Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Integrating outlier filtering in large margin training

Xi-chuan ZHOU[†1], Hai-bin SHEN[2], Jie-ping YE[3]

([1]*College of Communication Engineering, Chongqing University, Chongqing 400044, China*)

([2]*School of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*)

([3]*Department of Computer Science and Engineering, Arizona State University, Tempe 85281, USA*)

[†]E-mail: zxc@ccee.cqu.edu.cn

Received Oct. 15, 2010; Revision accepted Feb. 23, 2011; Crosschecked Apr. 7, 2011

Abstract: Large margin classifiers such as support vector machines (SVM) have been applied successfully in various classification tasks. However, their performance may be significantly degraded in the presence of outliers. In this paper, we propose a robust SVM formulation which is shown to be less sensitive to outliers. The key idea is to employ an adaptively weighted hinge loss that explicitly incorporates outlier filtering in the SVM training, thus performing outlier filtering and classification simultaneously. The resulting robust SVM formulation is non-convex. We first relax it into a semi-definite programming which admits a global solution. To improve the efficiency, an iterative approach is developed. We have performed experiments using both synthetic and real-world data. Results show that the performance of the standard SVM degrades rapidly when more outliers are included, while the proposed robust SVM training is more stable in the presence of outliers.

Key words: Support vector machines, Outlier filter, Semi-definite programming, Multi-stage relaxation
doi:10.1631/jzus.C1000361 **Document code:** A **CLC number:** TP301

1 Introduction

The support vector machine (SVM), as a large margin learning approach, has been widely accepted to be one of the most effective techniques designed for classification. The basic procedure is to find the hyperplane that separates two classes of data points with the largest margin space. It is intuitive and proven in theory that maximizing the margin could yield the greatest robustness to noise and reduce the possibility of future misclassification (Cortes and Vapnik, 1995; Bousquet and Elisseeff, 2002; Scholkopf and Smola, 2002).

The phenomenon of outliers frequently occurs in many machine learning applications. It can arise due to mechanical faults, changes in system behavior, fraudulent behavior, human error, instrument error, or simply through natural deviations in populations. However, the naive large margin principle yields poor results over the data set contaminated by outliers. This is because outliers tend to have the largest margin loss, thus playing an important role in determining the separating hyperplane. In fact, a single outlier with a relatively large hinge loss would significantly decrease the accuracy of SVM classification.

Previous attempts have been made to improve the robustness of the SVM training for outliers. Krause and Singer (2004) investigated the robust margin loss of SVM that ceased to increase the penalty after a certain point. A robust SVM objective was proposed by Song *et al.* (2002) that could scale the margin space with a heuristic weight. Herbrich and Weston (2000) formulated a new training objective based on minimizing a bound on the leave-one-out cross validation error of the soft margin SVM. Wu and Liu (2007) incorporated a truncated hinge loss in SVM, which makes the formulation more robust than the standard SVM formulation. Xu *et al.* (2006) also modified the hinge loss function of SVM, which improved SVM's

classification robustness against outliers. One property the above approaches share is that they do not attempt to identify outliers, but rather try to reduce the effect of misclassified points with optimization or heuristic approaches.

On the other hand, outlier detection, as a different research area, has played a more and more important role in many machine learning applications. Most previous work focused on the unsupervised case (Brodley and Friedl, 1996; Fawcett and Provost, 1997; Tax *et al.*, 1999; Steinwart *et al.*, 2005). More recently, a one-class SVM was proposed to detect outliers (Ratsch *et al.*, 2002). Tax (2001) calculated the smallest hyper-sphere to contain all the normal data points. Similar approaches have been proposed for different applications (Eskin *et al.*, 2001; Davy and Godsill, 2002; King *et al.*, 2002). Manevitz and Yousef (2002) and Laskov *et al.* (2004) proposed more variants of this technique. Relevant work in outlier detection concerning SVM classification also includes Tax *et al.* (1999), which uses SVMs for outlier detection. Thongkam *et al.* (2008) proposed a C-support vector classification filter to identify and remove the misclassified instances. The basic idea of these approaches is to handle outlier detection as a classification problem, with pre-labeled normal and abnormal data (outliers) in isolated areas. They were designed for outlier detection rather than classifying labeled data.

In this paper we propose a novel formulation which integrates outlier detection and removal directly in the SVM training. Different from previous work, the outlier filtering task, commonly performed as a separate preprocessing step, is incorporated in the SVM training phase. The proposed robust SVM formulation is non-convex. We first propose a semi-definite programming (SDP) relaxation, which yields a global optimal solution. However, the SDP formulation is computationally expensive to solve. To improve the efficiency, we develop a multi-stage relaxation of the original formulation, which leads to an iterative algorithm and involves the standard SVM training at each iteration.

2 Robust SVM training with outliers

Given a set of n training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is drawn from a domain $\mathcal{X}$ and each of the label y_i is an integer from $\mathcal{Y} = \{-1, 1\}$, the goal of the binary-class classification in SVM is to learn a model that assigns the correct label to an unseen test sample. For non-separable data points, the soft margin SVM employs a hinge loss (Fig. 1a) of the following form for each training point:

$$\xi_i = \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i),$$

where $\boldsymbol{w}$ is the optimization variable. The SVM minimizes the following regularized loss function in the training phase:

$$\min_{\boldsymbol{w}} \left(\frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \right). \quad (1)$$

The separating plane of the SVM is determined by the support vectors whose hinge loss ξ_i exceeds zero. Generally speaking, the larger the hinge loss is, the more it will affect the resulting separating plane.

Next, we use a synthetic example to explain how a single outlier could affect the SVM. We consider the binary classification, where each class consists of 50 Gaussian distributed instances. Let the class mean be $\boldsymbol{\mu}_1 = (0, 1)$ and $\boldsymbol{\mu}_{-1} = (0, -1)$. The covariance matrices of both classes are identity matrices. The outlier $(\boldsymbol{x}, -1)$ is randomly sampled on the ring of $\|\boldsymbol{x} - \boldsymbol{\mu}_1\| = r$ and added to the training set. The process is repeated 100 times and the average test error over 20 test points is calculated and shown in Fig. 1b. As one can see, the error rate increases when r increases.

To detect and remove outliers, we incorporate an adaptive weight β_i to the hinge loss as

$$\widetilde{\xi}_i = \beta_i \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i), \quad \beta_i \in \{1, 0\}. \quad (2)$$

Intuitively, the weight β_i equals zero if $\boldsymbol{x}_i$ is an outlier; otherwise, β_i equals one. The resulting robust

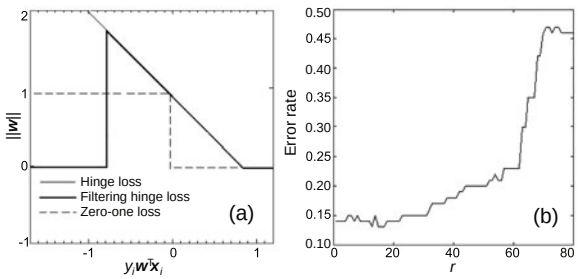

Fig. 1 Comparison of losses as a function of $y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i$ (a) and the average test error of SVM trained with a single outlier (b). In (a), the adaptively weighted hinge loss drops to zero if it is too large, which gives our approach the ability to detect and remove outliers

SVM formulation is given below:

$$\min_{\boldsymbol{w}} \min_{\boldsymbol{\beta}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \beta_i \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \right), \tag{3}$$

where $\beta_i \in \{0, 1\}$. The incorporation of variable β_i alone will not work, because the minimum SVM loss is achieved when all β_i's are zero. Thus, assuming there are at least M normal points out of n training points (M can be estimated from the data via cross-validation), we can add an extra constraint as follows:

$$\sum_{i=1}^{n} \beta_i \geq M.$$

By incorporating this constraint, we can remove $n - M$ potential outliers from the training set. One challenge of directly minimizing the objective function in Eq. (3) is its non-convexity in $\boldsymbol{w}$ and $\boldsymbol{\beta}$.

There are several methods to handle non-convex optimizations. One approach is to relax it into a convex problem to obtain a global optimal solution. The second approach is the alternating method, which optimizes different variables in turn and yields local minimal points. The advantage of the first method is its reproducibility and sound theoretical basis for convex optimization, while the second approach is in general more efficient.

3 Semi-definite programming relaxation

The proposed robust SVM formulation in Eq. (3) is non-convex. In this section, we relax it into a semi-definite programming, which admits a global solution. Eq. (3) can be equivalently rewritten as

$$\min_{\boldsymbol{w}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \min_{\beta_i \in \{0,1\}} \left(\beta_i \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \right) \right). \tag{4}$$

For the inner minimization in Eq. (3),

$$\min_{\beta_i \in \{0,1\}} \left(\beta_i \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \right),$$

we note that the integer constraint on the variables may be relaxed to $0 \leq \boldsymbol{\beta}_i \leq 1$ without changing the optimum. This is true since the minimization is over a linear function, the optimum will be at the vertices, and is therefore integral. We can equivalently reformulate the above equation as

$$\min_{\beta_i \in [0,1]} \left(\beta_i \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \right).$$

Thus, the formulation in Eq. (3) can be reformulated as

$$\min_{\boldsymbol{w}} \min_{\beta_i \in [0,1]} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \beta_i \max(0, 1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \right) \quad \text{s.t. } \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\beta} \geq M, \tag{5}$$

where $\boldsymbol{e} \in \mathbb{R}^n$ is a column vector of ones. The formulation in Eq. (5) can be further reformulated as the following constrained optimization:

$$\min_{\boldsymbol{w}, \boldsymbol{\xi}} \min_{\boldsymbol{\beta}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \beta_i \xi_i \right) \quad \text{s.t. } \forall i, y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i \geq 1 - \xi_i, \quad \boldsymbol{\xi} \geq 0,\ 0 \leq \boldsymbol{\beta} \leq 1,\ \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\beta} \geq M. \tag{6}$$

The challenge of solving the above formulation is that it is not jointly convex in $\boldsymbol{w}$ and $\boldsymbol{\beta}$. In the following, we show how to relax it into a convex problem. To this end, Eq. (6) is first reformulated as a min-max problem, as summarized in the following theorem:

Theorem 1 Suppose $\boldsymbol{Y}$ is a diagonal matrix whose ith diagonal entry is given by $Y_{ii} = y_i$. The minimization problem in Eq. (6) is equivalent to

$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{\beta} - \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \left((\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}) \circ (\boldsymbol{Y}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}) \right) \boldsymbol{\alpha} \right) \quad \text{s.t. } 0 \leq \boldsymbol{\alpha} \leq C,\ 0 \leq \boldsymbol{\beta} \leq 1,\ \boldsymbol{e}^{\mathrm{T}} \boldsymbol{\beta} \geq M, \tag{7}$$

where '$\circ$' denotes the point-wise product between two matrices.

Proof Let $\zeta_i = \beta_i \xi_i$. For a fixed $\boldsymbol{\beta}$ satisfying $\boldsymbol{e}^{\mathrm{T}} \boldsymbol{\beta} \geq M$, the formulation in Eq. (6) is equivalent to

$$\min_{\boldsymbol{w}, \boldsymbol{\zeta}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \zeta_i \right) \quad \text{s.t. } \forall i, \beta_i (1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) \leq \zeta_i,\ \zeta_i \geq 0. \tag{8}$$

The Lagrangian of the above formulation can be written as

$$L_1 = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \zeta_i + \sum_{i=1}^{n} \alpha_i \left(\beta_i (1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i) - \zeta_i \right) - \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\zeta}.$$

Computing its gradient with respect to ζ_i yields $C - \alpha_i - \mu_i = 0$, thus $\alpha_i \leq C$ when $\mu_i \geq 0$. Then L_1 can be equivalently expressed as

$$L_2 = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n} \alpha_i \beta_i (1 - y_i \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i).$$

Finally, taking the gradient with respect to $\boldsymbol{w}$ yields $\boldsymbol{w} = \sum_{i=1}^{n} \beta_i y_i \boldsymbol{x}_i \alpha_i$. This leads to $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} = \boldsymbol{\alpha}^{\mathrm{T}}((\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}) \circ (\boldsymbol{Y}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}))\boldsymbol{\alpha}$. We substitute $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ back into L_2 and write the dual form of Eq. (8) as

$$\max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \left((\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}) \circ (\boldsymbol{Y}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}) \right) \boldsymbol{\alpha} \right) \quad \text{s.t. } 0 \le \boldsymbol{\alpha} \le C. \tag{9}$$

Due to the strong duality, we have

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \zeta_i \right) = \max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \left((\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}) \circ (\boldsymbol{Y}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}) \right) \boldsymbol{\alpha} \right).$$

Substituting the dual form of Eq. (8) into Eq. (6) yields the result of the theorem. □

Note that we reformulate Eqs. (6) and (7) to rewrite the inner optimization as a maximum. This technique allows further convex reformulation of the outer minimization. The intuitive observation is that $\boldsymbol{\beta}$ appears only as $\boldsymbol{\beta}$ and $\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}$. Thus, by creating a matrix variable $\boldsymbol{D} = \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}$, the problem can be reformulated as a maximum of the linear combinations of $\boldsymbol{\beta}$ and $\boldsymbol{D}$, resulting in the following min-max problem:

$$\min_{0\le\boldsymbol{\beta}\le 1, \boldsymbol{D}=\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}} \max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{D} \circ \boldsymbol{Q})\boldsymbol{\alpha} \right) \quad \text{s.t. } 0 \le \boldsymbol{\alpha} \le C,\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\beta} \ge M,$$

where $\boldsymbol{Q} = \boldsymbol{Y}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}$. The only problem that remains is that $\boldsymbol{D} = \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}$ is a non-convex quadratic constraint. A common strategy is to relax the equality to $\boldsymbol{D} \succeq \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}$, which leads to the following convex problem:

$$\min_{0\le\boldsymbol{\beta}\le 1} \min_{\boldsymbol{D}\succeq\boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}}} \max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{D} \circ \boldsymbol{Q})\boldsymbol{\alpha} \right) \quad \text{s.t. } 0 \le \boldsymbol{\alpha} \le C,\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\beta} \ge M. \tag{10}$$

The above formulation can be equivalently written as an SDP problem:

Theorem 2 Solving Eq. (10) is equivalent to solving the following semi-definite programming:

$$\begin{aligned} &\min_{t,\boldsymbol{\nu},\boldsymbol{\lambda},\boldsymbol{\beta},\boldsymbol{D}} \ t \\ &\text{s.t. } \boldsymbol{D} \succeq \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}},\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\beta} \ge M, \\ &\quad \boldsymbol{\nu} \ge 0,\ \boldsymbol{\lambda} \ge 0,\ 0 \le \boldsymbol{\beta} \le 1, \\ &\begin{pmatrix} 2(\boldsymbol{D} \circ \boldsymbol{Q}) & \boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda} \\ \boldsymbol{\beta}^{\mathrm{T}} - \boldsymbol{\nu}^{\mathrm{T}} + \boldsymbol{\lambda}^{\mathrm{T}} & t - C\sum_{i=1}^{n} \nu_i \end{pmatrix} \succeq 0. \end{aligned} \tag{11}$$

Proof The formulation in Eq. (10) is equivalent to

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{D},t} \ t \\ &\text{s.t. } t \ge \max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{D} \circ \boldsymbol{Q})\boldsymbol{\alpha} \right) \\ &\quad \boldsymbol{D} \succeq \boldsymbol{\beta}\boldsymbol{\beta}^{\mathrm{T}},\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\beta} \ge M, \\ &\quad 0 \le \boldsymbol{\alpha} \le C,\ 0 \le \boldsymbol{\beta} \le 1. \end{aligned} \tag{12}$$

We can reformulate the maximization in the first constraint of the above optimization as

$$\max_{\boldsymbol{\alpha}} \left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\alpha} \right) \text{ s.t. } 0 \le \boldsymbol{\alpha} \le C, \tag{13}$$

where $\boldsymbol{G} = \boldsymbol{D} \circ \boldsymbol{Q}$. Let the Lagrangian of the above maximization be

$$L = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\alpha} + \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\alpha} + \boldsymbol{\nu}^{\mathrm{T}}(C\boldsymbol{e} - \boldsymbol{\alpha}).$$

Assuming $\boldsymbol{G} \succeq 0$, at the optimum we have

$$\boldsymbol{\alpha} = \boldsymbol{G}^{-1}(\boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda}),$$

and we can write the dual form of Eq. (13) as

$$\min_{\boldsymbol{\nu},\boldsymbol{\lambda}} \left((\boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda})^{\mathrm{T}}(2\boldsymbol{G})^{-1}(\boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda}) + C\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{e} \right) \quad \text{s.t. } \boldsymbol{\nu} \ge 0, \boldsymbol{\lambda} \ge 0. \tag{14}$$

The optimal objectives of the primal-dual problems are equal due to the strong duality. This implies that for any $t > 0$, the first constraint in Eq. (12) holds if and only if there exist $\boldsymbol{\nu} \ge 0, \boldsymbol{\lambda} \ge 0$ such that

$$(\boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda})^{\mathrm{T}}(2\boldsymbol{G})^{-1}(\boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda}) + C\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{e} \le t,$$

or, equivalently (by the Schur complement),

$$\begin{pmatrix} 2\boldsymbol{G} & \boldsymbol{\beta} - \boldsymbol{\nu} + \boldsymbol{\lambda} \\ \boldsymbol{\beta}^{\mathrm{T}} - \boldsymbol{\nu}^{\mathrm{T}} + \boldsymbol{\lambda}^{\mathrm{T}} & t - C\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{e} \end{pmatrix} \succeq 0$$

holds. It follows that Eq. (10) can be expressed as Eq. (11), which proves the theorem. □

4 Multi-stage relaxation

The SDP based formulation in Section 3 admits a global optimal solution; however, it is computationally expensive to solve. In this section, we propose an iterative method based on the multi-stage relaxation of Eq. (3). The multi-stage relaxation of non-convex problems is based on the concave duality property, which was employed in Zhang (2008) to solve a sparse learning problem.

4.1 Concave duality

The multi-stage relaxation considers the following optimization formulation:

$$\min_{\boldsymbol{w}} \left(R_0(\boldsymbol{w}) + CR_1(\boldsymbol{w})\right), \tag{15}$$

where $R_0(\boldsymbol{w})$ is convex in $\boldsymbol{w}$ and $R_1(\boldsymbol{w})$ is non-convex. We shall rewrite $R_1(\boldsymbol{w})$ using concave duality. Let $\boldsymbol{h}(\boldsymbol{w})$ be a vector function with range Ω. Assume that there exists a function $g(\boldsymbol{u})$ defined on Ω so that we can express $R_1(\boldsymbol{w})$ as

$$R_1(\boldsymbol{w}) = g(\boldsymbol{h}(\boldsymbol{w})).$$

Assume that we can find $\boldsymbol{h}(\boldsymbol{w})$ so that the function $g(\boldsymbol{u})$ is concave on $\boldsymbol{u} \in \Omega$. Under this assumption, we can rewrite the $R_1(\boldsymbol{w})$ as

$$R_1(\boldsymbol{w}) = \inf_{\boldsymbol{v}\in\Phi}[\boldsymbol{v}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{w}) + g^*(\boldsymbol{v})] \tag{16}$$

using concave duality. In this case $g^*(\boldsymbol{v})$ is the concave dual of $g(\boldsymbol{u})$:

$$g^*(\boldsymbol{v}) = \inf_{\boldsymbol{u}\in\Omega}[-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{u} + g(\boldsymbol{u})], \tag{17}$$

and the minimum of the right hand side of Eq. (16) is achieved at

$$\hat{\boldsymbol{v}} = \nabla_{\boldsymbol{u}} g(\boldsymbol{u})|_{\boldsymbol{u}=\boldsymbol{h}(\boldsymbol{w})}. \tag{18}$$

Given the vector function $\boldsymbol{h}(\boldsymbol{w})$ defined above and a fixed vector $\boldsymbol{v}$, a simple convex relaxation of Eq. (15) becomes

$$\min_{\boldsymbol{w}} \left(R_0(\boldsymbol{w}) + C\boldsymbol{v}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{w})\right). \tag{19}$$

This simple relaxation yields a solution that is different from the solution of Eq. (15). However, since the robust SVM satisfies the condition mentioned in Section 3, it is possible to write $R_1(\boldsymbol{w})$ using Eq. (16). With this new representation, we can write Eq. (15) as

$$\min_{\boldsymbol{w},\boldsymbol{v}} \left(R_0(\boldsymbol{w}) + C\left(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{w}) + g^*(\boldsymbol{v})\right)\right). \tag{20}$$

The above formulation is equivalent to the one in Eq. (15) because of Eq. (16). If we could find a good approximation of $\boldsymbol{v}$, which improves from the initial choice of $\boldsymbol{v} = 1$, then the above formulation can lead to a refined convex problem in $\boldsymbol{w}$ that is expected to be a better convex relaxation compared with the one in Eq. (19). In practice, the approximation of $\boldsymbol{v}$ can be derived by Eq. (18), when the derivative exists.

4.2 Algorithm

For robust SVM training, we consider the optimization in Eq. (3), which is equivalent to

$$\min_{\boldsymbol{w}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \beta_i \max(0, 1 - y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)\right),$$

where $\boldsymbol{\beta} \in \Delta$ with $\Delta = \{\boldsymbol{\beta} | \beta_i \in \{0,1\}, \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\beta} \geq M, i = 1, 2, \ldots, n\}$. To explain the concave duality in robust SVM training, we introduce the following notations:

Definition 1 Let $R_0(\boldsymbol{w})$, $R_1(\boldsymbol{w})$ be defined as

$$R_0(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2, R_1(\boldsymbol{w}) = \min_{\boldsymbol{\beta}\in\Delta} \sum_{i=1}^{n} \beta_i \max(0, 1 - y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i).$$

The non-convex function $R_1(\boldsymbol{w})$ can be represented by $g(\boldsymbol{h}(\boldsymbol{w}))$, where

$$g(\boldsymbol{u}) = \inf_{\boldsymbol{\beta}\in\Delta} \left(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{u}\right), \; h_i(\boldsymbol{w}) = \max(0, 1 - y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i).$$

With the above definition, Eq. (3) can be equivalently reformulated as Eq. (15). Moreover, $g(\boldsymbol{u})$ is concave in $\boldsymbol{u}$, because it is the point-wise infimum of a set of linear functions. Therefore, we can approximate Eq. (3) by the multi-stage relaxation. Specifically, our iterative method exploits the concave duality to improve the solution. The algorithm is summarized below:

Initialization: Set $\hat{\boldsymbol{v}} = 1$.

1. Fix $\boldsymbol{v} = \hat{\boldsymbol{v}}$ and calculate $\hat{\boldsymbol{w}}$ by solving

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{n} \hat{v}_i \max(0, 1 - y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)\right).$$

2. Fix $\boldsymbol{w} = \hat{\boldsymbol{w}}$ and calculate $\widehat{u}_i = \max(0, 1 - y_i\hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i)$. Suppose $\widehat{u}_1, \widehat{u}_2, \ldots, \widehat{u}_n$ are arranged in ascending order and we denote the order of $\widehat{u}_i$ by $s(i)$. Then set $\hat{v}_i = I(M - s(i))$, where I is the indicator function.

Note that the above algorithm begins with the standard SVM formulation by initializing $\boldsymbol{v} = 1$. Thus, by repeating the above iterations, the multi-stage robust SVM could converge to a solution better than the standard SVM. Another point worth noting is that the function $g(\boldsymbol{u})$ in Definition 1 is a piecewise linear function of $\boldsymbol{u}$, and its derivative over $\boldsymbol{u}$ exists when $\forall i \neq j \in \{1, 2, \ldots, n\}, \widehat{u}_i \neq \widehat{u}_j$. In practice, if $\widehat{u}_i = \widehat{u}_j$ and $i < j$, we choose $\boldsymbol{x}_i$ over $\boldsymbol{x}_j$ to make the result reproducible.

5 Experiment

A series of experiments were constructed on synthetic and real data sets to verify the effectiveness of the proposed algorithms. In all the experiments, different proportions of the training points were selected to generate outliers. Specifically, the labels of the selected points were changed manually to form outliers. For binary classification, the label was changed from -1 to $+1$ and vice versa. The modified data points become outliers in synthetic data sets when the data points of different classes are distant. For multi-class classification, we applied the one-against-the-rest strategy, and transformed the multi-class problem to several binary classification problems.

The first experiment was conducted on synthetic data. We assigned one Gaussian for each class, with the first given by $\boldsymbol{\mu} = [0.8, 0.8]$, $\boldsymbol{\Sigma} = \boldsymbol{I}$ and the second by $-\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The training set contained 60 instances, 30 from each Gaussian. A testing set was constructed that contained 60 instances drawn from the same distribution as the training data. Different proportions (2.5%–20%) of training points were randomly selected as outliers, and we changed their labels. The experiments were repeated 50 times. In each repetition a training set and corresponding outlier set were constructed. The experiment results were averaged with 95% confidence interval. We tested the algorithms proposed in this paper: SDP robust SVM (SDP RSVM) and multi-stage robust SVM (multi-stage RSVM). We compared the results with those of standard soft margin SVM and the adaptive margin SVM (Song *et al.*, 2002). The candidate set for the generalization tradeoff parameter C is $\{10^i\}_{i=-2}^{4}$. The SDP RSVM was implemented with the outlier parameter selected from the following candidate set: $M \in \{10i\}_{i=0}^{6}$. For the multi-stage RSVM, we stopped the iterative process (i.e., the algorithm converges) when the change of the objective values is less than 10^{-4}.

Fig. 2 shows the results for our robust SVM training methods and related existing algorithms. We can observe from the figure that the SVM is sensitive to outliers. The error rate increases significantly when the proportion of outliers increases. In contrast, the robust SVM algorithms are less sensitive to outliers. This is illustrated by the mean test error. The proposed algorithms outperform the adaptive margin SVM.

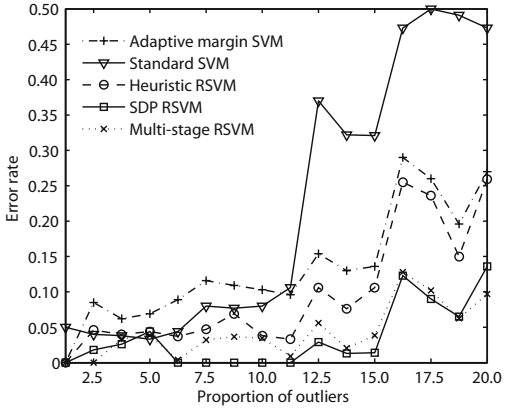

Fig. 2 Comparison of algorithms in terms of the classification error rate with different proportions of outliers

We also compared SDP RSVM and multi-stage RSVM with both a supervised approach (C-SVM) and an unsupervised approach (consensus filter) for outlier detection. The results are shown in Fig. 3. Overall, our methods have better detection accuracy than C-SVM (Thongkam *et al.*, 2008) and consensus filtering (Brodley and Friedl, 1996). The detection accuracy drops when the proportion of the outliers increases. This phenomenon is expected because more 'normal' points may be misclassified when outliers are introduced. Thus, the algorithm may predict a normal but misclassified point as an outlier.

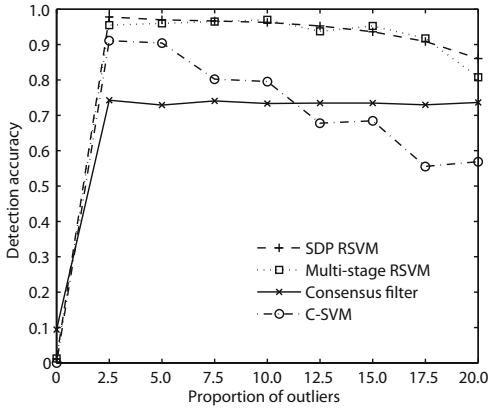

Fig. 3 Comparison of different methods in terms of outlier detection accuracy

Next, we studied the sensitivity of the proposed algorithm to the parameter C. We compared SDP RSVM and multi-stage RSVM with standard SVM.

Five percent of training points were randomly flipped as outliers. Seven values of C, from 10^{-2} to 10^4, were tested. The results are shown in Fig. 4. Our results show that the multi-stage RSVM has similar dependence on C to the standard SVM. On the other hand, the SDP RSVM is less sensitive to the parameter C.

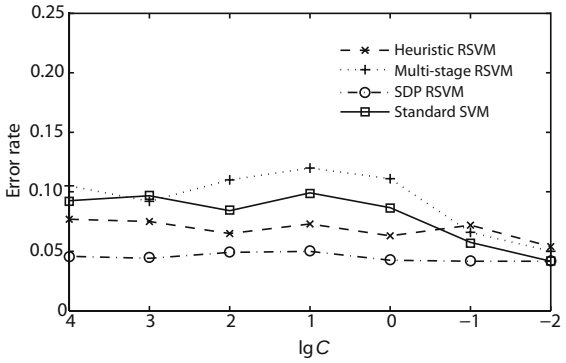

Fig. 4 Classification error rate on the synthetic data set with different values of the generalization tradeoff parameter C

We also evaluated the sensitivity of SDP RSVM and multi-stage RSVM to parameter M using the receiver operating characteristic (ROC) curves of the classification accuracy. Fig. 5 shows the ROC curve for the multi-stage RSVM with various values of M. In this experiment, 90% (54 points) of the training data were normal points. When M decreases, the ROC curve becomes worse. This indicates that M is consistent with the actual number of the normal points. Similar phenomena can be found for SDP RSVM, whose figure is omitted here due to lack of space. Note that M can be automatically estimated from the data via cross-validation in practice.

In the above experiments, outliers were generated by changing labels. For comparison, we performed a set of experiments in which outliers were added in the synthetic data set. The same clean data points were drawn from Gaussian distributions as in above experiments. Outliers were uniformly drawn from a ring whose inner-radius is R and outer-radius is $R + 0.5$. The outliers were randomly labeled with even probability. Results of the experiments are summarized in Table 1.

As one can see, in both the experiments when outliers were drawn on the ring or generated by changing labels, the proposed algorithms outperform the standard SVM in the presence of outliers. SDP RSVM has higher classification accuracy and outlier

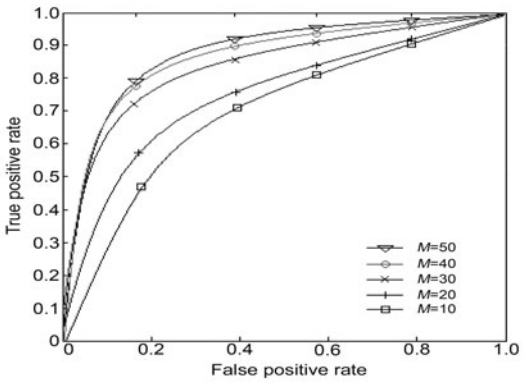

Fig. 5 The receiver operating characteristic (ROC) curve of multi-stage robust SVM with different numbers of normal points (M)

detection rate for the synthetic data set, and is also less sensitive to the parameter C. Due to higher computational complexity, SDP RSVM is preferred for medium size data sets. Specifically, SDP RSVM can be trained in $O(n^3)$ time by using incomplete Cholesky decomposition of the semi-definite matrix. Multi-stage RSVM is much faster and can be trained in $O(cd \times v)$ time, where c is a constant (usually less than 10) and v is the number of support vectors. Note that, multi-stage RSVM acquired better robustness at the expense of $c - 1$ times more computational complexity than the standard SVM.

Table 1 Comparison of different methods in terms of error rate when outliers were added in the synthetic data set

R	Algorithm	Error rate (%)		
		N_{out} =5%	N_{out} =10%	N_{out} =15%
2	SSVM	4.6±0.6	9.1±1.2	15.2±1.7
	AM SVM	3.0±0.4	7.3±0.5	13.7±1.4
	SDP RSVM	1.5±0.2	5.4±0.5	5.6±0.5
	MS RSVM	2.2±0.3	6.8±0.7	5.3±0.3
3	SSVM	5.7±0.5	12.3±1.4	25.7±3.7
	AM SVM	4.0±0.5	6.0±7.8	11.3±1.8
	SDP RSVM	1.6±0.2	4.4±0.5	6.3±0.7
	MS RSVM	2.9±0.1	5.4±0.6	7.4±0.9
4	SSVM	11.0±1.9	21.3±3.4	37.5±4.2
	AM SVM	7.3±0.8	11.3±1.3	14.2±1.5
	SDP RSVM	2.2±0.2	5.3±0.6	5.7±0.6
	MS RSVM	3.3±0.4	7.4±0.9	8.9±1.2
5	SSVM	13.2±1.5	35.5±4.4	47.9±7.8
	AM SVM	6.2±0.7	18.4±2.3	19.8±2.0
	SDP RSVM	3.2±0.4	9.3±1.2	9.4±1.3
	MS RSVM	4.4±0.5	13.3±2.2	14.9±2.2

R: inner radius of the ring; N_{out}: proportion of outliers. SSVM: standard SVM; AM SVM: adaptive margin SVM; MS RSVM: multi-stage RSVM

Table 2 Comparison of different methods in terms of the classification error rate with different proportions of outliers*

Data set	Algorithm	Detection rate (%)			Error rate (%)	
		No outliers	5% outliers	10% outliers	5% outliers	10% outliers
Iris**	Standard SVM	5.11±0.41			13.23±0.81	15.78±1.21
	Adaptive margin SVM	12.21±0.01			12.08±0.72	14.08±0.93
	SDP RSVM	5.31±0.22	91.21±4.01	85.05±7.42	5.93±0.35	5.01±0.34
	Multi-stage RSVM	5.05±0.21	88.03±5.01	77.74±5.01	10.33±4.01	10.53±0.70
Lung cancer**	Standard SVM	0.00±0.00			8.13±0.42	10.13±0.21
	Adaptive margin SVM	0.00±0.00			9.08±0.89	9.91±0.93
	SDP RSVM	0.00±0.00	85.35±6.23	81.25±5.42	7.93±0.55	7.42±0.84
	Multi-stage RSVM	0.00±0.00	82.56±4.23	72.51±7.21	9.43±5.34	9.12±0.90
Musk	Standard SVM	0.00±0.00			20.13±1.81	30.71±2.35
	Adaptive margin SVM	9.13±2.01			10.28±1.92	11.08±1.93
	Multi-stage RSVM	0.00±0.00	78.22±4.31	69.54±3.01	13.78±1.01	18.77±2.70
Hill valley	Standard SVM	0.00±0.00			13.23±0.81	21.78±1.21
	Adaptive margin SVM	2.11±0.12			11.31±0.52	14.11±0.78
	Multi-stage RSVM	0.00±0.00	80.21±6.04	71.21±6.32	10.76±0.77	18.55±0.65
Breast cancer	Standard SVM	3.21±0.34			3.89±0.21	4.98±0.34
	Adaptive margin SVM	4.92±0.38			4.03±0.51	4.35±0.75
	Multi-stage RSVM	3.34±0.43	79.21±3.78	72.41±6.35	3.22±0.35	3.39±0.42
Ionosphere	Standard SVM	0.00±0.00			3.73±0.21	4.95±0.65
	Adaptive margin SVM	2.15±0.30			3.78±0.72	5.08±0.51
	Multi-stage RSVM	0.00±0.00	75.03±7.33	72.13±6.43	2.13±0.11	4.53±0.20

* For efficiency reasons, SDP RSVM was not evaluated for some of these data sets; for all data sets, multi-stage RSVM converges in less than 10 iterations. The data sets marked with ** are multi-class data sets, and the rest are binary-class data sets

Finally, we performed experiments on real data. We compared different robust SVM training algorithms over six classification data sets from UCI (Frank and Asuncion, 2010). Two of them are multi-class data sets (Iris and Lung cancer), and four of them are binary-class data sets. For efficiency reasons, the SDP RSVM was not evaluated for some of these data sets. For all data sets, the multi-stage RSVM converges in less than 10 iterations. The results are summarized in Table 2. We can observe that the results on real data are consistent with those for the synthetic data set. The SVM shows excellent results with small label noises, but the robust SVM training algorithms outperform SVM when more outliers (over 5%) are included.

6 Conclusions

In this paper we propose two revised SVM training algorithms that are robust to the presence of outliers. The key idea is to employ an adaptively weighted hinge loss that explicitly incorporates outlier filtering in SVM training. To solve the proposed robust SVM formulation, we develop two algorithms based on SDP relaxation and multi-stage relaxation. SDP relaxation can obtain the global optimal point at the cost of expensive computation; in contrast, multi-stage relaxation yields a more efficient algorithm for computing a local solution.

We have performed experiments using both synthetic and real-world data. Results demonstrate the effectiveness of the proposed algorithms. In the future, we plan to study the theoretical properties of the multi-stage robust SVM formulation using ideas from Zhang (2008). Another interesting extension of our work is nonlinear generalization. The SDP formulation proposed here cannot be directly generalized using kernel trick due to the parameter β_i we introduce. But the multi-stage relaxation algorithm can be generalized to a nonlinear situation.

Specifically, given v_i, the first step of the multi-stage algorithm is similar to that of the standard SVM. Thus, it can use the kernel trick to handle nonlinearly separable data sets.

References

Bousquet, O., Elisseeff, A., 2002. Stability and generalization. *J. Mach. Learn. Res.*, **2**(3):499-526. [doi:10.1162/153244302760200704]

Brodley, C.E., Friedl, M.A., 1996. Identifying and Eliminating Mislabeled Training Instances. Proc. 13th National Conf. on Artificial Intelligence, **1**:799-805.

Cortes, C., Vapnik, V., 1995. Support vector networks. *Mach. Learn.*, **20**(3):273-297. [doi:10.1023/A:1022627411411]

Davy, M., Godsill, S., 2002. Detection of Abrupt Spectral Changes Using Support Vector Machines: an Application to Audio Signal Segmentation. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, p.1313-1316.

Eskin, E., Lee, W., Stolfo, S.J., 2001. Modeling System Calls for Intrusion Detection with Dynamic Window Sizes. Proc. DARPA Information Survivability Conf. and Exposition, p.1-11.

Fawcett, T., Provost, F.J., 1997. Adaptive fraud detection. *Data Min. Knowl. Disc.*, **1**(3):291-316. [doi:10.1023/A:1009700419189]

Frank, A., Asuncion, A., 2010. UCI Machine Learning Repository. School of Information and Computer Science, University of California, Irvine.

Herbrich, R., Weston, J., 2000. Adaptive Margin Support Vector Machines for Classification. Advances in Large Margin Classifiers. MIT Press, Cambridge, Massachusetts, USA, p.281-295.

King, S.P., King, D.M., Astley, K., Tarassenko, L., Hayton, P., Utete, S., 2002. The Use of Novelty Detection Techniques for Monitoring High-Integrity Plant. Proc. Int. Conf. on Control Applications, **1**:221-226. [doi:10.1109/CCA.2002.1040189]

Krause, N., Singer, Y., 2004. Leveraging the Margin More Carefully. Proc. 21st Int. Conf. on Machine Learning, p.1-8. [doi:10.1145/1015330.1015344]

Laskov, P., Schafer, F., Kotenko, I., 2004. Intrusion Detection in Unlabeled Data with Quarter-Sphere Support Vector Machines. Proc. DIMVA, p.71-82.

Manevitz, L.M., Yousef, M., 2002. One-class SVMs for document classification. *J. Mach. Learn. Res.*, **2**(2):139-154.

Ratsch, G., Mika, S., Scholkopf, B., Muller, K.R., 2002. Constructing boosting algorithms from SVMs: an application to one-class classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**(9):1184-1199. [doi:10.1109/TPAMI.2002.1033211]

Scholkopf, B., Smola, A.J., 2002. Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge, Massachusetts, USA, p.135-141.

Song, Q., Hu, W., Xie, W., 2002. Robust support vector machine with bullet hole image classification. *IEEE Trans. Syst. Man Cybern. C*, **32**(4):440-448.

Steinwart, I., Hush, D., Scovel, C., 2005. A classification framework for anomaly detection. *J. Mach. Learn. Res.*, **6**:211-232.

Tax, D., Ypma, A., Ypma, E., Duin, R.P.W., 1999. Support Vector Data Description Applied to Machine Vibration Analysis. Annual Conf. of the Advanced School for Computing and Imaging, p.398-405.

Tax, D.M.J., 2001. One-Class Classification: Concept-Learning in the Absence of Counter-Examples. PhD Thesis, Delft University of Technology, Delft, the Netherlands.

Thongkam, J., Xu, G., Zhang, Y., Huang, F., 2008. Support Vector Machine for Outlier Detection in Breast Cancer Survivability Prediction. APWeb Workshop, p.99-109. [doi:10.1007/978-3-540-89376-9-10]

Wu, Y., Liu, Y., 2007. Robust truncated hinge loss support vector machines. *J. Am. Statist. Assoc.*, **102**(479):974-983. [doi:10.1198/016214507000000617]

Xu, L., Crammer, K., Schuurmans, D., 2006. Robust Support Vector Machine Training via Convex Outlier Ablation. Proc. National Conf. of Artificial Intelligence, **21**:536-542.

Zhang, T., 2008. Multi-stage Convex Relaxation for Learning with Sparse Regularization. NIPS, p.1929-1936.