} + (1 - t_2) \theta_2 \right))^\top (\theta - \theta_2) & \geq 0.
\end{align*}
\]

Let $t = \frac{t_1}{t_2} \geq 0$. By substituting $\theta = \theta_2$ and $\theta = \theta_1$ into Eq. (14) and Eq. (15), respectively, and then combining the two inequalities, the following equations can be obtained:

\[
\mathbf{B}_t = \left\{ \theta_2 : (\theta_2 - \mathbf{c})^\top (\theta_2 - \mathbf{c}) \leq t^2 \right\},
\]

\[
c = \frac{1}{2} \left( t_1 \frac{1}{\lambda_1} + t_2 \frac{1}{\lambda_2} + \theta_1 + \theta_2 \right), t = \frac{1}{2} \left| t_1 \frac{1}{\lambda_1} + \frac{1}{\lambda_2} - \theta_1 \right|_2.
\]

As the value of $t$ changes from 0 to $\infty$, Eq. (16) generates a series of hyperballs. When $t = 0$, $\mathbf{c} = \frac{1}{2} \left( \frac{1}{\lambda_1} + \theta_1 \right)$ and $t = \frac{1}{2} \left| \frac{1}{\lambda_2} - \theta_1 \right|_2$. This corresponds to the hyperball defined by Eq. (12). The following theorems provide some insights about the properties of the hyperballs generated by Eq. (16):

**Theorem 3.2.** Let $\mathbf{a} = \frac{1}{\frac{1}{\lambda_1} - \theta_1}$. The radius of the hyperball generated by Eq. (16) reaches its minimum when

\[
t = 1 + \left( \frac{1}{\frac{1}{\lambda_2} - \frac{1}{\lambda_1}} \right) \left\| \frac{1}{\lambda_1} - \theta_1 \right\|_2.
\]

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

\[
\hat{\mathbf{c}} = \frac{1}{2} \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) P_a (1) + \theta_1, \ l = \frac{1}{2} \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \| P_a (1) \|.
\]

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

**Proof.** The theorem can be proved by minimizing the $r$ defined in Eq. (16).

**Theorem 3.3.** Let the intersection of the hyperplane $\left( \theta_1 - \frac{1}{\lambda_1} \right)^\top (\theta_2 - \theta_1) = 0$ and the hyperball defined by Eq. (16) be $\mathbf{P}_t$. The following equation holds:

\[
\mathbf{P}_t \mathbf{v}_1 = \mathbf{P}_t \mathbf{v}_2, \text{ for } \forall t_1, t_2 \geq 0, t_1 \neq t_2.
\]

**Proof.** The theorem can be proved by showing that $\mathbf{P}_t$ is independent of $t$.

This theorem shows that the intersection between the hyperball $\mathbf{B}_t$ and the hyperplane $(\theta_1 - \frac{1}{\lambda_1})^\top (\theta_2 - \theta_1) = 0$ is the same for different $t$ values.
Theorem 3.4. Let the intersection of the halfspace $(\theta_1 - \frac{1}{\lambda_1})^\top (\theta_2 - \theta_1) \geq 0$ and the hyperball defined by Eq. (16) be $Q_t$. The following inequality holds:

$$Q_{t_1} \subseteq Q_{t_2}, \text{ for } \forall t_1, t_2 \geq 0, t_1 \leq t_2.$$ 

Proof. The theorem can be proved by showing that $\forall t_1, t_2 \geq 0$ and $t_1 \leq t_2$, if $\theta_2 \in Q_{t_1}$, then $\theta_2 \in Q_{t_2}$ also holds.

This theorem shows that the volume of $Q_t$ becomes larger when $t$ becomes larger. And $Q_{t_1} \subseteq Q_{t_2}$ if $t_1 \leq t_2$.

Fig. 1 shows two circles in a 2D space. The red circle corresponds to the one obtained by setting $t_1 = 0$ in Eq. (16). The blue circle corresponds to the one obtained by setting $t_2 = 1 + \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \frac{a^\top 1}{\| \frac{1}{\lambda_2} - \theta_1 \|^2}$

in Eq. (16). It can be observed that the intersections of the two circles pass the line $(\theta_1 - \frac{1}{\lambda_1})^\top (\theta_2 - \theta_1) = 0$. This is consistent with Theorem 3.3. Also since $t_1 \leq t_2$, $Q_{t_1} \subseteq Q_{t_2}$, which is consistent with Theorem 3.4.

Theorem 3.4 suggests that $Q_{t=0}$ should be used to construct $K$, because when $t = 0$, the volume of $Q_t$ is minimized. The equality $\theta^\top y = 0$ in Eq. (3) of the dual formulation can also be used to further reduce the volume of $K$.

$$K = \left\{ \theta_2 : (\theta_2 - c)^\top (\theta_2 - c) \leq t^2, \right\}$$

$$\left( \theta_1 - \frac{1}{\lambda_1} \right)^\top (\theta_2 - \theta_1) \geq 0, \quad \theta_2^\top y = 0 \}.$$ 

Here $c = \frac{1}{2} \left( \frac{1}{\lambda_2} + \theta_1 \right)$ and $t = \frac{1}{2} \frac{a^\top 1}{\| \frac{1}{\lambda_2} - \theta_1 \|^2}$. Let $\theta_2 = c + r$, $a = \frac{\lambda_2 - \theta_1}{\| \lambda_2 - \theta_1 \|^2}$, and $b = \frac{1}{2} \left( \frac{1}{\lambda_2} - \theta_1 \right)$. $K$ can be written as:

$$K = \left\{ \theta_2 : \theta_2 = c + r, \quad \| r \|^2 \leq \| b \|^2, \right\}$$

$$a^\top (b + r) \leq 0, \quad (c + r)^\top y = 0 \}.$$ 

Theorem 3.3 shows that when $t$ varies, the intersection of the hyperball $B_1$, and the hyperplane $(\theta_1 - \frac{1}{\lambda_1})^\top (\theta_2 - \theta_1) = 0$ remains unchanged. This suggests that if the maximum value of $\theta^\top \hat{f}$ is achieved with a $\theta$ that is in this area, no matter which $B_1$ is used, the maximum value will be the same. This property can be used to simplify the computation. Section 3.2.4 will show that when the maximum value of $\theta^\top \hat{f}$ is achieved with a $\theta$ on the intersection of the hyperball $B_{t=0}$ and the hyperplane $(\theta_1 - \frac{1}{\lambda_1})^\top (\theta_2 - \theta_1) = 0$, the computation of the maximum value can be simplified by switching to $B_t$ with $t = 1 + \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \frac{a^\top 1}{\| \frac{1}{\lambda_2} - \theta_1 \|^2}$.

3.2 Computing the Upper Bound

Given the convex set $K$ defined in Eq. (17), the maximum value of $\theta^\top \hat{f}$ can be computed by solving the problem:

$$\max \left| (c + r)^\top \hat{f} \right|$$ 

$$\text{s.t. } a^\top (b + r) \leq 0, \| r \|^2 - \| b \|^2 \leq 0, \quad (c + r)^\top y = 0.$$ 

In Eq. (18), $c = c + r$. Also, $r$ is unknown, and $\hat{f}$, $a$, $b$, $c$, and $y$ are known. Since the following equation holds:

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

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

$$\max \left| (c + r)^\top \hat{f} \right|$$ can be decomposed to two subproblems:

$$m_1 = - \min \theta_2^\top \hat{f} = - \min r^\top \hat{f} - c^\top \hat{f}$$

$$\text{s.t. } a^\top (b + r) \leq 0, \| r \|^2 - \| b \|^2 \leq 0, \quad (c + r)^\top y = 0,$$ 

$$m_2 = \max \theta_2^\top \hat{f} = - \min r^\top (-\hat{f}) + c^\top \hat{f}$$

$$\text{s.t. } a^\top (b + r) \leq 0, \| r \|^2 - \| b \|^2 \leq 0, \quad (c + r)^\top y = 0,$$ 

and

$$\max \left| \theta_2^\top \hat{f} \right| = \max \left| (c + r)^\top \hat{f} \right| = \max (m_1, m_2).$$

Eq. (19) and Eq. (20) suggest that the key to bound $\theta_2^\top \hat{f}$ is to solve the following problem:

$$\min r^\top \hat{f}$$

$$\text{s.t. } a^\top (b + r) \leq 0, \| r \|^2 - \| b \|^2 \leq 0, \quad (c + r)^\top y = 0.$$ 

Its Lagrangian $L(r, \alpha, \beta, \rho)$ can be written as:

$$r^\top \hat{f} + \alpha^\top (b + r) + \frac{1}{2} \beta (\| r \|^2 - \| b \|^2) + \rho (c + r)^\top y.$$ 

And the Karush-Kuhn-Tucker (KKT) conditions are:

$$\begin{align*}
\text{(dual feasibility)} & & \alpha \geq 0, \beta \geq 0, \\
\text{(primal feasibility)} & & \| r \|^2 - \| b \|^2 \leq 0, \\
\text{(complementary slackness)} & & \alpha^\top (b + r) \leq 0, \\
\text{(stationarity)} & & \nabla_r L(r, \alpha, \beta, \rho) = 0.
\end{align*}$$

Since $\| r \|^2 \leq \| b \|^2$, the problem specified in Eq. (21) is bounded from below by $-\| b \|^2 / ||\hat{f}||$. Thus, min$_r L(r, \alpha, \beta, \rho)$ is also bounded from below.

According to whether the inequality constraints are active, the problem can have different minimum values. This requires a discussion of the following four different cases:

$$\begin{align*}
& \text{(1)} \beta = 0, \quad \hat{f} + \alpha a + \rho y \neq 0; \\
& \text{(2)} \beta = 0, \quad \hat{f} + \alpha a + \rho y = 0; \\
& \text{(3)} \beta > 0, \alpha = 0; \\
& \text{(4)} \beta > 0, \alpha > 0.
\end{align*}$$

The following sections study these cases in detail.

3.2.1 The Case $\beta = 0, \hat{f} + \alpha a + \rho y \neq 0$

In this case, set $r = t (\hat{f} + \alpha a + \rho y)$, and let $t \to -\infty$. Then $L(r, \alpha, 0, \rho) \to -\infty$. This contradicts the observation that min$_r L(r, \alpha, \beta, \rho)$ must be bounded from below. So when $\hat{f} + \alpha a + \rho y \neq 0$, $\beta$ must be positive.

3.2.2 The Case $\beta = 0, \hat{f} + \alpha a + \rho y = 0$

Let $P_u(v) = v - \frac{v^\top u}{u^\top u} u$ be the projection of $v$ onto the null-space of $u$. Given $\hat{f} + \alpha a + \rho y = 0$, it is easy to verify that $\alpha P \hat{f}(a) = -P \hat{f}(\hat{f})$. This suggests that $\alpha P \hat{f}(a)$ and $P \hat{f}(\hat{f})$ are colinear. Also since $\alpha \geq 0$, the following must
Given $\alpha P_y(a) = -P_y(\hat{f})$, $\alpha$ can be computed by:

$$\alpha = -\frac{\|P_y(\hat{f})\|_2}{\|P_y(a)\|_2^2} \frac{P_y(a)^T P_y(\hat{f})}{\|P_y(a)\|_2 \|P_y(\hat{f})\|_2}.$$

Similarly, the value of $\rho$ can be computed by:

$$\rho = -\frac{\hat{f}^T y}{\|y\|_2^2} - \alpha \frac{a^T y}{\|y\|_2^2} = -\frac{\|P_y(\hat{f})\|_2 a^T y}{\|P_y(a)\|_2 \|y\|_2^2}.$$

By plugging $\beta = 0$ and the obtained value of $\alpha$ and $\rho$ into Eq. (22), $L(r, \alpha, 0, \rho)$ can be written as:

$$L(r, \alpha, 0, \rho) = -\frac{\|P_y(\hat{f})\|_2}{\|P_y(a)\|_2} a^T \theta_1 - c^T \hat{f}. \tag{30}$$

It can be verified that in this case, all KKT conditions specified in Eq. (23)–Eq. (29) are satisfied. Since the problem defined in Eq. (19) is convex and its domain is also convex, Eq. (30) defines the minimum value of the problem. The following theorem summarizes the result when $\beta = 0$.

**Theorem 3.5.** When

$$\frac{\partial \mu^y(a) \partial P_y(\hat{f})}{\partial \mu^y(a) \partial P_y(\hat{f})} = -1,$$

$r^T \hat{f}$ achieves its minimum value at $\beta = 0$, which can be computed as:

$$\min_r r^T \hat{f} = -\frac{\|P_y(\hat{f})\|_2}{\|P_y(a)\|_2} a^T \theta_1 - c^T \hat{f}.$$

And in this case, the value of the dual variables are:

$$\alpha = \frac{\|P_y(\hat{f})\|_2}{\|P_y(a)\|_2}, \quad \beta = 0, \quad \rho = -\frac{\hat{f}^T y}{\|y\|_2^2} - \frac{\|P_y(\hat{f})\|_2 a^T y}{\|P_y(a)\|_2 \|y\|_2^2}.$$

Since $\alpha = \frac{\|P_y(\hat{f})\|_2}{\|P_y(a)\|_2} > 0$, the minimum value is achieved on the hyperplane defined by $a^T (b + r) = 0$.

**Corollary 3.6.** When

$$\frac{\partial \mu^y(a) \partial P_y(\hat{f})}{\partial \mu^y(a) \partial P_y(\hat{f})} = 1,$$ $r^T \hat{f}$ achieves its maximum value at $\beta = 0$. In this case, $-\min \theta^T \hat{f}$ can be computed as:

$$-\min \theta^T \hat{f} = -\min r^T \hat{f} - c^T \hat{f} = \frac{\|P_y(\hat{f})\|_2}{\|P_y(a)\|_2} a^T \theta_1. \tag{31}$$

$$\max \theta^T \hat{f}$$ can be computed by replacing $\hat{f}$ with $-\hat{f}$ in Eq. (31).

In the computation, $\|P_y(\hat{f})\|_2$ is independent to $\lambda_1$, $\lambda_2$, and $\theta_1$. Therefore, it can be precomputed. $\|P_y(a)\|_2$ and $a^T \theta_1$ are shared by all features. These properties can be used to accelerate the computation.

**3.2.3 The Case: $\beta > 0$, $\alpha = 0$**

In this case, since $\beta > 0$ and $\alpha = 0$, the minimum value of $r^T \hat{f}$ is achieved on the boundary of the hyperball. In Figure 1, this corresponds to the arc of the red circle under the blue line. Plugging $\alpha = 0$ in Eq. (22) results in:

$$L(r, 0, \beta, \rho) = r^T \hat{f} + \frac{1}{2} \beta (\|r\|_2^2 - \|b\|_2^2) + \rho (c + r)^T y. \tag{32}$$

The dual function $g(0, \beta, \rho)$ can be obtained by setting

$$\nabla_r L(r, 0, \beta, \rho) = \nabla_r \hat{f} + \beta r + \rho y = 0 \Rightarrow r = -\frac{1}{\beta} (\hat{f} + \rho y).$$

Since $\beta > 0$, $\|b\|_2 = \|r\|_2$. Therefore $\beta$ can be written as:

$$\beta = \frac{\|\hat{f} + \rho y\|_2}{\|b\|_2}.$$

Plugging the obtained $r$ and $\beta$ into $L(r, 0, \beta, \rho)$ leads to:

$$g(\rho) = \min_r L(r, 0, \beta, \rho) = -\|b\|_2 \|\hat{f} + \rho y\|_2 + \rho \|y\|_2^2. \tag{33}$$

The dual function can be maximized by setting $\frac{\partial g(\rho)}{\partial \rho} = 0$. Also since $b^T y = c^T y$, the following equation holds:

$$-\|b\|_2 (\rho y^T \hat{f} + \hat{f}^T y) + b^T y = 0. \tag{34}$$

Squaring both sides of the equation and solving the obtained equation leads to the result:

$$\rho = \frac{-\hat{f}^T y + \|P_y(\hat{f})\|_2 b^T y}{\|y\|_2^2} \frac{\|P_y(\hat{f})\|_2 b^T y}{\|y\|_2^2}.$$

To obtain this equation, the following facts are used:

$$b^T b - \frac{(b^T y)^2}{y^T y} = \|b - \frac{b^T y}{y^T y} y\|_2^2 = \|P_y(b)\|_2^2.$$

$$\hat{f}^T \hat{f} - \frac{(\hat{f}^T y)^2}{y^T y} = \|\hat{f} - \frac{\hat{f}^T y}{y^T y} y\|_2^2 = \|P_y(\hat{f})\|_2^2.$$

Since $(c + r)^T y = 0$ and $r = -\frac{1}{\beta} (\hat{f} + \rho y)$, it can be verified that $\beta = \frac{\hat{f}^T y + \rho y^T \hat{f}}{c^T y}$. To ensure that $\beta$ is positive, the following equation must hold:

$$\rho = \frac{-\hat{f}^T y + \|P_y(\hat{f})\|_2 b^T y}{\|y\|_2^2} \frac{\|P_y(\hat{f})\|_2 b^T y}{\|y\|_2^2} \frac{(\hat{f}^T y)^2}{y^T y}.$$

And in this case, $\beta$ can be written in the form:

$$\beta = \frac{\|\hat{f} + \rho y\|_2}{\|b\|_2} \frac{\|P_y(\hat{f})\|_2}{\|y\|_2}.$$

To compute $\max g(\rho)$, first, Eq. (34) can be rewritten as:

$$\|b\|_2 \|\hat{f} + \rho y\|_2 = \|b\|_2^2 \frac{\|P_y(\hat{f})\|_2}{\|y\|_2} (\hat{f}^T y + \rho y^T \hat{f}) b^T y.$$ 

Plugging Eq. (35) and Eq. (37) into Eq. (33) leads to:

$$\max_{\rho} g(\rho) = -\|P_y(b)\|_2 \frac{\|P_y(\hat{f})\|_2}{\|y\|_2} - \|P_y(b)\|_2 \|P_y(\hat{f})\|_2 - \hat{f}^T y b^T y.$$

Since $\hat{f}^T y b^T y = \hat{f}^T b - \|P_y(b)\|_2 \|P_y(\hat{f})\|_2$, $\max g(\rho)$ can also be written in the following form:

$$\max_{\rho} g(\rho) = -\|P_y(b)\|_2 \frac{\|P_y(\hat{f})\|_2}{\|y\|_2} + P_y(b)^T P_y(\hat{f}) - \hat{f}^T b.$$

It can be verified that in this case, all the KKT conditions specified in Eq. (23)–Eq. (24) and Eq. (26)–Eq. (29) are satisfied. The additional condition for Eq. (25) to be satisfied...
can be derived as follows. First setting the derivative of Eq. (22) to be zero leads to the following equation:

$$r = -\frac{1}{\beta} \left( \hat{f} + \alpha a + \rho y \right)$$

Plugging this equation to $a^\top (b + r) \leq 0$ results in:

$$\alpha \geq \beta a^\top b - a^\top \hat{f} - \rho a^\top y.$$  

If $\beta a^\top b - a^\top \hat{f} - \rho a^\top y > 0$, $\alpha > 0$ must hold. According to the complementary slackness condition, $a^\top (b + r) = 0$. Therefore, $\alpha = \beta a^\top b - a^\top \hat{f} - \rho a^\top y > 0$. However, this contradicts the requirement that $\alpha = 0$. On the other hand, if $\beta a^\top b - a^\top \hat{f} - \rho a^\top y \leq 0$, $\alpha = 0$ must hold. Otherwise, $\alpha > 0$ and $\alpha = \beta a^\top b - a^\top \hat{f} - \rho a^\top y \leq 0$ form a contradiction. Therefore, to satisfy Eq. (25), the condition $\beta a^\top b - a^\top \hat{f} - \rho a^\top y \leq 0$ must be true. Plugging Eq. (35) and Eq. (36) in $\beta a^\top b - a^\top \hat{f} - \rho a^\top y \leq 0$ leads to:

$$\left\| P_y (\hat{f}) \right\|_2 P_y (a)^\top \left( P_y (b) \frac{P_y (\hat{f})}{\left\| P_y (b) \right\|_2} - P_y (\hat{f}) \frac{P_y (\hat{f})}{\left\| P_y (\hat{f}) \right\|_2} \right) \leq 0$$

Under this condition, the KKT condition $a^\top (b + r) \geq 0$ must be satisfied. The following theorem summarizes the result for the case $\beta > 0$ and $a = 0$.

**Theorem 3.7.** When $P_y (a)^\top \left( P_y (b) \frac{P_y (\hat{f})}{\left\| P_y (b) \right\|_2} - P_y (\hat{f}) \frac{P_y (\hat{f})}{\left\| P_y (\hat{f}) \right\|_2} \right) \leq 0$, $r^\top \hat{f}$ achieves its minimum value at $\beta > 0$ and $a = 0$.

**Corollary 3.8.** When $P_y (a)^\top \left( P_y (b) \frac{P_y (\hat{f})}{\left\| P_y (b) \right\|_2} - P_y (\hat{f}) \frac{P_y (\hat{f})}{\left\| P_y (\hat{f}) \right\|_2} \right) \leq 0$, $r^\top \hat{f}$ achieves its minimum value at $\beta > 0$ and $a = 0$. And in this case, $-\min \theta^\top \hat{f}$ can be computed as:

$$-\min \theta^\top \hat{f} = -\min r^\top \hat{f} - c^\top \hat{f} = \left\| P_y (b) \right\|_2 \left\| P_y (\hat{f}) \right\|_2 - P_y (b)^\top P_y (\hat{f}) - \hat{f}^\top \theta_1$$

max $\theta^\top \hat{f}$ can be computed by replacing $\hat{f}$ with $-\hat{f}$ in Eq. (40).

In the computation, $\left\| P_y (\hat{f}) \right\|_2$ does not rely on $\lambda_1$, $\lambda_2$ and $\theta_1$. Therefore, it can be precomputed. $P_y (a)^\top P_y (b)$ and $\left\| P_y (b) \right\|_2$, although rely on $\lambda_1$, $\lambda_2$ or $\theta_1$, are shared by all features and only need to be computed once. These properties can be used to accelerate computation.

3.2.4 The Case: $\beta > 0$, $a > 0$

In this case, the minimum value of $r^\top \hat{f}$ is achieved on the intersection of the boundary of the hyperball and the hyperplane. In Figure 1, this corresponds to the two points on the intersection of the red circle and the blue line. It turns out that when $\beta > 0$ and $a > 0$, deriving a closed form solution for the problem specified in Eq. (19) is difficult. Theorem 3.3 suggests that when the minimum value is achieved on the intersection of the hyperball and the hyperplane, one could switch the hyperball used in Eq. (19) to simplify the computation. It turns out that a closed form solution can be obtained by using the hyperball $B_1$ with $t = 1 + \left( \frac{1}{x_2} - \frac{1}{z_1} \right) \frac{a^\top}{\left\| a \right\|_2}$. This corresponds to the hyperball defined in Theorem 3.2. As proved in Theorem 3.3, the intersections of different $B_1$ and $\left( \frac{1}{x_2} - \theta_1 \right)^\top (\theta_1 - \theta_2) = 0$ are identical. Therefore, switching the hyperball $B_1$ does not change the maximum value of $\theta^\top \hat{f}$.

When $B_1$ with $t = 1 + \left( \frac{1}{x_2} - \frac{1}{z_1} \right) \frac{a^\top}{\left\| a \right\|_2}$ is used and assuming that the minimum is achieved on the intersection of the hyperball and the hyperplane, the problem specified in Eq. (19) can be rewritten as:

$$\operatorname{arg}_r \min r^\top \hat{f}$$

subject to $a^\top r = 0, \left\| r \right\|_2^2 - t^2 \leq 0, (\hat{c} + r)^\top y = 0$.

And its Lagrangian can be written as:

$$L (r, \alpha, \beta, \rho) = r^\top \hat{f} + a^\top r + \frac{\alpha}{2} \left( \left\| r \right\|_2^2 - t^2 \right) + \rho (\hat{c} + r)^\top y.$$  

In the preceding equation, $c$ is the center of the hyperfall, and $l$ is the radius of the hyperfall. They are defined as:

$$\hat{c} = \frac{1}{2} \left( \frac{1}{x_2} - \frac{1}{z_1} \right) P_a (1) + \theta_1, \quad l = \frac{1}{2} \left( \frac{1}{x_2} - \frac{1}{z_1} \right) \left\| P_a (1) \right\|.$$  

The dual function $g (\alpha, \beta, \rho) = \min_r L (r, \alpha, \beta, \rho)$ can be obtained by setting $\nabla_r L (r, \alpha, \beta, \rho) = 0$, which leads to:

$$\beta = \frac{1}{\rho} \left( \hat{f} + a^\top + \rho y \right).$$  

Since $\beta \neq 0, \left\| r \right\|_2 = l$. Therefore, $\beta$ can be written as:

$$\beta = \frac{\left\| \hat{f} + a^\top + \rho y \right\|_2}{l}.$$  

Since $\alpha \neq 0, a^\top r = 0$. Therefore, $\alpha$ can be written as:

$$\alpha = -a^\top (\hat{c} + r).$$  

Plugging the obtained $r$, $\alpha$, and $\beta$ into $L (r, \alpha, \beta, \rho)$ leads to:

$$g (\rho) = \min_r L (r, \alpha, \beta, \rho) = -l \left( \hat{f} + a^\top + \rho y \right) + \rho \hat{c}^\top y = -l \left( \hat{f} - a^\top \hat{a} + \rho y - a^\top ya \right) + \rho \hat{c}^\top y = -l \left( P_a (\hat{f}) + \rho P_a (y) \right) + \rho \hat{c}^\top y.$$

$g (\rho)$ can be maximized by setting $\frac{dg (\rho)}{d\rho} = 0$, which leads to:

$$\frac{l P_a (y) + P_a (\hat{f}) - P_a (y)}{l} = \hat{c}^\top y.$$

Squaring both sides of the equation and solving the resulting problem yields a closed-form solution for $\rho$:

$$\rho = \frac{P_a (\hat{f}) - P_a (y)}{l} \frac{l P_a (y) + P_a (\hat{f}) - P_a (y)}{l} = \frac{P_a (\hat{f}) - P_a (y)}{l} \frac{l P_a (y) + P_a (\hat{f}) - P_a (y)}{l}.$$

To obtain this equation, the following facts are used:

$$\hat{c}^\top y = \frac{1}{4} \left( \frac{1}{x_2} - \frac{1}{z_1} \right)^2 P_a (y)^\top P_a (1),$$  

$$l^2 = \frac{1}{4} \left( \frac{1}{x_2} - \frac{1}{z_1} \right)^2 P_a (1)^\top P_a (1).$$
Since \((\hat{c} + r)^\top y = 0\), \((P_a(\hat{c}) + P_a(r))^\top P_a(y) = 0\). It can be verified that \(P_a(r) = -\frac{1}{\beta} \left( P_a(\hat{f}) + \rho P_a(y) \right) \). Therefore, \(\beta = \frac{P_a(\hat{f})^\top P_a(y) + \rho P_a(y)^\top P_a(y)}{P_a(\hat{c})^\top P_a(y)}\). To ensure that \(\beta\) is positive, the following equation holds:

\[
\rho = -\frac{P_a(\hat{f})^\top P_a(y) - \|P_{\beta a}(y)(P_a(1))\|_2^2}{P_a(\hat{c})^\top P_a(y)}.
\]

In this case, \(\beta\) can be written in the form:

\[
\beta = 2 \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right)^{-1} \frac{\|P_{\beta a}(y)(P_a(\hat{f}))\|_2^2}{\|P_{\beta a}(y)(P_a(1))\|_2^2}.
\]

To compute \(\max_\rho g(\rho)\), first, Eq. (43) can be rewritten as:

\[
\frac{1}{2} \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \left( -\|P_a(y)(P_a(\hat{f}))\|_2^2 + \|P_a(y)(P_a(1))\|_2^2 - P_a(\hat{f})^\top P_a(y)(P_a(1))^\top P_a(y) + \frac{P_a(\hat{f})^\top P_a(y)(P_a(1))^\top P_a(y)}{\|P_a(y)(P_a(1))\|_2^2} \right).
\]

Since \(P_a(1)^\top P_a(\hat{f}) - P_a(y)^\top P_a(y)(P_a(1)^\top P_a(\hat{f}))\), \(\max_\rho g(\rho)\) can also be written in the form:

\[
\frac{1}{2} \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \left( -\|P_{\beta a}(y)(P_a(\hat{f}))\|_2^2 + \|P_{\beta a}(y)(P_a(1))\|_2^2 \right) + \frac{P_{\beta a}(y)(P_a(1)^\top P_{\beta a}(y)(P_a(\hat{f}))) - P_{\beta a}(y)(P_a(\hat{f}))}{\|P_a(y)(P_a(1))\|_2^2}.
\]

Theorem 3.9 summarizes the result when \(\beta > 0\) and \(\alpha > 0\).

**Theorem 3.9.** When \(\rho > 0\) and \(\alpha > 0\), it can be computed as:

\[
\min_\rho \rho^\top \hat{f} = \frac{1}{2} \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \left( -\|P_{\beta a}(y)(P_a(\hat{f}))\|_2^2 + \|P_{\beta a}(y)(P_a(1))\|_2^2 \right) + \frac{P_{\beta a}(y)(P_a(1)^\top P_{\beta a}(y)(P_a(\hat{f}))) - P_{\beta a}(y)(P_a(\hat{f}))}{\|P_a(y)(P_a(1))\|_2^2}.
\]

**Corollary 3.10.** When \(\theta^\top \hat{f}\) achieves its minimum value at \(\beta > 0\) and \(\alpha > 0\), it can be computed as:

\[
\min_\beta \theta^\top \hat{f} = \min_\rho \rho^\top \hat{f} - \hat{c}^\top \hat{f} = \frac{1}{2} \left( \frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) \left( -\|P_{\beta a}(y)(P_a(\hat{f}))\|_2^2 + \|P_{\beta a}(y)(P_a(1))\|_2^2 \right) - \frac{P_{\beta a}(y)(P_a(1))^\top P_{\beta a}(y)(P_a(\hat{f}))) - \hat{f}^\top \theta}{\|P_a(y)(P_a(1))\|_2^2}.
\]

**3.3 The Screening Algorithm**

Algorithm 1 shows the procedure of screening features for \(\ell_1\)-regularized \(\ell_2\)-SVM. Given \(\lambda_1, \lambda_2,\) and \(\theta_1\), the algorithm returns a list \(L\), which contains the indices of the features that are potentially active in the optimal solution that corresponds to \(\lambda_2\). The algorithm first weights all features using \(y^3\) in Line 3. It then computes \(\max_\ell \hat{f}\) for features in Line 4 and Line 5. If the value is larger than 1, it adds the index of the feature to \(L\) in Line 7. The function \(\text{neg-min}(\hat{f})\) computes \(-\min_\ell \hat{f}\). Since \(P_a(\nu) = -P_a(\nu)\), the intermediate results computed for \(\text{neg-min}(\hat{f})\) can be used by \(\text{neg-min}(-\hat{f})\) to accelerate its computation.

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 is surprisingly fast. First, \(y^3\) and \(\hat{f}\) are independent of \(\theta_1, \lambda_1, \) and \(\lambda_2\). Therefore, they can be precomputed before training, and the cost is \(O(mn)\). \(\theta_1^\top y, \theta_1^\top 1,\) and \(\theta_1^\top \theta_1\) are shared by all the features. So they can be computed at the beginning of screening, and the cost is \(O(n)\). Given these intermediate results, most substeps for computing \(\max_\ell \hat{f}\) can be obtained in \(O(1)\). The only expensive substep is to compute \(\hat{f}^\top \theta_1\), and its cost is \(O(mn)\) for \(m\) features. However, when a solver fits a \(\ell_1\)-regularized \(\ell_2\)-SVM model, it might have already computed \(\hat{f}^\top \theta_1\) as an intermediate result for all the features. In this case, \(\hat{f}^\top \theta_1\) can be obtained from the solver for screening features at no cost.

In summary, in the worst case of the proposed procedure, the total computational cost for screening a data set that has \(m\) features and \(n\) samples is \(O(mn)\). And if \(\hat{f}^\top \theta_1, \hat{f}^\top 1,\) \(\hat{f}^\top y,\) and \(\hat{f}^\top \hat{f}\) can be obtained from the intermediate results generated by the \(\ell_1\)-regularized \(\ell_2\)-SVM solver, the total cost can decrease to just \(O(m)\).

**4. EMPIRICAL STUDY**

The proposed screening method was implemented in the C language and compiled as a library that can be conveniently accessed in a high-level programming language, such as the Python or SAS®. This section evaluates its power for accelerating model selection for \(\ell_1\)-regularized \(\ell_2\)-SVM. Experiments are performed on a Windows Server 2008 R2 with two Intel Xeon® L5530 CPUs and 72GB memory.
Three are text data sets: the gli85 data set is downloaded from the LIBSVM data repository and is inactive in most cases. In contrast, the improved one solves the problem specified in Eq. (1) exactly. Therefore, the bias term is not penalized and is always active. Since \(-\mathbf{1} \cdot \mathbf{\theta}_1\) is the gradient on a coordinate, \(-\mathbf{1} \cdot \mathbf{\theta}_1\) is computed in the solver as an intermediate result. Therefore, in screening, \(-\mathbf{1} \cdot \mathbf{\theta}_1\) values can be obtained from the solver at no cost.

For each given benchmark data set, the cgd solver is used to fit the \(\ell_1\)-regularized \(\ell_2\)-SVM model along a sequence of 20 \(\lambda\) values: \(\{\lambda_k = \frac{1}{2} \lambda_{\text{max}} - \epsilon, k = 1, \ldots, 20, \epsilon = 10^{-8}\}\). When \(\lambda = \lambda_{\text{max}} - \epsilon\), only one feature is active. Denote \(n_+\) and \(n_-\) as the number of positive and negative samples, respectively. And let \(\mathbf{m} = \sum_{i=1}^{n_i} (y_i - \frac{n_+ - n_-}{n}) \mathbf{x}_i\). This feature corresponds to the largest element in \(\mathbf{m}\).

For each given benchmark data set, the cgd solver runs in seven different configurations: (1) In org, the solver runs without any accelerating technique. (2) In warm, the solver runs with warm-start. In the \(k\)th iteration, the \(\mathbf{w}_{k-1}\) obtained in the \((k-1)\)th iteration is used as the initial \(\mathbf{w}_k\) for fitting the model. When \(\lambda_k\) and \(\lambda_{k-1}\) are close, warm-start can effectively speed up training by reducing the number of iterations for the solver to converge. (3) In shr, the solver runs with the shrinking strategy. During each iteration of the cgd solver run, if a feature’s current weight is 0 and its gradient is very small, the feature is set to be inactive [6]. (4) In warmshr, the solver runs with both warm-start and the shrinking strategy. (5) In scr, the solver runs with the screening technique. (6) In warmscr, the solver runs with both warm-start and the screening technique. (7) In scrshr, the solver runs with both the shrinking strategy and the screening technique.

Warm-start and screening are designed to speed up model selection, and shrinking is designed to speed up training. These techniques can be combined for further performance improvement. The main purpose of running the \(\ell_1\)-regularized \(\ell_2\)-SVM solver with different configurations is not only to compare different accelerating techniques, but also to provide a sensitivity study for exploring how these techniques can be combined to achieve the best performance. Both screening and shrinking reduce computational cost by removing inactive features. Their major differences include the following: (1) Shrinking is performed in each iteration of training to reduce the search space of the solver, whereas screening is performed only once before training. (2) Shrinking is a heuristic method for removing inactive features. Sometimes it might also remove active features; when this happens, recovering the true result leads to extra cost. In contrast, the proposed screening technique is safe, because all the removed features are guaranteed to be inactive the optimal solution. (3) The introduced shrinking technique works only for the cgd solver. In contrast, the proposed screening technique can be applied with any \(\ell_1\)-regularized \(\ell_2\)-SVM solver to speed up model selection. Therefore, the proposed screening technique is more general.

4.1 Experiment Setup

Five benchmark data sets are used in the experiment. One is a microarray data set: gli85. Three are text data sets: rcv1.binary(rcv1b), real-sim, and news20.binary (news20b). And one is a educational data mining data set: kddb. The gli85 data set is downloaded from Gene Expression Omnibus,\(^2\) and the other four data sets are downloaded from the LIBSVM data repository.\(^3\) 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 gli85 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.

A solver based on the coordinate gradient descent (cgd) algorithm [13] is implemented in the C language for training the \(\ell_1\)-regularized \(\ell_2\)-SVM model. This solver improves the one that is implemented in the liblinear package [6]. In liblinear, the bias term \(b\) is also penalized by the \(\ell_1\) regularizer and is inactive in most cases. In contrast, the improved one solves the problem specified in Eq. (1) exactly. Therefore, the bias term is not penalized and is always active. Since \(-\mathbf{1} \cdot \mathbf{\theta}_1\) is the gradient on a coordinate, \(-\mathbf{1} \cdot \mathbf{\theta}_1\) is computed in the solver as an intermediate result. Therefore, in screening, \(-\mathbf{1} \cdot \mathbf{\theta}_1\) values can be obtained from the solver at no cost.

For each given benchmark data set, the cgd solver is used to fit the \(\ell_1\)-regularized \(\ell_2\)-SVM model along a sequence of 20 \(\lambda\) values: \(\{\lambda_k = \frac{1}{2} \lambda_{\text{max}} - \epsilon, k = 1, \ldots, 20, \epsilon = 10^{-8}\}\). When \(\lambda = \lambda_{\text{max}} - \epsilon\), only one feature is active. Denote \(n_+\) and \(n_-\) as the number of positive and negative samples, respectively. And let \(\mathbf{m} = \sum_{i=1}^{n_i} (y_i - \frac{n_+ - n_-}{n}) \mathbf{x}_i\). This feature corresponds to the largest element in \(\mathbf{m}\).

For each given benchmark data set, the cgd solver runs in seven different configurations: (1) In org, the solver runs without any accelerating technique. (2) In warm, the solver runs with warm-start. In the \(k\)th iteration, the \(\mathbf{w}_{k-1}\) obtained in the \((k-1)\)th iteration is used as the initial \(\mathbf{w}_k\) for fitting the model. When \(\lambda_k\) and \(\lambda_{k-1}\) are close, warm-start can effectively speed up training by reducing the number of iterations for the solver to converge. (3) In shr, the solver runs with the shrinking strategy. During each iteration of the cgd solver run, if a feature’s current weight is 0 and its gradient is very small, the feature is set to be inactive [6]. (4) In warmshr, the solver runs with both warm-start and the shrinking strategy. (5) In scr, the solver runs with the screening technique. (6) In warmscr, the solver runs with both warm-start and the screening technique. (7) In scrshr, the solver runs with both the shrinking strategy and the screening technique.

Warm-start and screening are designed to speed up model selection, and shrinking is designed to speed up training. These techniques can be combined for further performance improvement. The main purpose of running the \(\ell_1\)-regularized \(\ell_2\)-SVM solver with different configurations is not only to compare different accelerating techniques, but also to provide a sensitivity study for exploring how these techniques can be combined to achieve the best performance. Both screening and shrinking reduce computational cost by removing inactive features. Their major differences include the following: (1) Shrinking is performed in each iteration of training to reduce the search space of the solver, whereas screening is performed only once before training. (2) Shrinking is a heuristic method for removing inactive features. Sometimes it might also remove active features; when this happens, recovering the true result leads to extra cost. In contrast, the proposed screening technique is safe, because all the removed features are guaranteed to be inactive the optimal solution. (3) The introduced shrinking technique works only for the cgd solver. In contrast, the proposed screening technique can be applied with any \(\ell_1\)-regularized \(\ell_2\)-SVM solver to speed up model selection. Therefore, the proposed screening technique is more general.

### Table 1: Summary of the benchmark data sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>sample (n)</th>
<th>feature (m)</th>
<th>m/n</th>
</tr>
</thead>
<tbody>
<tr>
<td>gli85</td>
<td>85</td>
<td>22283</td>
<td>262.15</td>
</tr>
<tr>
<td>rcv1b</td>
<td>20242</td>
<td>47236</td>
<td>2.33</td>
</tr>
<tr>
<td>real-sim</td>
<td>72309</td>
<td>20958</td>
<td>0.29</td>
</tr>
<tr>
<td>news20b</td>
<td>19996</td>
<td>1355191</td>
<td>67.77</td>
</tr>
<tr>
<td>kddb</td>
<td>19264097</td>
<td>29890095</td>
<td>1.55</td>
</tr>
</tbody>
</table>


\(^3\)www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/
4.2 Results

Table 2 and Table 3 show the results of the total run time and the total number of iterations for the $\ell_1$-regularized $\ell_2$-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 $\ell_1$-regularized $\ell_2$-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 445 for the gli_85 data and 53 for the news20b data. For the $m \approx n$ group, the speed-up ratio is about 10 for the rcv1b data and 11 for the kddb data. And for the $m \ll n$ group, the speed-up ratio is about 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 and scr_shr. 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. A similar trend is observed when shr is compared to warm_shr.

When org is compared to scr, the result suggests that the proposed screening technique can significantly improve the performance of the $\ell_1$-regularized $\ell_2$-SVM solver. This justifies that screening can effectively reduce the computational cost of training by removing most inactive features. When shr is compared to scr, the result suggests that screening performs faster. This is because shrinking is a heuristic method for removing inactive features. Sometimes, it might remove active features during training. When this happens, violations can be detected by using the KKT conditions for the optimal solution. However, recovering the optimal solution leads to extra cost. This is supported by the observation that with shr the solver usually takes more iterations to converge than with org and scr. When warm is compared to scr and shr, the results suggest that removing inactive features for training is more effective than providing a good starting point for optimization.

The results presented in Table 2 and Table 3 suggest that the performance of screening and shrinking can be further improved by combining them with warm-start. This is because warm-start can effectively speed up convergence by providing a good starting point for optimization. However, combining screening with shrinking does not improve the performance of screening because that screening has already removed many inactive features before training is performed. When shrinking is used in training, its benefit for removing inactive features becomes insubstantial and is overwhelmed by the cost of recovering the optimal solution when it accidentally removes active features.

Figure 2 shows detailed information about how different combinations of accelerating techniques perform on the news20b data set when $\lambda$ decreases from $\lambda_{\text{max}}$ to $\frac{1}{b} \lambda_{\text{max}}$. The result shows that screening with warm-start is effective for accelerating and its performance is stable. It also shows that when $\lambda$ decreases, the proposed screening technique can stably select a small set of features for reducing computational cost. Let $k$ be the number of active features. On the news20b data set, when $\lambda$ decreases from $\lambda_{\text{max}}$ to $\frac{1}{b} \lambda_{\text{max}}$, the proposed screening technique retains about $k+430$ features for training the $\ell_1$-regularized $\ell_2$-SVM model. This number is much smaller than the dimensionality of the news20b data set, which is about 1.3 million. Similar trends are also observed on other data sets and are not presented in the paper because of the space limit.

Table 4 compares the time used by screening to the time used by training. Notice that for training, the solver uses only the features that are selected by screening. Compared to training, the time used by screening is marginal.

The results presented in this section indicate that the proposed screening technique is effective for removing inactive features to improve training efficiency. And with warm-start they form the most powerful combination for accelerating model selection for the $\ell_1$-regularized $\ell_2$-SVM.

### Table 2: Total run time of the $\ell_1$-regularized $\ell_2$-SVM solver when different combinations of accelerating techniques are used to speed up model selection.

<table>
<thead>
<tr>
<th>Tech.</th>
<th>gli_85</th>
<th>rcv1b</th>
<th>real-sim</th>
<th>news20b</th>
<th>kddb</th>
</tr>
</thead>
<tbody>
<tr>
<td>org</td>
<td>328.7</td>
<td>17.92</td>
<td>20.81</td>
<td>943.67</td>
<td>9209.06</td>
</tr>
<tr>
<td>warm</td>
<td>376.6</td>
<td>10.30</td>
<td>13.48</td>
<td>682.08</td>
<td>7752.80</td>
</tr>
<tr>
<td>shr</td>
<td>2.78</td>
<td>4.49</td>
<td>7.25</td>
<td>62.44</td>
<td>3374.53</td>
</tr>
<tr>
<td>warm_shr</td>
<td>3.10</td>
<td>2.31</td>
<td>4.32</td>
<td>32.62</td>
<td>2395.45</td>
</tr>
<tr>
<td>scr</td>
<td>0.78</td>
<td>3.35</td>
<td>6.67</td>
<td>25.53</td>
<td>1126.05</td>
</tr>
<tr>
<td>warm_scr</td>
<td><strong>0.74</strong></td>
<td><strong>1.78</strong></td>
<td><strong>4.30</strong></td>
<td><strong>17.84</strong></td>
<td><strong>831.87</strong></td>
</tr>
<tr>
<td>scr_shr</td>
<td>1.45</td>
<td>4.00</td>
<td>7.14</td>
<td>48.29</td>
<td>2603.94</td>
</tr>
</tbody>
</table>

### Table 3: Total number of iterations for the $\ell_1$-regularized $\ell_2$-SVM solver to converge when different combinations of accelerating techniques are used.

<table>
<thead>
<tr>
<th>Tech.</th>
<th>gli_85</th>
<th>rcv1b</th>
<th>real-sim</th>
<th>news20b</th>
<th>kddb</th>
</tr>
</thead>
<tbody>
<tr>
<td>org</td>
<td>15535</td>
<td>1062</td>
<td>568</td>
<td>2579</td>
<td>755</td>
</tr>
<tr>
<td>warm</td>
<td>14610</td>
<td>615</td>
<td><strong>373</strong></td>
<td><strong>1898</strong></td>
<td>628</td>
</tr>
<tr>
<td>shr</td>
<td>16046</td>
<td>1737</td>
<td>815</td>
<td>4995</td>
<td>2008</td>
</tr>
<tr>
<td>warm_shr</td>
<td>14888</td>
<td>713</td>
<td>431</td>
<td>2157</td>
<td>1046</td>
</tr>
<tr>
<td>scr</td>
<td>15376</td>
<td>1059</td>
<td>596</td>
<td>2862</td>
<td>843</td>
</tr>
<tr>
<td>warm_scr</td>
<td><strong>14599</strong></td>
<td><strong>590</strong></td>
<td>390</td>
<td>1999</td>
<td><strong>569</strong></td>
</tr>
<tr>
<td>scr_shr</td>
<td>16150</td>
<td>1695</td>
<td>942</td>
<td>4908</td>
<td>1901</td>
</tr>
</tbody>
</table>

### Table 4: Comparison of screening to training time

<table>
<thead>
<tr>
<th>Tech.</th>
<th>gli_85</th>
<th>rcv1b</th>
<th>real-sim</th>
<th>news20b</th>
<th>kddb</th>
</tr>
</thead>
<tbody>
<tr>
<td>scr</td>
<td>0.03</td>
<td>0.06</td>
<td>0.04</td>
<td>1.91</td>
<td>41.65</td>
</tr>
<tr>
<td>tr</td>
<td>0.75</td>
<td>3.29</td>
<td>6.63</td>
<td>23.63</td>
<td>1084.40</td>
</tr>
<tr>
<td>ratio</td>
<td>0.04</td>
<td>0.02</td>
<td>0.01</td>
<td>0.08</td>
<td>0.04</td>
</tr>
<tr>
<td>warm_scr</td>
<td>0.03</td>
<td>0.07</td>
<td>0.04</td>
<td>1.91</td>
<td>41.71</td>
</tr>
<tr>
<td>tr</td>
<td>0.70</td>
<td>1.72</td>
<td>4.26</td>
<td>15.93</td>
<td>790.15</td>
</tr>
<tr>
<td>ratio</td>
<td>0.05</td>
<td>0.04</td>
<td>0.01</td>
<td>0.12</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 4 compares the time used by screening to the time used by training. Notice that for training, the solver uses only the features that are selected by screening. Compared to training, the time used by screening is marginal.
Figure 2: Detailed information about how different combinations of accelerating techniques perform when \( \lambda \) decreases from \( \lambda_{\text{max}} \) to \( \frac{1}{2} \lambda_{\text{max}} \). Results are reported for the news20b data set. “Run time” is the time that is used for training. For scr, warm_scr, and scr_shr, run time includes screening time. “Iterations” is the number of iterations for the solver to converge. “Over hits” is the number of inactive features that are not removed by screening. The results show that the proposed screening technique improves efficiency significantly. And when \( \lambda \) decreases, the number of leftover inactive features is small and stable.

5. CONCLUSION
Screening is an effective technique for improving model selection efficiency by eliminating inactive features. This paper proposes a novel technique to screen features for \( \ell_1 \)-regularized \( \ell_2 \)-SVM. The key contribution of this paper is the usage of the variational inequality for deriving closed-form criteria to screen features for the \( \ell_1 \)-regularized \( \ell_2 \)-SVM model in different situations. Empirical study shows that the proposed technique can greatly improve model selection efficiency by stably eliminating a large number of inactive. Our ongoing work will extend the technique to screen features for the \( \ell_1 \)-regularized \( \ell_1 \)-SVM model.

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

7. REFERENCES