A new perspective on convex relaxations of sparse SVM

Noam Goldberg * Sven Leyffer† Todd Munson‡

Abstract
This paper proposes a convex relaxation of a sparse support vector machine (SVM) based on the perspective relaxation of mixed-integer nonlinear programs. We seek to minimize the zero-norm of the hyperplane normal vector with a standard SVM hinge-loss penalty and extend our approach to a zero-one loss penalty. The relaxation that we propose is a second-order cone formulation that can be efficiently solved by standard conic optimization solvers. We compare the optimization properties and classification performance of the second-order cone formulation with previous sparse SVM formulations suggested in the literature.

1 Introduction
Given a dataset $(A, y) \in \mathbb{R}^{m \times n} \times \{-1, 1\}^m$ we consider binary classification by support vector machines (SVMs), computing a hyperplane \( \{ x \in \mathbb{R}^n \mid w^T x + b = 0 \} \) in order to classify any \( x \in \mathbb{R}^n \) based on \( \text{sgn}(w^T x + b) \). Letting \( [k] = \{1, \ldots, k\} \), we seek a hyperplane to separate a subset of \( \{ i \in [m] \mid y_i = 1 \} \) from a subset of \( \{ i \in [m] \mid y_i = -1 \} \): a separator that we would like to generalize well for the test data. In the standard SVM formulation this is achieved by minimizing \( ||w||_2^2 \) or, equivalently maximizing the margin of separation with respect to the training data. Since generally the data is inseparable, one usually minimizes the sum of \( ||w||_2^2 \) and the hinge-loss penalty \( c \sum_{i=1}^m [1 - y_i (A_i w + b)]_+ \) for some \( c \geq 0 \), where \( A_i \) is the \( i \)-th row of \( A \), and \( [\cdot]_+ = \max(\cdot, 0) \). The hinge-loss, however, is only a surrogate for the quantity of interest: the number of misclassifications that is measured by the zero-one loss; see Höfgen et al. [19] and the discussion and references in Bennett and Bredensteiner [3]. We denote the zero-norm of a vector \( a \in \mathbb{R}^n \) by \( ||a||_0 = \{|j \in [n] \mid a_j \neq 0\} \). Next we consider sparse SVM formulations that minimize the sum of \( ||w||_0 \) and the hinge-loss penalty. We also extend the model to replace the hinge-loss with a zero-one penalty.

These formulations are motivated by early generalization bounds of boosting [12] and the compression interpretation of learning [5]. Our formulations are also motivated by the perspective reformulation of mixed-integer nonlinear programs (MINLPs), a MINLP formulation and relaxation technique that may help to eliminate (in some cases) the need for big-M constants that typically result in weak continuous relaxations [18]. The weakness of big-M relaxations in the context of classifier ensemble mixed-integer programming formulations was demonstrated by Goldberg and Eckstein [15].

2 Convex Relaxations of Sparse SVM
Chan et al. [9] consider a sparse SVM formulation that applies a constraint \( ||w||_0 \leq r \) to the standard SVM formulation:

\[
\begin{align*}
(2.1) \quad & \min_{\xi, w, b} \{ ||w||_2^2 + c ||\xi||_1 \mid Y(Aw + b) + \xi \geq 1 \}, \\
& \text{where } Y \text{ denotes the } m \times m \text{ diagonal matrix with diagonal } y. \text{ They propose a quadratically constrained quadratic program (QCQP) and a semidefinite programming (SDP) as convex relaxations of a sparse SVM (SSVM) formulation. Both relaxations are inspired by and rely on the vector norm inequality } ||a||_1 \leq \sqrt{||a||_0} ||a||_2. \text{ The QCQP-SSVM is}\n\end{align*}
\]

\[
\begin{align*}
(2.2a) \quad & \min_{\xi, w, b, t} \quad t + c \sum_{i=1}^m \xi_i \\
(2.2b) \quad & \text{such that } \quad Y(Aw + b) + \xi \geq 1 \\
(2.2c) \quad & ||w||_2^2 \leq t \\
(2.2d) \quad & ||w||_1^2 \leq rt \\
(2.2e) \quad & \xi \geq 0.
\end{align*}
\]

This formulation is equivalent to replacing the standard SVM objective by

\[
\min_{\xi, w, b} \left\{ \max \left\{ \frac{1}{r} ||w||_1^2, ||w||_2^2 \right\} + c \sum_{i=1}^m \xi_i \right\}.
\]

For the sake of brevity we omit the formulation of the SDP relaxation (SDP-SSVM) which can be found in Chan et al. [9].
Guan et al. [17] proposed a MINLP for optimally solving a closely related problem where $||w||_0$ is penalized in the objective instead of being subject to a hard constraint. Because of the intractability of the problem, however, they were limited to solving the problem only for small datasets. Although solving the discrete problem to optimality may better optimize the tradeoff of hinge-loss and $||w||_0$, tractable convex relaxations may suffice to improve on classification performance and/or sparsity; see for example [9, 14]. Recently, Tan et al. [22] proposed an algorithm for sparse SVM to solve a continuous relaxation with an infinite number of constraints. Their method for solving that formulation is specialized for problems with a large number of features. Here we consider finite conic formulations that are closer to the approach [9]; these can be solved by standard conic optimization solvers. Further, with the advent of first-order methods for conic optimization our models may apply with a large number of observations and potentially in online settings.

Alternatively, smooth approximations of $||w||_0$ [7, 24] and LP-SVM [8, 13] have been used for sparse classification. However, smooth approximation techniques give rise to nonconvex optimization problems requiring one to settle for local minima. On the other hand, LP-SVM and linear programming formulations in general are poor relaxations of the zero-norm minimization problem because of the required big-M constants. The sparsity however can still be controlled by setting smaller hinge-loss penalties (i.e., smaller values of $c$); see [15] for a detailed analysis of the setting where $A \in \{-1,0,1\}^{m \times n}$. In the following we consider a convex (conic) relaxation that avoids the use of big-M constants that may otherwise be needed to formulate $||w||_0$ within a mathematical program.

3 Combining the 2-norm and Zero-Norm Penalties

We now consider a sparse SVM formulation that minimizes a linear combination of the two-norm and zero-norm of $w$:

$$\min_{\xi,w} \left\{ ||w||_2^2 + c \xi ||1|| + d ||w||_0 \mid Y(Aw + 1b) + \xi \geq 1 \right\}.$$ 

This problem can be motivated by test error (or generalization) bounds that appear in the literature, some given in terms of the margin of separation (the inverse of the 2-norm of $w$), some given in terms of sparsity, and some combining both; see, for example, [20].

Guan et al. [17] considered this problem\footnote{In their formulation, Guan et al. [17] introduce an additional constraint for enforcing $||w||_0 \geq 1$. However, one may set $c$ and $d$ to $0$ to enforce $||w||_0 \geq 1$ endogenously.} and formulated it as a MINLP, using binary indicator variables $z \in \{0,1\}^n$ and auxiliary variables $u \in \mathbb{R}_+^n$.

\begin{align}
(3.3a) \min_{\xi,w} & \sum_{j=1}^n u_j + c \sum_{i=1}^m \xi_i + d \sum_{j=1}^n z_j \\
(3.3b) \text{such that} & \quad Y(Aw + 1b) + \xi \geq 1 \\
(3.3c) & \quad w_j^2 - z_j u_j \leq 0 \quad j \in [n] \\
(3.3d) & \quad \xi \geq 0, z \in \{0,1\}^n.
\end{align}

The constraints (3.3c) ensure that

$$u_j \begin{cases} \leq u_j & z_j = 1 \\ = 0 & z_j = 0. \end{cases}$$

We note that for sufficiently large values of $c$ and $d$, the $u_j$ terms of (3.3a) become negligible and an optimal solution of (3.3) is also optimal in

$$\min_{u,\xi} \left\{ ||w||_0 + c \xi ||1|| \mid Y(Aw + 1b) + \xi \geq 1 \right\}$$

for $\tilde{c} = c/d$. Hence, (3.3) generalizes previously considered zero-norm minimization problems (e.g., Weston et al. [24] and Amaldi and Kann [1]), known to be $\mathcal{NP}$-hard, implying that (3.3a) is $\mathcal{NP}$-hard.

In addition to the fact that this problem is computationally challenging MINLP, another obstacle is that the left-hand side of (3.3c) is not convex. Further, even general nonlinear programming solvers have difficulty in computing local minima for continuous relaxations of (3.3) because the constraints (3.3c) violate constraint qualification. Consequently, to solve small instances of (3.3), Guan et al. [17] resorted to replacing the constraints (3.3c) by big-M constraints of the form $|w_j| \leq Mz_j$.

4 A Second-Order Cone Relaxation

As $u, z \geq 0$, constraint (3.3c) can be rewritten as a convex second-order cone constraint, for each $j \in [n]$,

$$||(2w_j, u_j - z_j)|| \leq u_j + z_j.$$ 

Let $\mathbb{Q}^n$ denote the $n$-dimensional second-order (Lorentz) cone; see Ben-Tal and Nemirovski [2] for a definition and related results concerning second-order cone programming. Now, relaxing the variables and letting $z_j \in [0,1]$ in place of $z_j \in \{0,1\}$, a second-order cone relaxation of (3.3) in which we also replace the $d ||w||_0$ penalty by a hard constraint for a given
parameter $r \geq 0$ is
\[
\begin{align*}
\text{(5.4a)} \quad &\text{min}_{\xi, w, b, u, z} \sum_{j=1}^{n} u_j + c \sum_{i=1}^{m} \xi_i \\
\text{(5.4b) such that} \quad & Y(Aw + I b) + \xi \geq 1 \\
\text{(5.4c) } \quad & (2w_j, u_j - z_j, u_j + z_j) \in \mathbb{Q}^3 \quad j \in [n] \\
\text{(5.4d) } \quad & \sum_{j=1}^{n} z_j \leq r \\
\text{(5.4e) } \quad & \xi \geq 0, z \in [0, 1]^n.
\end{align*}
\]

This is a convex optimization problem that can be solved efficiently and also rapidly in practice by specialized interior-point conic optimization solvers. We refer to this novel formulation as CQ-SSVM. Note that for a sufficiently large values of $r$, in particular for $r = n$, the problem reduces to the standard SVM problem (2.1).

5 Extending the Formulation for the Zero-One Loss

We now consider a formulation that minimizes the zero-one loss in place of the standard SVM hinge-loss:
\[
\begin{align*}
\text{(5.5) } \quad &\text{min}_{\xi, w, b} \{ ||w||_0 + c \||\xi||_0 | Y(Aw + I b) + \xi \geq 1 \}.
\end{align*}
\]

Similar formulations that attempt to minimize the zero-one loss in conjunction with penalizing $||w||_0$ for all $r = n$, and constraints to (3.3) for each of the variables $\xi_i$, for $i \in [n]$, we may formulate this problem as a MINLP:
\[
\begin{align*}
\text{(5.6a) } \quad &\text{min}_{\xi, w, b, u, z, q, x} \sum_{j=1}^{n} (u_j + d z_j) + \sum_{i=1}^{m} (s_i + c q_i) \\
\text{(5.6b) such that} \quad & Y(Aw + I b) + \xi \geq 1 \\
\text{(5.6c) } \quad & w^2_j - z_j u_j \leq 0 \quad j \in [n] \\
\text{(5.6d) } \quad & \xi^2_i - q_i s_i \leq 0 \quad i \in [m] \\
\text{(5.6e) } \quad & \xi \geq 0, z \in [0, 1]^n, \quad q \in [0, 1]^m.
\end{align*}
\]

The following proposition establishes a lower bound for $c$ and $d$ as a sufficient condition for an optimal solution of (5.7) to be optimal for (5.6). It is assumed that the data is integer; however, note that every rational matrix can be scaled so that its entries are integer.

**Proposition 5.1.** Suppose $A \in \mathbb{Z}^{m \times n}$, $A_{ij} \leq U$ for all $i, j$, and $c, d \geq U^{2m}m^{m+2}$. Then, for every $(\xi^*, w^*, b^*, a^*, s^*, q^*, z^*)$ that is optimal to (5.7), $(\xi^*, w^*, b^*)$ must be optimal to (5.6).

**Proof.** Clearly, $(\xi^*, w^*, b^*)$ is feasible for (5.6). Also note that by optimality to (5.7) we have $u_j^* = w_j^2$ for $j \in [n]$, and $s_i^* = \xi_i^2$ for $i \in [m]$. Now, assume for the sake of deriving a contradiction some $(\hat{w}, \hat{\xi})$ that is feasible for (5.6) (and hence also to (5.7b)) with $||\hat{w}||_0 + c ||\hat{\xi}||_0 < ||w^*||_0 + c ||\xi^*||_0$. Now, the system of inequalities (5.7b), for $i \in [m]$, with $\xi_i = 0$ fixed for $i$ with $\hat{\xi}_i = 0$, and $w_i = 0$ fixed for $j$ with $\hat{w}_j = 0$, has a basic feasible solution. Let $B \in \mathbb{R}^{m \times r}$ denote a submatrix of $(Y(Aw + I b))$ that forms a basis. This basis corresponds to a set of active inequalities $B(\hat{w} \ b \ \hat{\xi}) = 1$, for some $B \in \mathbb{R}^{k \times l}$, which is a submatrix of $B$, $\hat{w} \in \mathbb{R}^k$, and $\hat{\xi} \in \mathbb{R}^l$ with $k \in \{0\} \cup [n]$ and $l \in \{0\} \cup [m]$. Applying standard techniques using the matrix inequalities that are optimal to (5.7), we have $\hat{w}_j, \hat{\xi}_j \leq U^{m}m^{m/2}$. It follows that $||\hat{w}, \hat{\xi}||_1 \leq U^m m^{m/2 + 1}$. Hence, by a standard norm inequality and, respectively, the supposition of the proposition $||(\hat{w})^T, (\hat{\xi})^T||_2^2 \leq U^{2m}m^{m+2} \leq c, d$. By construction $||\hat{w}||_0 + c ||\hat{\xi}||_0 < ||w^*||_0 + c ||\xi^*||_0$, so
\[
\begin{align*}
||(\hat{w})^T, (\hat{\xi})^T||_2^2 + d ||\hat{w}||_0 < U^{2m}m^{m+3} + c ||\hat{\xi}||_0 \\
+ d ||\hat{w}||_0 \leq \sum_{j=1}^{n} (u_j + d z_j) + \sum_{i=1}^{m} (r_i + c q_i)
\end{align*}
\]

thereby establishing a contradiction. ■
Figure 1: Illustration of the perspective relaxation: the optimum (with respect to feature $j$) $(\hat{w}_j, \hat{u}_j, \hat{z}_j)$ is taken over the convex hull of $(0, 0, 0)$ and $(t, t^2, 1)$ for $t \geq 0$. Such an optimal solution satisfies $\hat{z}_j = \frac{\hat{w}_j^2}{\hat{u}_j}$. In a big-$M$ formulation one would have $\hat{z}_j = \frac{\hat{w}_j^2}{M}$ for a potentially very large constant $M$.

6 Evaluating the Quality of the Relaxation

Ideally one can solve the MINLP (3.3) in order to compare its optimal solution with the solution of the relaxation. However, the MINLP becomes increasingly difficult to solve with more than a small number of features.

The perspective (see [6, 18]) of $g(w_j, u_j) = w_j^2 - u_j$ is illustrated in Figure 1. Now, consider a solution $(\xi, w, b, u, z)$ that is optimal to (4.5). If $u_j = w_j^2$, then the constraint (4.5c) (which is equivalent to (3.3c)) implies that $z_j = 1$ if and only if $w_j > 0$. Otherwise, if $u_j > w_j^2$ and (4.5d) is binding, it must be that $0 < z_j = w_j^2/u_j < 1$. Further, it is precisely when $c$ is large compared with unity (the objective coefficients of the $u_j$’s), and when $r$ is small, that for each $j \in [n]$, that $u_j$ may tend to overestimate $w_j^2$ in order to allow $z_j < 1$. Intuitively, the larger the values of $c$, and the smaller $r$ is, the larger the Lagrangian multipliers that are associated with the constraints (4.5c) and (4.5d) and that “push against” $z_j$ being large, for each $j \in [n]$.

Note that formulations (4.5) and (5.8) facilitate comparison with (2.2) and also with an integer solution. Consider $(\xi, w, b, u, z)$ that is optimal to (4.5) with $\|w\|_0 = r$. Then, the fact that $\|w\|_0 = r$ implies the feasibility of $z_j = 1$ for all $j$ with $w_j > 0$. By optimality and the fact that $z_j \in [0, 1]$ it follows that $u_j = u_j z_j = w_j^2$ for all $j \in [n]$. Therefore, $\|w\|_0 = r$, implies that $z \in \{0, 1\}^n$. Empirical evidence for quality of the relaxation is given in Section 7.1.

7 Computational experiments

Chan et al. [9] suggested that investigating the trade-off of $\|w\|_0$ and accuracy would be of interest although not in the scope of their work. Here we more closely examine this tradeoff for both SDP-SSVM and QCQP-SSVM as well as for our formulations CQ-SSVM and CQ01-SSVM. We compare the quality of the different relaxations for different values of the penalty parameters. We also compare the classification performance and generalization of our two novel formulations CQ-SSVM and CQ01-SSVM with previous relaxations.

We solve the optimization models using the SDPT3 solver [23] version 4.0. SDPT3 is a specialized interior-point solver for conic optimization problems. We note that all the formulations considered in this paper can be cast as conic optimization problems. To implement and solve all formulations we also used CVX, a package for specifying and solving convex programs [16].

7.1 Optimization and Relaxation Quality

We ran experiments on the entire datasets in order to examine the quality of the relaxation of the MINLP by applying each type of relaxation. In Figures 2(a) and 2(c) we show the actual values of $\|w\|_0$ as $r$ is varied in formulations (4.5), (5.8), (2.2), and SDP-SSVM using the UCI-Ionosphere dataset [11]. Following the discussion of Section 6, points that lie on the diagonal line correspond to integer solutions. Figures 2(b) and 2(d) show the training accuracy vs. the density of $w$ as $r$ is varied, for $c = 2^6$ and $c = 2^{-6}$, respectively. In the case that $c$ is “optimally selected” for the dataset and QCQP-SSVM method, as with $c = 2^6$, then all four methods perform nearly the same. However, for most other choices of the parameter $c$, such as the case of $c = 2^{-6}$ in Figure 2(d), then there appears to be a clear advantage of CQ-SSVM and CQ01-SSVM over SDP-SSVM and QCQP-SSVM in terms of integrality of the indicator variable vector $z$, and more importantly in terms of the training accuracy--$\|w\|_0$ trade-off.

The average SDP-SSVM running time in the experiments of Figures 2(a)–2(d) was 40.99 seconds compared with 1.87 seconds on average to solve the CQ01-SSVM formulation, a formulation with $m+n$ second-order cone constraints. Due to the computational cost of SDP-SSVM and the fact that Figures 2(a)–2(d) demonstrated that the SDP-SSVM solutions were similar to those of QCQP-SSVM, we did not further consider SDP-SSVM.

Table 1 indicates the size of each dataset and shows the running times of CQ-SSVM and CQ01-SSVM compared with standard SVM, LP-SVM and QC-SSVM. The running times, except for the Madelon dataset, apply to runs using roughly 80% of the dataset, which is used as the training set. For the Madelon roughly 67% of the dataset is used for the training optimization problem. The running times indicated include both the processing time of CVX as well as the
Figure 2: Integrality and sparsity experiments using the Ionosphere dataset. Note that the diagonal curves in the figures on the left correspond to integer solutions. Relaxations typically result in a larger $||w||_0$ than $r$.

Running time of the SDPT3 solver. The results of the table indicate that for half of the datasets considered CQ-SSVM is faster to compute than QCQP-SSVM.

**7.2 Classification Performance Evaluation**

For the classification experiments we did not further consider the SDP-SSVM method, as mentioned, because of its computational cost and the fact that the resulting classifiers were similar to those of QCQP-SSVM. In our experiments we set the parameter $c$ based on an internal 3-fold stratified cross validation (CV) for all datasets. The parameter setting was chosen to provide the best accuracy with $||w||_0$ as a tiebreaker. We experimented with the range of parameter values $c \in \{2^{-10}, 2^{-8}, \ldots, 2^8, 2^{10}\}$. The parameter $r$ is set to one for CQ-SSVM and CQ01-SSVM; this setting seemed to provide a reasonable tradeoff of accuracy and sparsity. More careful fine-tuning of the parameter can be applied to improve the classification results even further. For QCQP-SSVM we applied the setting of $r = 0.1$ as recommended by Chan et al. [9].

Table 2 displays the classification performance of the three methods compared also with the LP-SVM [13] formulation. The results summarized in the table indicate that CQ01-SSVM provides the sparsest solutions on average. This may be due to the setting of $r = 1$ having a greater impact in formulation (5.8) than in (4.5). In pairwise comparisons CQ-SSVM outperforms the...
Table 1: CPU time statistics of the solver on cross-validation instances solved. The mean CPU-seconds is given plus and minus a standard deviation. The UCI dataset sizes are given after preprocessing; categorical attributes are converted into a several features, one for each attribute value, and observations with missing numerical attribute values are removed. Below $\Phi$ denotes the positive label proportion of the data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$m$</th>
<th>$\Phi$</th>
<th>$n$</th>
<th>SVM</th>
<th>LP-SVM</th>
<th>QCQP-SSVM</th>
<th>CQ-SSVM</th>
<th>CQ01-SSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voting</td>
<td>435</td>
<td>0.38</td>
<td>48</td>
<td>4.6 ± 0.9</td>
<td>4.0 ± 1.6</td>
<td>19.4 ± 5.4</td>
<td>20.8 ± 4.8</td>
<td>45.0 ± 6.1</td>
</tr>
<tr>
<td>Wisc.</td>
<td>683</td>
<td>0.35</td>
<td>10</td>
<td>18.8 ± 2.6</td>
<td>0.9 ± 0.2</td>
<td>14.1 ± 2.2</td>
<td>30.4 ± 3.7</td>
<td>19.4 ± 1.9</td>
</tr>
<tr>
<td>WDBC</td>
<td>569</td>
<td>0.37</td>
<td>31</td>
<td>17.9 ± 4.3</td>
<td>1.0 ± 0.2</td>
<td>11.5 ± 1.8</td>
<td>28.5 ± 5.5</td>
<td>32.5 ± 4.0</td>
</tr>
<tr>
<td>WPBC</td>
<td>396</td>
<td>0.78</td>
<td>32</td>
<td>10.2 ± 2.3</td>
<td>2.4 ± 0.5</td>
<td>18.8 ± 2.9</td>
<td>13.6 ± 1.7</td>
<td>26.4 ± 3.2</td>
</tr>
<tr>
<td>SPECT</td>
<td>80</td>
<td>0.50</td>
<td>22</td>
<td>0.4 ± 0.1</td>
<td>0.3 ± 0.1</td>
<td>1.6 ± 0.4</td>
<td>1.5 ± 0.3</td>
<td>10.1 ± 1.8</td>
</tr>
<tr>
<td>SPECTF</td>
<td>80</td>
<td>0.50</td>
<td>44</td>
<td>0.6 ± 0.1</td>
<td>2.4 ± 0.6</td>
<td>10.3 ± 2.6</td>
<td>7.5 ± 1.6</td>
<td>10.5 ± 1.8</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>0.64</td>
<td>34</td>
<td>6.3 ± 1.2</td>
<td>4.0 ± 1.1</td>
<td>19.7 ± 4.1</td>
<td>18.4 ± 2.7</td>
<td>26.9 ± 6.0</td>
</tr>
<tr>
<td>PIMA</td>
<td>708</td>
<td>0.65</td>
<td>8</td>
<td>17.9 ± 2.6</td>
<td>0.8 ± 0.4</td>
<td>9.4 ± 1.9</td>
<td>27.6 ± 5.4</td>
<td>18.3 ± 4.0</td>
</tr>
<tr>
<td>Spam</td>
<td>4601</td>
<td>0.61</td>
<td>57</td>
<td>74.6 ± 19.9</td>
<td>3.0 ± 0.7</td>
<td>165.5 ± 23.8</td>
<td>96.7 ± 26.2</td>
<td>435.0 ± 84.2</td>
</tr>
<tr>
<td>Arrhythmia</td>
<td>68</td>
<td>0.29</td>
<td>279</td>
<td>0.2 ± 0.0</td>
<td>0.5 ± 0.1</td>
<td>5.8 ± 0.7</td>
<td>5.8 ± 0.5</td>
<td>7.1 ± 0.7</td>
</tr>
<tr>
<td>Madelon*</td>
<td>2000</td>
<td>0.50</td>
<td>500</td>
<td>335.5 ± 94.6</td>
<td>68.4 ± 4.9</td>
<td>346.4 ± 56.6</td>
<td>485.3 ± 122.1</td>
<td>768.1 ± 88.1</td>
</tr>
</tbody>
</table>

*For the Madelon dataset 5 repetitions (of a 3-fold CV experiment) were performed.

competing methods in most cases with respect to accuracy. CQ01-SSVM outperforms all other methods in terms of the average accuracy over all datasets. QCQP-SSVM provides classifiers that are nearly as sparse as CQ01-SSVM with the exception of its poor performance (in terms of both accuracy and sparsity) on the Madelon dataset. Nevertheless, CQ01-SSVM seems to provide the best overall balance of accuracy and sparsity. Although, it should be noted that QCQP-SSVM and CQ01-SSVM are more computationally intensive compared to the other methods. The more computationally tractable CQ-SSVM alternative demonstrates classification performance that is second only to CQ01-SSVM. It also provides a consistent balance of sparsity and accuracy over the datasets and an improvement in terms of average classification accuracy compared with the competing QCQP-SSVM and LP-SVM methods.

7.3 Experiments with Label Noise Here we trained the classifiers on a training set $(A, \hat{y}) \in \mathbb{R}^{m \times n} \times \{-1, 1\}^m$ with a fraction $\rho \in (0, 1)$ of the original observation labels $y$ flipped. Specifically, a subset $S \subseteq [m]$ with $|S| = \lfloor \rho m \rfloor$ is selected uniformly at random from $[m]$ so that $\hat{y}$ satisfies

$$\hat{y}_i = \begin{cases} 
(−1)y_i & i \in S \\
y_i & i \in [m] \setminus S.
\end{cases}$$

The accuracy is then evaluated on the test set based on the true label vector $y$. We experimented with $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. In these experiments $r = 1$ for CQ-SSVM and CQ01-SSVM for the Ionosphere dataset and $r = 2$ for the Spam dataset. The parameter $c$ is selected based on internal cross-validation using the noisy labels $\hat{y}$.

Figure 3 shows the results of the experiments using the Ionosphere and Spam datasets [11]. Each bar corresponds to the average of five replications of a $k$-fold experiment with $k = 5$. The plot shows that both SVM and L1-SVM are less robust to increased label noise. For the Ionosphere dataset the sparse SVM methods and in particular CQ01-SSVM demonstrates an increased advantage over the other methods as the label noise increases. For the Spam dataset the advantage of sparse SVM is less apparent and especially with low levels of noise. However, with increased levels of noise, e.g., when $\rho = .4$, the CQ01-SSVM method shows a clear advantage over other methods.

8 Conclusions and Future Work

We propose novel second-order cone relaxations of sparse SVM. In empirical tests our relaxation is tighter than the norm-inequality based convex relaxations of Chan et al. [9]. Further, it is roughly as fast to compute as the QCQP convex relaxation and significantly faster than SDP-SSVM, also proposed by Chan et al [9]. Overall, our proposed convex relaxation CQ-SSVM yields improvements in classification performance and its advantage is especially apparent for datasets with more features than observations. The formulations we propose appear to be more robust to the choice of the penalty parameters, obtaining a reasonable tradeoff of accuracy and sparsity without extensive fine-tuning of
the penalty parameters. The improvement in the overall tradeoff of sparsity and classification performance is especially apparent for CQ01-SSVM, a formulation that applies a similar perspective relaxation technique to the margin deviation variables in order to approximate the 0-1 loss. Although more computationally expensive than alternative quadratic programming formulations considered herein, CQ01-SSVM remains less computationally expensive than the SDP-SSVM alternative. The extension for the zero-one loss also shows promise for cases in which labels of observations are subject to noise. It may also be interesting to experiment with adversarial label flipping; see, for example, [4] and references therein for different approaches.

Note that as an optimal solution of (3.3) may be sparse, including many zero margin-deviation variables, we do not necessarily need to add all $m$ observations, and potentially non-binding constraints (5.8d), ahead of time. In this paper we experimented with an interior point solver (SDPT3) and hence dynamically generating the constraints and resolving may not have been sensible. Resolving the problem and dynamically generating constraints may be sensible when solving these formulations using first-order methods. Recent and ongoing development of first-order methods for conic optimization and in particular for second-order cone programming may allow the application of our methods to large-scale datasets. The ability to restart from any initial point could also make it suitable for an online setting (e.g., see [21, 10] for specialized large-scale methods for standard SVMs).

References


Figure 3: Experiments with label noise added to UCI datasets. $\rho$ is a proportion of the training data labels that are flipped. Each bar corresponds to the average of 25 runs (five replications of a 5-fold experiment).

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory (“Argonne”) under Contract DE-AC02-06CH11357 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.


