Safe Sample Screening for Robust Support Vector Machine

Zhou Zhai,[1] Bin Gu,[1,2*] Xiang Li,[3] Heng Huang[4]
[1]School of Computer & Software, Nanjing University of Information Science & Technology, P.R.China
[2]JD Finance America Corporation
[3]Computer Science Department, University of Western Ontario, Canada
[4]Computer Engineering, University of Pittsburgh, USA
zhouzhai@nuist.edu.cn, {jsgubin, henghuanghh}@gmail.com, lxiang2@uwo.ca

Abstract

Robust support vector machine (RSVM) has been shown to perform remarkably well to improve the generalization performance of support vector machine under the noisy environment. Unfortunately, in order to handle the non-convexity induced by ramp loss in RSVM, existing RSVM solvers often adopt the DC programming framework which is computationally inefficient for running multiple outer loops. This hinders the application of RSVM to large-scale problems. Safe sample screening that allows for the exclusion of training samples prior to or early in the training process is an effective method to greatly reduce computational time. However, existing safe sample screening algorithms are limited to convex optimization problems while RSVM is a non-convex problem. To address this challenge, in this paper, we propose two safe sample screening rules for RSVM based on the framework of concave-convex procedure (CCCP). Specifically, we provide screening rule for the inner solver of CCCP and another rule for propagating screened samples between two successive solvers of CCCP. To the best of our knowledge, this is the first work of safe sample screening to a non-convex optimization problem. More importantly, we provide the security guarantee to our sample screening rules to RSVM. Experimental results on a variety of benchmark datasets verify that our safe sample screening rules can significantly reduce the computational time.

Introduction

In supervised learning, support vector machine (SVM) (Chang and Lin 2011; Platt 1998; Cortes and Vapnik 1995) is a powerful classification method that is widely used to separate data by maximizing the margin between two classes. However, real-world data tend to be massive in quantity but with quite a few unreliable outliers. Traditional SVM usually use convex hinge loss function to calculate the loss of misclassified samples. Since the convex function is unbounded and puts an extremely large penalty on outliers, traditional SVM is unstable in the presence of outliers. Robust support vector machine (RSVM) (Wu and Liu 2007; Shen et al. 2003; Xu, Crammer, and Schuurmans 2006) suppresses the influence of outliers on the decision function through clipping the convex hinge loss to the non-convex ramp loss, and has been shown to perform remarkably well under the noisy environment.

The non-convex objective function of RSVM can be viewed as a difference of convex (DC) (Sriperumbudur and Lanckriet 2009) programming problem which is normally solved by the concave-convex procedure (CCCP) (Yuille and Rangarajan 2003; Collobert et al. 2006) algorithm. The CCCP algorithm iteratively solves a sequence of constrained convex optimization problems. For each loop of CCCP algorithm, it solves a surrogate convex optimization problem which linearizes the concave part of the original DC programming problem. The inner surrogate convex optimization problem is very similar to the problem of SVM, and is normally solved by the sequential minimal optimization (SMO) algorithm (Platt 1998; Vapnik 2013). As pointed out in (Chang and Lin 2011), the time complexity of SMO algorithm is $O(n^\kappa)$, where $1 < \kappa < 2.3$, n is the number of the training samples. Thus, the time complexity of CCCP algorithm to solve RSVM is $O(tn^\kappa)$, where t is the number of loops of the CCCP algorithm. The high computational cost severely hinders the implementation of RSVM and its application to big data.

To address the above challenging problem, one promising approach is safe screening. Ghaoui et al. (2010) first exploited safe screening rules to discard inactive features prior to starting a Lasso solver. They exploited the geometric quantities of the feature space to bound the Lasso dual solution to be within a compact region and only need to solve a smaller optimization problem on the reduced datasets which leads to huge savings in the computational cost and memory usage. Since then, the concept of safe screening has been expanded in two main directions. The first direction is called *sequential screening*, which performs screening along the entire regularization path which is the sequence of optimal solutions w.r.t. different values of regularization parameter. Sequential screening relies on an additional feasible or optimal solution obtained in advance, which can provide a warm start of the screening process. This direction has been pursued in (Wang et al. 2013; 2014; Liu et al. 2013; Xu and Ramadge 2013; Xiang, Wang, and Ramadge 2016; El Ghaoui, Viallon, and Rabbani 2011). However, they are

*Contact Author

Table 1: Representative safe screening algorithm. ("Type" represents the algorithm screening samples or features).

Problem	Reference	Type	Type of screening	Warm-start	Type of optimization problems
SVM	Zimmert et al. (2015)	Samples	Dynamic	No	Convex
SVM	Ogawa et al. (2014)	Samples	Sequential	Yes	Convex
SVM	Ogawa et al. (2013)	Samples	Sequential	Yes	Convex
Logistic Regression	Wang et al. (2014)	Features	Sequential	Yes	Convex
Lasso	Liu et al. (2013)	Features	Dynamic	No	Convex
Proximal Weighted Lasso	Rakotomamonjy et al. (2019)	Features	Dynamic	Yes	Non-convex
RSVM	Our	Samples	Dynamic	Yes	Non-convex

only applicable to algorithms that also compute the regularization paths. The second direction is called *dynamic screening* (Bonnefoy et al. 2014; 2015), which performs the screening throughout the optimization algorithm itself. For example, Fercoq et al. (2015) proposed a duality gap based safe feature screening algorithm for lasso. Although dynamic screening might be useless early in the training process, it might become efficient as the algorithm proceeds towards the optimal solution. Further, Rakotomamonjy et al. (2019) expanded safe feature screening rule to lasso with non-convex sparse regularizers. They handled the non-convexity of the objective through the majorization-minimization (MM) principle and provided a warm-start process that allows to propagate screened features from one MM iteration to the next.

Recently, Ogawa et al. (2013) first proposed a safe screening to indentify non-support vectors for SVM. They extended the existing feature-screening methods to sample-screening. On this basis, Ogawa et al. (2014) and Wang et al. (2014) improved its ability to screen inactive samples. However, as sequential screening algorithms, they relies on an additional feasible or optimal solution obtained in advance, which can be very time consuming. To overcome this difficult, Zimmert et al. (2015) proposed a dynamic screening rule using a duality gap function in the primal variables of hinge loss kernel SVM. We summarized several representative safe screening algorithms in table 1. It shows that existing safe feature screening algorithms have been widely used in convex and non-convex problems while existing safe samples algorithms are limited to convex problems. Dynamic screening algorithm for SVM can not provide a warm-start for training the model, so it only works during the training the model. It is obvious that the dynamic samples screening rule for RSVM is still an open problem.

In this paper, we propose two safe sample screening rules for RSVM based on the framework of concave-convex procedure (CCCP). Specifically, we first provide a screening rule for the inner solver of CCCP. Secondly, we provide a new rule for propagating screened samples between two successive solvers of CCCP. To the best of our knowledge, this is the first work of safe sample screening to a non-convex optimization problem. More importantly, we provide the security guarantee to our sample screening rules to RSVM. Experimental results on a variety of benchmark datasets verify that our safe sample screening rules can significantly reduce the computational time.

Contributions. The main contributions of this paper are summarized as follows:

1. To the best of our knowledge, we are the first to propose a safe samples screening rule for the non-convex problem.
2. By utilizing an iterative CCCP strategy to solve RSVM, we proposed a safe samples screening rule for propagating screened samples between two successive solvers of CCCP.

Preliminaries of Robust Support Vector Machine

In this section, we first give a brief review of RSVM. Then, we give the primal and dual form of RSVM. At last, we give the screening set in the RSVM.

Robust Support Vector Machine

We consider a training set $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ constituted with n samples , where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. SVM has a discrimination hyperplane in the following form:

$$f_\theta(x_i) = w^T \phi(x_i) + b, \tag{1}$$

where $\theta = (w, b)$ are the parameters of the model, and $\phi(\cdot)$ is a transformation function from an input space to a high-dimensional reproducing kernel Hilbert space. SVM solves the following minimization problem:

$$\min_\theta \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} H_1(y_i f_\theta(x_i)) \tag{2}$$

where the function $H_s(z) = \max(0, s - z)$ is the hinge loss. Since the convex hinge loss function is unbounded and puts an extremely large penalty on outliers, traditional SVM is unstable in the presence of outliers. We clip the hinge loss to get the ramp loss $R_s(z) = \min(s - z, H_1(z)) = H_1(z) - H_s(z)$, where $s \leq 0$. The ramp loss is bounded, meaning that noisy samples cannot influence the solution beyond that of any other misclassified point. Thus, RSVM can effectively suppress the influence of outliers and it solves the following minimization problem:

$$\min_\theta \underbrace{\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} H_1(y_i f_\theta(x_i))}_{o(\theta)} \underbrace{- C\sum_{i=l}^{n} H_s(y_i f_\theta(x_i))}_{v(\theta)} \tag{3}$$

where o and v are real-valued convex functions. It is easy to see that the objective function (3) is a form of DC.

Primal and Dual Problem

The non-convex objective function of RSVM can be viewed as a DC programming problem which is normally solved by the CCCP algorithm. The main mechanism of CCCP algorithm is to iteratively construct an optimized surrogate objective function which linearizes the concave part of the original DC programming problem. In order to apply the CCCP algorithm to solve the problem (3), we first have to calculate derivative of the concave part with respect to θ:

$$\theta \cdot \nabla v(\theta) = -\sum_{i=1}^{n} \mu_i y_i f_\theta(x_i) \tag{4}$$

$$\text{where} \quad \mu_i = \begin{cases} C & \text{if } y_i f_\theta(x_i) < s \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The primal problem can be transformed into the following optimized surrogate objective:

$$P(\theta) = \min_{\theta} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} H_1(y_i f_\theta(x_i)) + \sum_{i=1}^{n} \mu_i y_i f_\theta(x_i) \tag{6}$$

In this paper, we call the formulation (6) as convex inner loop (CIL) problem. Using Lagrange multiplier method (Bertsekas 2014), we directly give the dual form of the primal problem as follows:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T H \alpha - y^T \alpha \\ s.t. \quad & \sum_{i=1}^{n} \alpha_i = 0; \quad \underline{C}_i < \alpha_i < \overline{C}_i \end{aligned} \tag{7}$$

where H is a positive semidefinite matrix with $H_{ij} = K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ for all $1 \le i, j \le n$, $K(x_i, x_j)$ is the kernel function, $\underline{C}_i = \min(0, Cy_i) - \mu_i y$, $\overline{C}_i = \max(0, Cy_i) - \mu_i y$, $\alpha_i = y_i(\beta_i - \mu_i)$ and β_i is the Lagrange multiplier.

According to the convex optimization theory (Boyd and Vandenberghe 2004), the dual CIL problem (7) can be transformed into the following min-max form:

$$D(\theta') = \min_{\alpha} \max_{b' \in \mathbb{R}} \frac{1}{2}\alpha^T H \alpha - y^T \alpha + b' \left(\sum_{i=1}^{n} \alpha_i \right) \tag{8}$$

where $\theta' = (\alpha, b')$ are the parameters of the dual CIL problem and b' is the Lagrangian multiplier. Further, from the KKT theorem (Bazaraa and Shetty 2012), the first-order derivative of $D(\theta')$ with respect to α leads to the following KKT conditions:

$$\nabla D(\theta')_i \overset{\text{def}}{=} \frac{\partial D(\theta')}{\partial \alpha_i} = \sum_{j=1}^{n} \alpha_j H_{ij} + b' - y_i \tag{9}$$

Screening Set

Safe sample screening is built on the KKT optimality condition (Bertsekas 1997). According to the gradient $\nabla D(\theta')_i$, we can categorize the n training samples into three cases:

$$\nabla D(\theta')_i \begin{cases} > 0 & \alpha_i = \underline{C}_i \\ = 0 & \alpha_i \in [\underline{C}_i, \overline{C}_i] \\ < 0 & \alpha_i = \overline{C}_i \end{cases} \tag{10}$$

Switching to the primal problem, (10) leads to the following four cases:

$$y_i f_\theta(x_i) > 1 \quad \Rightarrow \quad \alpha_i = 0; \tag{11}$$

$$y_i f_\theta(x_i) = 1 \quad \Rightarrow \quad \alpha_i \in [\underline{C}_i, \overline{C}_i]; \tag{12}$$

$$s \le y_i f_\theta(x_i) < 1 \quad \Rightarrow \quad \alpha_i = y_i C; \tag{13}$$

$$y_i f_\theta(x_i) < s \quad \Rightarrow \quad \alpha_i = 0 \tag{14}$$

If some of the training samples are known to satisfy the case (11) or (14) in advance, we can throw away those samples prior to the training stage. Similarly, if we know that some samples satisfy case (13), we can fix the corresponding α_i at the following training process. Namely, if some knowledge on these five cases are known a-priori, our training task would be extremely easy. The samples satisfy the case (11) or (14) are often called non-support vectors because they have no influence on the resulting classifier.

In this paper, we show that, through our safe sample screening rule, some of the non-SVs and some of the samples satisfying case (13) or (14) can be screened out prior to the training process. Then, in the latter training process, we can train the model with fewer samples to reduce computation time while ensuring consistent results. Suppose that we obtain an active set A (a subset of D) after applying our safe sample screening rule, correspondingly, we can define an inactive set $\bar{A} = D - A$ that the variables $\alpha_{\bar{A}}$ are fixed. The original optimization (7) can be reduced into a smaller optimization problem as follows.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha_A^T H_{AA} \alpha_A - (y_A - H_{A\bar{A}}\alpha_{\bar{A}})^T \alpha_A \\ s.t. \quad & \sum_{i=1}^{n} \alpha = 0; \quad \underline{C}_i < \alpha_i < \overline{C}_i \end{aligned} \tag{15}$$

which is a smaller optimization problem. Notice that different from existing approximate shrinking heuristics (Chang and Lin 2011; Joachims 1999b; Gu et al. 2018; Joachims 1999a; Fan, Chen, and Lin 2005) which sample through a boundary without well theory guarantees, the active set obtained by our sample screening rule is safe and reliable.

Safe Screening Rule for Single CIL Problem

In this section, we first provide the safe screening rule for single CIL problem. Then, we give the implementation of single CIL problem. Finally, we analyze the security analysis of sample screening rule.

Safe Screening Rule

As stated in the duality theory, the dual problem $D(\theta')$ is a lower bound on the primal problem $P(\theta)$. When strong duality theorem (Boyd and Vandenberghe 2004) is satisfied, the optimal solution of the dual problem is equal to the optimal solution of the primal problem. We define the duality gap functions $G_P(\theta)$ as follows:

$$\begin{aligned} G_P(\theta) &= \min_{\theta = \theta(\theta')} G_D(\theta(\theta')) \\ &= P(\theta) - \max_{\theta = \theta(\theta')} D(\theta') \end{aligned} \tag{16}$$

$$= \|w\|^2 + C\sum_{i=1}^{n} H_1(y_i f_\theta(x_i)) + \sum_{i=1}^{n} \mu_i y_i f_\theta(x_i) - \max_{\alpha}\left(\sum_{i=1}^{n}(y_i - b')\alpha_i\right)$$

and $G_D(\theta') = P(\theta(\theta')) - D(\theta')$ respectively. The weak duality theorem (Boyd and Vandenberghe 2004) guarantees that duality gap is always greater than 0. In the following, we first show that the duality gap is a strong convex function.

Property 1. *The duality gap $G_P(\theta)$ is strongly convex with parameter θ. Then*

$$G_P(\theta_1) \geq G_P(\theta_2) + \langle \nabla G_P(\theta_2), \theta_1 - \theta_2 \rangle + \|\theta_1 - \theta_2\|_{\mathcal{H}}^2$$

We provide the detailed proof in the appendix. According to the strongly convex property of the duality gap, we can easily get that the euclidean distance between arbitrarily feasible solution and the optimal solution is always less than the current duality gap.

Corollary 1. *Let $\theta^* = (w^*, b^*)$ be the optimal solution of the primal problem. Then we have*

$$\|\theta - \theta^*\| \leq \sqrt{G_D(\theta')} \tag{17}$$

We provide the detailed proof in the appendix. According to (9), we can obtain the relation between the feasible solution and the optimal solution:

$$\begin{aligned} \nabla D(\theta'^*)_i &= \sum_{j=1}^{n} \alpha_j^* H_{ij} + b'^* - y_i \qquad (18)\\ &= \sum_{j=1}^{n}(\alpha_j^* - \alpha_j)H_{ij} + \sum_{j=1}^{n}\alpha_j H_{ij} - y_i \\ &\quad + b'^* - b' + b' \\ &= \nabla D(\theta')_i + \sum_{j=1}^{n}(\alpha_j^* - \alpha_j)H_{ij} + b'^* - b' \\ &= \nabla D(\theta')_i + \langle \theta^* - \theta', \phi(x_i)\rangle \end{aligned}$$

Based on Corollary 1, we further obtain the inequality relation between the euclidean distance from the feasible solution to the optimal solution of any sample and the current duality gap.

Corollary 2. *Let $\theta'^* = (\alpha^*, b'^*)$ be the optimal solution of the dual problem. Denote K_{ii} the entries of the associated kernel matrix, then for all $i = 1, \cdots, n$ we have:*

$$|\nabla D(\theta'^*)_i - \nabla D(\theta')_i| \leq \sqrt{K_{ii} \cdot G_D(\theta')} \tag{19}$$

We provide the detailed proof in the appendix. According to Corollary 2, we know that the optimal solution $\nabla D(\theta'^*)_i$ is always in a circle with the feasible solution $\nabla D(\theta')_i$ as the center of the circle and $r = \sqrt{K_{ii} G_D(\theta')}$ as the radius. Thus, when this circle does not contain the point $\nabla D(\theta'^*)_i = 0$, we can screen out this sample. The safe sample screening rule is summarized as follows:

$$\nabla D(\theta')_i > \sqrt{K_{ii} G_D(\theta')} \Rightarrow \alpha_i^* = \underline{C}_i \tag{20}$$

$$\nabla D(\theta')_i < -\sqrt{K_{ii} G_D(\theta')} \Rightarrow \alpha_i^* = \overline{C}_i \tag{21}$$

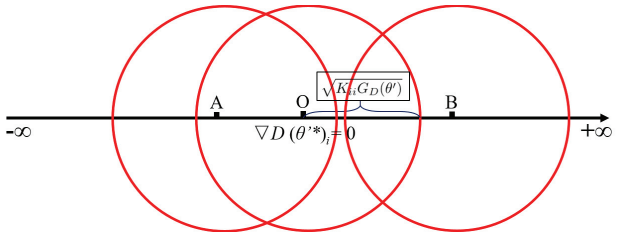


Figure 1: Illustration of safe sample screening rule. The points O in the figure represents support vector *i.e.* $\nabla D(\theta'^*)_i = 0$. During the training process, if a certain sample i is a support vector, the feasible solution $\nabla D(\theta')_i$ must be in a circle centered at O and radius $r = \sqrt{K_{ii} G_D(\theta')}$ in any iteration. Correspondingly, if the feasible solution $\nabla D(\theta')_i$ is at point B, its optimal solution must be in a circle of the same radius r. At this time, the optimal solution $\nabla D(\theta'^*)_i$ must be greater than 0. On the contrary, when the feasible solution $\nabla D(\theta')_i$ is at point A, we are not sure that the optimal solution $\nabla D(\theta')_i$ is equal to 0.

We give the illustration of safe sample screening rule in Figure 1.

Interpretation

We use a SMO algorithm to solve the CIL problem in its dual form (7). The core idea of SMO algorithm is to heuristically select two samples that violate the KKT condition to the largest extent to update. Then, we will update the gradient of all the samples and the parameter b. SMO algorithm repeat the process until it converges. The gradient of all the samples g_i is defined as follows:

$$g_i = \sum_{j=1}^{n} \alpha_j H_{ij} - y_i \tag{22}$$

During the training process, in order to use safe sample screening rule, we need to compute the duality gap. Major time-consuming of duality gap is compute $\|w\|^2$. In the following, we will show that how to use the gradient in the update process to compute the duality gap easily. According to the KKT conditions, the $\|w\|^2$ is defined as follows:

$$\|w\|^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j H_{ij} = \sum_{i=1}^{n}(g_i + y_i)\alpha_i \tag{23}$$

and the $y_i f(x_i)$ can be easily solved as:

$$y_i f(x_i) = y_i(g_i + b) - 1 \tag{24}$$

Thus, we can avoid recalculating kernel by maintaining the gradient in each iteration of SMO algorithm. The duality gap in the early stage can always be large, which makes the dual and primal estimations inaccurate and finally results in ineffective screening rules. We typically start sample screening after 50 iterations and screen the samples every 10 iterations. We summarized the safe sample screening rule for single CIL problem in Algorithm 1.

Algorithm 1 Safe sample screening for single CIL problem

Input: Training set D, optimization precision ϵ .
Output: The optimal solution of α.
1: Initialize $\alpha = 0$.
2: **while** optimaliy conditions are not satisfied with ϵ **do**
3: Select two samples points update.
4: Compute the $\nabla D(\theta')_i$ and duality gap.
5: Screening the samples that satisfy (20)-(21).
6: **end while**

Theoretical Analysis

The main advantage of our safe sample screening algorithm is its theoretic guarantee. In the following, we prove that all inactive samples would be detected and screened by our screening rule after a finite number of iterations of the algorithm.

Property 2. *Define the screening set of the CIL problem as* $\mathcal{R}^* = \{i \in D \quad |\nabla D(\theta'^*)_i| = 0\}$ *and* $\mathcal{R}_k = \{i \in D \quad |\nabla D(\theta'^k)_i| \leq \sqrt{K_{ii} G_D(\theta'^k)}\}$ *obtained at iteration* k *of an algorithm solving the CIL problem. Then, there exists* $k_0 \in \mathcal{N}$ *s.t.* $\forall k \geq k_0, \mathcal{R}_k = \mathcal{R}^*$.

We provide the detailed proof in the appendix.

Safe Screening Rule for Successive CIL Problems

In this section, we give the propagation behavior of the screened samples.

Propagating Screening Rule

Prior to this we have introduced the inner solver for single CIL problem and its safe screening rule, we are going to analyze how this rule can be improved into solving successive CIL problems. Each iteration of the CCCP algorithm approximates the concave part by its tangent and minimizes the resulting convex function, and the its tangent is always an upper bound of the concave part:

$$-C^* H_0(y_i(w^*\phi(x_i) + b^*)) \leq \mu_i^* y_i(w^*\phi(x_i) + b^*) \quad (25)$$

For each inner problem of CCCP, we can perform screening by using μ_i as defined in (5), which improves the efficiency of CCCP. However, since the μ_i's are also expected to vary for different iterations, we do not know whether the a screen sample of iteration k can be safely screened in iteration $k+1$. Thus, in the next iteration, we usually need to solve a new CIL problem due to the change of μ_i. In the following, we derive conditions that could be used to propagate screened samples from one iteration to the next in a CCCP algorithm.

First of all, we give the relation of feasible solutions of any two subproblems:

$$\nabla D(\theta'^{k+1}) = \sum_{j=1}^{n} (\alpha_j^{k+1} - \alpha_j^k) H_{ij} + (b'^{k+1} - b'^k) + \nabla D(\theta'^k)$$

Then, we consider the relation of duality gap of any two subproblems:

$$\begin{aligned} \sqrt{G_D(\theta'^{k+1})} &\leq \sqrt{|G_D(\theta'^{k+1}) - G_D(\theta'^k)| + G_D(\theta'^k)} \\ &\leq \sqrt{|G_D(\theta'^{k+1}) - G_D(\theta'^k)|} + \sqrt{G_D(\theta'^k)} \end{aligned}$$

By utilizing the relation from one iteration to the next in a CCCP algorithm, we can obtain the Property 3.

Property 3. *Consider a CIL problem with* μ_k *and its feasible solutions* θ'^k *allowing to screen samples according to (20)-(21). Suppose that we have a new set of weight of* μ^{k+1} *defining a new CIL problem. Given a primal-dual feasible solution* θ'^{k+1} *for the latter problem, a safe sample screening rule for sample* i *reads*

$$\nabla D(\theta'^k)_i - \sqrt{K_{ii} G_D(\theta'^k)} + m\|I_i\| + n - q > 0 \quad (26)$$

$$\nabla D(\theta'^k)_i + \sqrt{K_{ii} G_D(\theta'^k)} + m\|I_i\| + n + q < 0 \quad (27)$$

where m, n *and* q *are constants such that* $\|\alpha_i^{k+1} - \alpha_i^k\|_2 < m$, $|b'^{k+1} - b'^k| < n$ *and* $\sqrt{K_{ii}|G_D(\theta'^{k+1}) - G_D(\theta'^k)|} < q$, I_i *denote vector* $[H_{i1}, H_{i2}, \cdots, H_{i(n)}]$ *for all* $1 \leq i \leq n$.

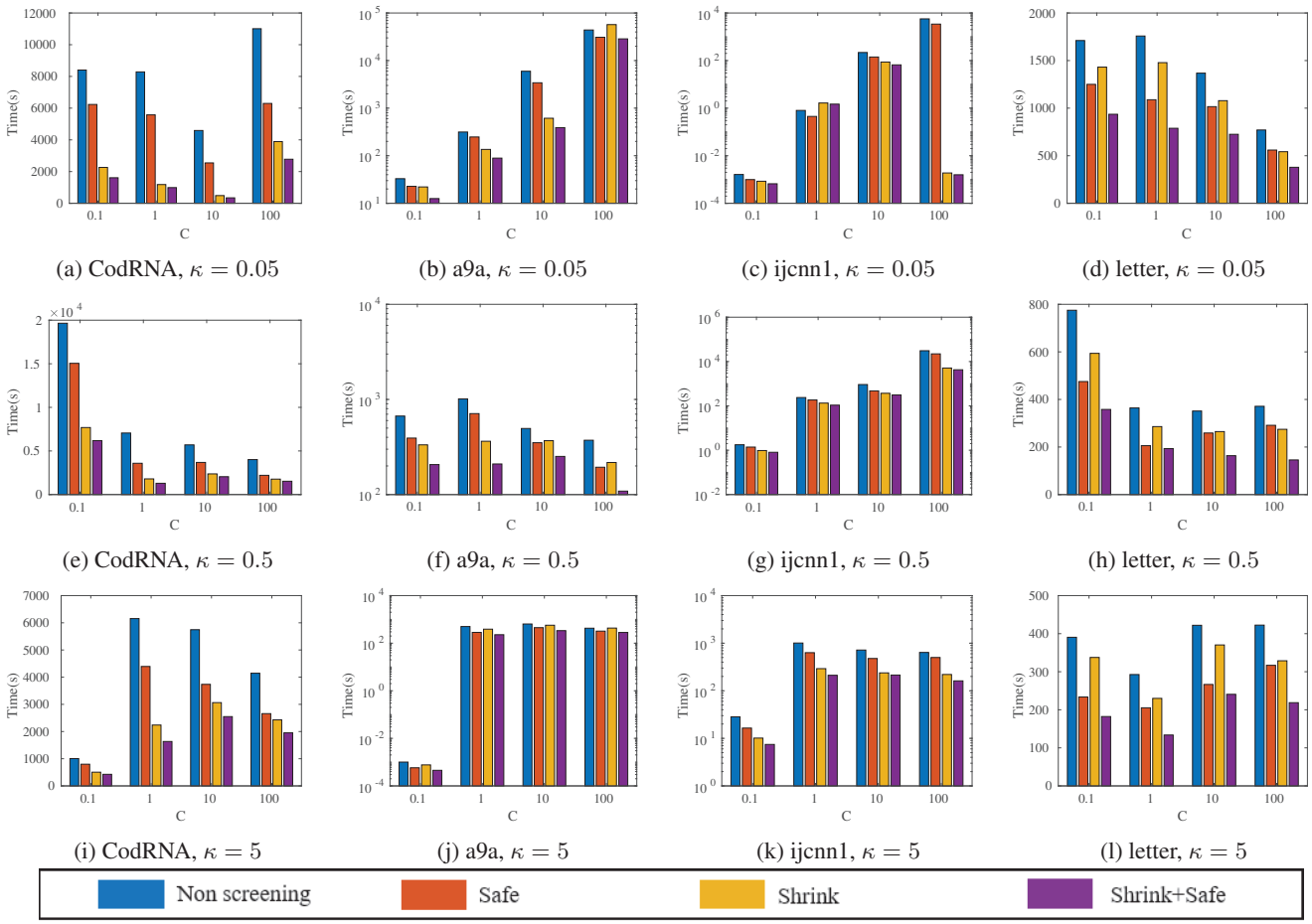
Algorithm 2 Safe sample screening for successive CIL problem

Input: The training set D.
Output: The optimal solution of RSVM.
1: Initialize the μ^0.
2: Solve a CIL problem with μ^0.
3: $k \leftarrow 1$.
4: Compute the μ_k according to (5).
5: **while** μ^k are not convergence **do**
6: Screening the samples that satisfy (26)-(27).
7: Solve a CIL problem with μ^k.
8: $k \leftarrow k + 1$.
9: Compute the μ_k according to (5).
10: **end while**

We provide the detailed proof in the appendix. To make this safe sample screening rule tractable, we first need a feasible solution θ'^{k+1} of the dual CIL problem in iteration $k+1$, then we can obtain an upper bound on the norm of $\|\alpha'^{k+1} - \alpha'^k\|$ and a bound on the difference of the duality gap $|G_D(\theta'^{k+1}) - G_D(\theta'^k)|$ and threshold $|b'^{k+1} - b'^k|$ respectively. Interestingly, when using the primal solution $\alpha^k = (\beta^k - \mu^{k+1})y$ as our feasible solution in iteration $k+1$, the safe sample screening rule given in Property 3 does not involve any additional dot product and is cheaper to compute. The propagation of screened samples provide a warm-start process for the next iteration in a CCCP algorithm. We summarized the safe sample screening rule for successive CIL problems in Algorithm 2.

Experiments

In this section, we first present the experimental setup, and then provide the experimental results and discussions.



(a) CodRNA, $\kappa = 0.05$ (b) a9a, $\kappa = 0.05$ (c) ijcnn1, $\kappa = 0.05$ (d) letter, $\kappa = 0.05$

(e) CodRNA, $\kappa = 0.5$ (f) a9a, $\kappa = 0.5$ (g) ijcnn1, $\kappa = 0.5$ (h) letter, $\kappa = 0.5$

(i) CodRNA, $\kappa = 5$ (j) a9a, $\kappa = 5$ (k) ijcnn1, $\kappa = 5$ (l) letter, $\kappa = 5$

Figure 2: Average computational time of four contrast algorithms under different setting.

Experimental Setup

Design of Experiments: In the experiments, we compare the computation time of different algorithms for computing the optimization problem (3) to verify the effectiveness of our algorithm. The active set technique (also called shrinking technique) is used by two of the most commonly used state-of-the-art SVM solvers, LIBSVM (Chang and Lin 2011) and SVMLight (Joachims 1999b). Similarly, the active set technique solves a smaller optimization problem (15) by screening inactive samples to reduce the computational time. However, these methods do not have theoretic guarantee w.r.t. whether a training sample can be safely remove. In the experiments, we combine our safe sample screening rules with active set technology to reduce the computational time. Specifically, we use our safe sample screening rules at the beginning of the experiment during which the screening operation is invoked every 10 iterations until the duality gap is smaller than 10^{-4}. Then, we use active set technique for the rest of the training process.

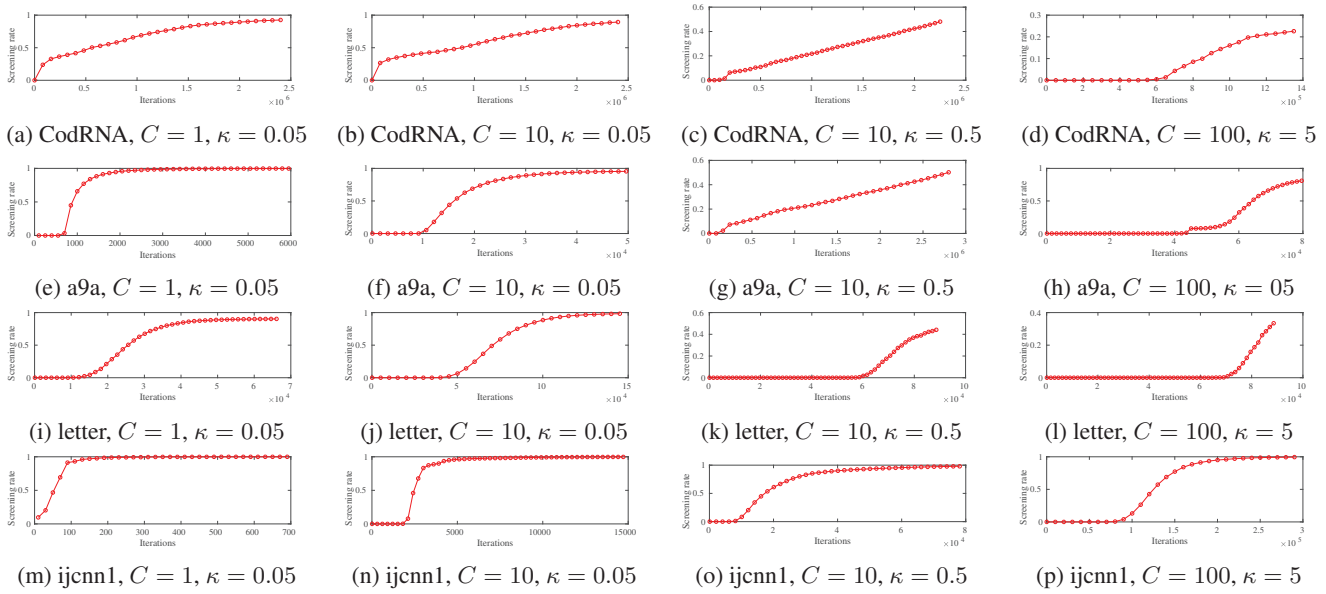
In addition, we compared our safe sample screening algorithm with traditional RSVM algorithm, which uses all the samples to train the model during the whole training process. The compared algorithms are summarized as follows.

1. Safe: Our proposed safe sample screening algorithm.
2. Shrink: The active set technique without safe screening guarantee (Chang and Lin 2011; Joachims 1999b).
3. Shrink+Safe: our safe sample screening rules combined with active set technique.
4. Non screening: The traditional RSVM algorithm with CCCP (Collobert et al. 2006).

Implementation: We implement our algorithm in MATLAB. For kernel, the linear kernel and Gaussian kernel $K(x_1, x_2) = \exp(-\kappa\|x_1 - x_2\|^2)$ are used in all experiments. The parameter C is selected from the set $\{0.1, 1, 10, 100\}$. The Gaussian kernel parameter κ is selected from the set $\{0.05, 0.5, 5\}$. The ramp loss function parameter s is fixed at 0. The optimization precision ϵ is set to be 10^{-8}. For each dataset, we randomly selected 20000 samples for training.

Datasets: Table 2 summarizes the four benchmark datasets (CodRNA, ijcnn1, a9a, letter) used in the experiments. There are from LIBSVM [1]sources. Originally, the Letter is a 26-class dataset (*i.e.*, the alphabet "A"-"Z"). We created a bi-

[1]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.



(a) CodRNA, $C = 1$, $\kappa = 0.05$ (b) CodRNA, $C = 10$, $\kappa = 0.05$ (c) CodRNA, $C = 10$, $\kappa = 0.5$ (d) CodRNA, $C = 100$, $\kappa = 5$

(e) a9a, $C = 1$, $\kappa = 0.05$ (f) a9a, $C = 10$, $\kappa = 0.05$ (g) a9a, $C = 10$, $\kappa = 0.5$ (h) a9a, $C = 100$, $\kappa = 05$

(i) letter, $C = 1$, $\kappa = 0.05$ (j) letter, $C = 10$, $\kappa = 0.05$ (k) letter, $C = 10$, $\kappa = 0.5$ (l) letter, $C = 100$, $\kappa = 5$

(m) ijcnn1, $C = 1$, $\kappa = 0.05$ (n) ijcnn1, $C = 10$, $\kappa = 0.05$ (o) ijcnn1, $C = 10$, $\kappa = 0.5$ (p) ijcnn1, $C = 100$, $\kappa = 5$

Figure 3: The screening rate of different datasets.

nary version of Letter dataset by classifying alphabet A to M versus N to Z.

Table 2: The benchmark datasets used in the experiments.

Dateset	Dimensionality	Samples	Source
CodRNA	8	59535	LIBSVM
a9a	123	32561	LIBSVM
letter	16	20000	LIBSVM
ijcnn1	22	49990	LIBSVM

Results and Discussions

Figure 2 presents the average computational time of four competing algorithms under different setting. Compared with *Non screening*, our safe sample screening rule can effectively reduce almost 50% computational time in most settings. The result clearly demonstrate that our safe sample screening rule combined with active set technology is the most efficient method for reducing computational time, significantly outperforming the standalone active set technique. This is because the active set technique is not safe, meaning that, when it erroneously screens useful samples, it needs to repeat the training after correcting those mistakes. Our safe sample screening rule can safely screen samples until close to the optimal solution. Then, when using the activity set technique, we can try to avoid screening active samples by mistake as much as possible. Even if some samples are wrongly screened, the retrained process only requires fewer samples.

Figure 3 presents the screening rate of different datasets. The results clearly demonstrate that when the Gaussiam kernel parameter is small, our safe sample screening rule can effectively screen half of the inactive samples at the beginning of training process. As the number of iterations increases, our safe sample screening rule can screen almost all inactive samples. In Figure 3 (d), (k), (I), we only screen a few inactive samples because the SVM model contains a lot of support vectors and is not sparse in samples at this setting.

Conclusion

In this paper, we propose two safe sample screening rules for RSVM based on the CCCP algorithm. Specifically, we first provide a screening rule for the inner solver of CCCP. Secondly, we provide a new rule for propagating screened samples between two successive solvers of CCCP. We also provide the security guarantee to our sample screening rules to RSVM. To the best of our knowledge, this is the first work of safe sample screening to a non-convex optimization problem. Experimental results on a variety of benchmark datasets verify that our safe sample screening rules can significantly reduce the computational time.

Acknowledgments

This work was supported by Six talent peaks project (No. XYDXX-042) and the 333 Project (No. BRA2017455) in Jiangsu Province and the National Natural Science Foundation of China (No: 61573191).

References

Bazaraa, M. S., and Shetty, C. M. 2012. *Foundations of optimization*, volume 122. Springer Science & Business Media.

Bertsekas, D. P. 1997. Nonlinear programming. *Journal of the Operational Research Society* 48(3):334–334.

Bertsekas, D. P. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

Bonnefoy, A.; Emiya, V.; Ralaivola, L.; and Gribonval, R. 2014. A dynamic screening principle for the lasso. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, 6–10. IEEE.

Bonnefoy, A.; Emiya, V.; Ralaivola, L.; and Gribonval, R. 2015. Dynamic screening: Accelerating first-order algorithms for the lasso and group-lasso. *IEEE Transactions on Signal Processing* 63(19):5121–5132.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2(3):27.

Collobert, R.; Sinz, F.; Weston, J.; and Bottou, L. 2006. Large scale transductive svms. *Journal of Machine Learning Research* 7(Aug):1687–1712.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

El Ghaoui, L.; Viallon, V.; and Rabbani, T. 2011. Safe feature elimination for the lasso. *Submitted, April*.

Fan, R.-E.; Chen, P.-H.; and Lin, C.-J. 2005. Working set selection using second order information for training support vector machines. *Journal of machine learning research* 6(Dec):1889–1918.

Fercoq, O.; Gramfort, A.; and Salmon, J. 2015. Mind the duality gap: safer rules for the lasso. *arXiv preprint arXiv:1505.03410*.

Ghaoui, L. E.; Viallon, V.; and Rabbani, T. 2010. Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219*.

Gu, B.; Shan, Y.; Geng, X.; and Zheng, G. 2018. Accelerated asynchronous greedy coordinate descent algorithm for svms. In *IJCAI*, 2170–2176.

Joachims, T. 1999a. Making large-scale support vector machine learning practical, advances in kernel methods. *Support vector learning*.

Joachims, T. 1999b. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund* 19(4).

Liu, J.; Zhao, Z.; Wang, J.; and Ye, J. 2013. Safe screening with variational inequalities and its application to lasso. *arXiv preprint arXiv:1307.7577*.

Ogawa, K.; Suzuki, Y.; Suzumura, S.; and Takeuchi, I. 2014. Safe sample screening for support vector machines. *arXiv preprint arXiv:1401.6740*.

Ogawa, K.; Suzuki, Y.; and Takeuchi, I. 2013. Safe screening of non-support vectors in pathwise svm computation. In *International conference on machine learning*, 1382–1390.

Platt, J. 1998. Sequential minimal optimization: A fast algorithm for training support vector machines.

Rakotomamonjy, A.; Gasso, G.; and Salmon, J. 2019. Screening rules for lasso with non-convex sparse regularizers. *arXiv preprint arXiv:1902.06125*.

Shen, X.; Tseng, G. C.; Zhang, X.; and Wong, W. H. 2003. On ψ-learning. *Journal of the American Statistical Association* 98(463):724–734.

Sriperumbudur, B. K., and Lanckriet, G. R. 2009. On the convergence of the concave-convex procedure. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, 1759–1767. Curran Associates Inc.

Vapnik, V. 2013. *The nature of statistical learning theory*. Springer science & business media.

Wang, J.; Zhou, J.; Wonka, P.; and Ye, J. 2013. Lasso screening rules via dual polytope projection. In *Advances in Neural Information Processing Systems*, 1070–1078.

Wang, J.; Zhou, J.; Liu, J.; Wonka, P.; and Ye, J. 2014. A safe screening rule for sparse logistic regression. In *Advances in neural information processing systems*, 1053–1061.

Wang, J.; Wonka, P.; and Ye, J. 2014. Scaling svm and least absolute deviations via exact data reduction. In *International conference on machine learning*, 523–531.

Wu, Y., and Liu, Y. 2007. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association* 102(479):974–983.

Xiang, Z. J.; Wang, Y.; and Ramadge, P. J. 2016. Screening tests for lasso problems. *IEEE transactions on pattern analysis and machine intelligence* 39(5):1008–1027.

Xu, P., and Ramadge, P. J. 2013. Three structural results on the lasso problem. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3392–3396. IEEE.

Xu, L.; Crammer, K.; and Schuurmans, D. 2006. Robust support vector machine training via convex outlier ablation. In *AAAI*, volume 6, 536–542.

Yuille, A. L., and Rangarajan, A. 2003. The concave-convex procedure. *Neural computation* 15(4):915–936.

Zimmert, J.; de Witt, C. S.; Kerg, G.; and Kloft, M. 2015. Safe screening for support vector machines. In *NIPS 2015 Workshop on Optimization in Machine Learning (OPT)*.