Direct Convex Relaxations of Sparse SVM

Antoni B. Chan ABCHAN@UCSD.EDU
Nuno Vasconcelos NUNO@ECE.UCSD.EDU
Gert R. G. Lanckriet GERT@ECE.UCSD.EDU
Department of Electrical and Computer Engineering, University of California, San Diego, CA, 92037, USA

Abstract

Although support vector machines (SVMs) for binary classification give rise to a decision rule that only relies on a subset of the training data points (support vectors), it will in general be based on all available features in the input space. We propose two direct, novel convex relaxations of a non-convex sparse SVM formulation that explicitly constrains the cardinality of the vector of feature weights. One relaxation results in a quadratically-constrained quadratic program (QCQP), while the second is based on a semidefinite programming (SDP) relaxation. The QCQP formulation can be interpreted as applying an adaptive soft-threshold on the SVM hyperplane, while the SDP formulation learns a weighted inner-product (i.e. a kernel) that results in a sparse hyperplane. Experimental results show an increase in sparsity while conserving the generalization performance compared to a standard as well as a linear programming SVM.

1. Introduction

Support vector machines (SVMs) (Vapnik, 1995) address binary classification problems by constructing a maximum-margin hyperplane that separates two classes of data points. Typically, the SVM hyperplane is a function of a relatively small set of training points, i.e., those on or over the margin. Although sparse with respect to data points, the SVM hyperplane is usually not sparse in the original feature space; the decision surface spans all dimensions of the latter, and all features contribute to the decision rule. In many applications this may not be desired. First, if it is known that some of the features are noise it may be sensible to simply ignore the associated dimensions, improving the generalization ability of the decision rule. Second, if some features are redundant (e.g. linear combinations of other features), it may be possible to ignore the redundant features without drastically changing the classification performance, and thus reduce the computational (or experimental) requirements of feature extraction. Third, feature selection is often desirable for reasons of interpretability, i.e. there is a need to find which features are important for a given physical process. For example in biological experiments, the features may be gene expression data on DNA microarrays, and identifying important features may lead to a better understanding of the underlying biological process. Finally, if the input space is high dimensional, and obtaining the features is expensive (e.g., the result of costly or time-consuming biological experiments), economical considerations may strongly advocate for a classifier based on a small subset of features.

An example of the improved generalization achievable with a sparse decision-rule is presented in Figure 1, which shows a two-dimensional classification problem where the second feature is noise. Twenty feature-vectors were randomly drawn from each class $y \in \{-1, 1\}$, with the first feature containing the signal $x_1 \sim \mathcal{N}(3y, 3)$, and the second feature containing only noise $x_2 \sim \mathcal{N}(0, 3)$, where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian of mean μ and variance σ^2. A standard and a sparse SVM (using an SDP relaxation to be described later) are trained, and the hyperplanes and margins of the two SVMs are shown in the figure. Note that the standard SVM hyperplane is skewed by the noise in the second feature, while the sparse SVM finds a hyperplane that only depends on the first feature. The latter has better generalization properties.

Numerous methods for feature selection have been proposed (Guyon & Elisseeff, 2003; Blum & Langley, 1997). In the area of feature selection in SVMs, previous work falls into two categories: 1) algorithms that

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

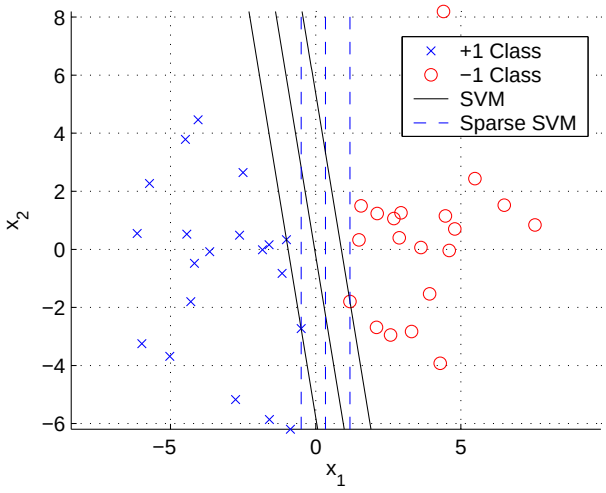

Figure 1. Example of a two-dimensional classification problem where the second feature is noise. The normal SVM fits the noise in the data, while the sparse SVM is robust and ignores the noise.

adopt a feature selection strategy disjoint from SVM training, and 2) algorithms that *simultaneously* learn the optimal subset of features and the SVM classifier. Algorithms in the first category rank features based on the parameters of the SVM hyperplane. In (Guyon et al., 2002), recursive feature elimination is combined with linear SVMs. After training an SVM, the feature with smallest weight in the decision rule is removed. A new SVM is trained on the remaining features and the process is repeated until the desired number of features remains. This method was later extended in (Rakotomamonjy, 2003) to other ranking criteria, still based on the hyperplane parameters. (Weston et al., 2003) proposes another iterative method, alternating between SVM training and re-scaling of the data, according to the feature weights.

A second category of approaches integrates feature selection and classifier training by adjusting the SVM formulation to ensure sparsity of the resulting decision rule. The proposed adjustments indirectly attempt to minimize the ℓ_0-norm (i.e., cardinality) of the hyperplane normal, a non-convex and difficult problem. The linear programming SVM (LP-SVM) (Bennett & Mangasarian, 1992; Bradley & Mangasarian, 2000; Zhu et al., 2003) achieves sparsity by minimizing the convex envelope of the ℓ_0-norm, i.e., the ℓ_1-norm of the normal vector (rather than the ℓ_2-norm as in the standard SVM). It has been applied to various problems in computational biology (Grate et al., 2002; Fung & Mangasarian, 2004) and drug-design (Bi et al., 2003). Several methods achieve sparsity by augmenting the SVM objective function with a penalty term on the cardinality of hyperplane normal. (Bradley & Mangasarian, 1998) proposes to modify the LP-SVM with a penalty term based on an ℓ_0-norm approximation. Similarly, (Neumann et al., 2005) proposes two modified SVMs that add penalty terms based on the ℓ_1-norm and on an ℓ_0-norm approximation. In contrast to adding a penalty term on the cardinality, several methods introduce adaptive scale parameters that multiply with the hyperplane normal, and feature selection is achieved by encouraging sparsity of the scale parameters. (Grandvalet & Canu, 2003) proposes to learn the scale parameters simultaneously with the standard SVM problem. In (Weston et al., 2000) the scale parameters and SVM are learned by minimizing a bound on the leave-one-out error, while (Peleg & Meir, 2004) uses the global minimization of a data-dependent generalization error-bound.

In this paper, we study a sparse SVM formulation that is obtained by augmenting the standard SVM formulation with an explicit cardinality constraint on the hyperplane normal. Note that this formulation is in contrast to previous work, which either penalizes an ℓ_0-norm approximation of the hyperplane in the objective, or uses adaptive scale-parameters. We explore two direct convex relaxations of the sparse SVM formulation. A first relaxation results in a quadratically-constrained quadratic program (QCQP) (Boyd & Vandenberghe, 2004), while the second is based on a semidefinite programming (SDP) relaxation (Lemaréchal & Oustry, 1999). Empirical results show an increase in sparsity, with roughly identical classification performance, compared to both the standard and LP-SVM. The remainder of the paper is organized as follows. In Section 2, we briefly review the standard SVM and the LP-SVM. Section 3 presents the sparse SVM and derives the QCQP and SDP convex relaxations. Finally, in Section 4 we present the results of experiments on a synthetic example and on a large set of UCI databases.

2. Standard SVM

Given a set of feature vectors $\{x_i\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$, and corresponding labels $\{y_i\}_{i=1}^N$ with $y_i \in \{-1, 1\}$, the standard (linear) SVM learns the hyperplane that separates the two classes of training points with maximal margin, measured in ℓ_2-norm. The C-SVM (Vapnik, 1995) formulation introduces slack variables to allow errors for data that may not be linearly separable. The SVM hyperplane is learned by solving a quadratic programming problem:

Problem 1 (C-SVM)

$$\begin{aligned} \min_{w,\xi,b} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0. \end{aligned}$$

The dual of the C-SVM is also a quadratic program.

Problem 2 (C-SVM dual)

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\ \alpha_j\ y_i\ y_j\ x_i^T x_j \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,N. \end{aligned}$$

The normal of the maximum-margin hyperplane can be computed as $w = \sum_{i=1}^{N}\alpha_i y_i x_i$ (Vapnik, 1995). The α_i variables are usually sparse (i.e., many of them are zero), and the hyperplane only depends on a few training points. The hyperplane is, however, generally not sparse in the original feature space; w is usually a dense vector and the decision rule depends on all features. An alternative formulation, which encourages feature selection, is the LP-SVM, introduced in (Bennett & Mangasarian, 1992). It replaces the ℓ_2-norm of the C-SVM formulation with the ℓ_1-norm.

Problem 3 (LP-SVM)

$$\begin{aligned} \min_{w,\xi,b} \quad & \|w\|_1 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0. \end{aligned}$$

The dual of the LP-SVM is also a linear program,

Problem 4 (LP-SVM dual)

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & -1 \leq \sum_{i=1}^{N}\alpha_i y_i x_i \leq 1, \qquad \sum_{i=1}^{N}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,N. \end{aligned}$$

3. Sparse SVM

The goal of the sparse SVM (SSVM) methods now proposed is to construct a maximum-margin hyperplane based on a limited subset of features in input space. This is to be achieved by computing the hyperplane parameters and the optimal feature subset simultaneously. The sparsity of the hyperplane normal w can be explicitly enforced by adding a cardinality constraint on w

Problem 5 (Sparse SVM)

$$\begin{aligned} \min_{w,\xi,b} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0, \\ & \text{Card}(w) \leq r. \end{aligned}$$

where $\text{Card}(w)$ is the cardinality of w, i.e., its ℓ_0-norm or the number of non-zero entries. This is a difficult optimization problem, since the cardinality constraint is not convex. However, a convex relaxation of the SSVM can be found by replacing the non-convex cardinality constraint by a weaker, non-convex, constraint. Indeed, for all $w \in \mathbb{R}^d$, it follows from the Cauchy-Schwartz inequality that

$$\text{Card}(w) = r \quad \Rightarrow \quad \|w\|_1 \leq \sqrt{r}\,\|w\|_2\,, \tag{1}$$

enabling the replacement of the cardinality constraint by $\|w\|_1^2 \leq r\,\|w\|_2^2$. This weaker, non-convex, constraint can now be relaxed in two ways, leading to two convex relaxations of the sparse SVM, a quadratically-constrained quadratic program (QCQP) and a semidefinite program (SDP).

3.1. QCQP Relaxation

If the ℓ_2-norm of w is bounded by another variable t, i.e., $\|w\|_2^2 \leq t$, then the constraint $\|w\|_1^2 \leq r\,\|w\|_2^2$ can be relaxed to the weaker, convex, constraint $\|w\|_1^2 \leq rt$. Relaxing the cardinality constraint of Problem 5 in this way, gives rise to a quadratically-constrained quadratic programming relaxation of the sparse SVM (QCQP-SSVM):

Problem 6 (QCQP-SSVM)

$$\begin{aligned} \min_{w,\xi,b,t} \quad & \frac{1}{2}t + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0, \\ & \|w\|_2^2 \leq t, \quad \|w\|_1^2 \leq r\ t. \end{aligned}$$

This problem is equivalent to

Problem 7 (QCQP-SSVM)

$$\begin{aligned} \min_{w,\xi,b,t} \quad & \frac{1}{2}\max\left(\frac{1}{r}\|w\|_1^2, \|w\|_2^2\right) + C\sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0. \end{aligned}$$

In other words, the QCQP-SSVM is a combination of the C-SVM and the LP-SVM where the (squared) ℓ_1-norm encourages sparsity and the ℓ_2-norm encourages a large margin (in ℓ_2-norm). The dual of QCQP-SSVM is given by (see (Chan et al., 2007) for a derivation):

Problem 8 (QCQP-SSVM dual)

$$\begin{aligned} \max_{\alpha,\nu,\eta,\mu} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2\eta}\left\|\sum_{i=1}^N \alpha_i y_i x_i + \nu\right\|_2^2 - \frac{r\mu^2}{2(1-\eta)} \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,N, \\ & -\mu \leq \nu_j \leq \mu, \quad j = 1,\ldots,d, \\ & 0 \leq \eta \leq 1, \quad 0 \leq \mu. \end{aligned}$$

The hyperplane normal w can be recovered from the dual as $w = \frac{1}{\eta}\left(\sum_{i=1}^N \alpha_i y_i x_i + \nu\right)$. Note that the dual formulation is similar to the C-SVM dual (Problem 2) in that they both contain similar linear and quadratic terms of α_i. However, the QCQP-SSVM dual introduces a d-dimensional vector ν, subjected to a box constraint of μ, which adds to the SVM hyperplane $q = \sum_{i=1}^N \alpha_i y_i x_i$. The role of the vector ν is to apply a soft-threshold to the entries of q. Consider the case where we only optimize ν, while holding all the other variables fixed. It can be shown that the optimal ν^* is

$$\nu_j^* = \begin{cases} -q_j & , \ |q_j| \leq \mu \\ -\mu & , \ q_j > \mu \\ +\mu & , \ q_j < -\mu \end{cases} \tag{2}$$

Hence, when computing the hyperplane $w = \frac{1}{\eta}(q+\nu^*)$, the feature weight w_j with corresponding $q_j \leq \mu$ will be set to zero, while all other weights will have their magnitudes reduced by μ. This is equivalent to applying a soft-threshold of μ on the hyperplane weights q_j (see Figure 2), and leads to sparse entries in w. In the general case, the magnitude of the soft-threshold μ (and hence the sparsity) is regularized by a quadratic penalty term weighted by the parameter r. In this sense, the QCQP-SSVM dual is automatically learning an adaptive soft-threshold on the original SVM hyperplane.

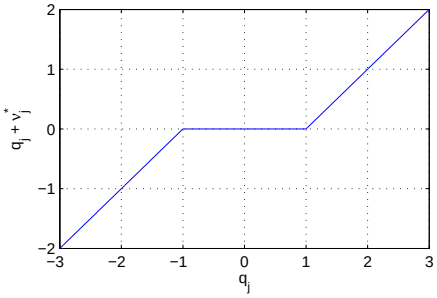

Figure 2. Example of a soft-threshold of $\mu = 1$ on hyperplane weight q_j.

There are two interesting modes of operation of the QCQP-SSVM dual with respect to the trade-off parameter η. The first is when $\eta = 0$, which corresponds to when the ℓ_1-norm constraint is active and the ℓ_2-norm constraint is inactive in the primal (Problem 6). Noting that $\nu = -\sum_{i=1}^N \alpha_i y_i x_i$ maximizes the quadratic α term, the dual problem reduces to

Problem 9

$$\begin{aligned} \max_{\alpha,\mu} \quad & \sum_{i=1}^N \alpha_i - \frac{r\mu^2}{2} \\ \text{s.t.} \quad & -\mu \leq \sum_{i=1}^N \alpha_i y_i x_i \leq \mu, \quad \sum_{i=1}^N \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,N. \end{aligned}$$

Problem 9 is similar to the dual of the LP-SVM (Problem 4), except for two additions: 1) the variable μ which sets the box constraint on $\sum_i \alpha_i y_i x_i$ (for the LP-SVM, this is set to $\mu = 1$); and 2) a quadratic penalty, weighted by r, that regularizes μ, i.e. keeps the box constraint from becoming too large. This can be interpreted as an extension of the LP-SVM dual where the box-constraint is automatically learned.

The second interesting mode is when $\eta = 1$, which corresponds to when the ℓ_2-norm constraint is active and the ℓ_1-norm constraint is inactive in the primal (Problem 6). In this case, $\mu = 0$ and $\nu = 0$, and the QCQP-SSVM dual simplifies to the standard SVM dual (Problem 2).

3.2. SDP Relaxation

A semidefinite programming relaxation (Lemaréchal & Oustry, 1999) of the sparse SVM is obtained by first rewriting the weak, non-convex, cardinality constraint (1) using the matrix $W = ww^T$

$$\|w\|_1^2 \leq r\|w\|_2^2 \quad \Leftrightarrow \quad \begin{array}{l} 1^T|W|1 \leq r\,\mathrm{tr}(W), \\ W = ww^T. \end{array} \tag{3}$$

where $|W|$ is the element-wise absolute value of the matrix W. Replacing the cardinality constraint of Problem 5 with the non-convex constraint in (3) yields

Problem 10

$$\begin{aligned} \min_{W,w,b,\xi} \quad & \frac{1}{2}\mathrm{tr}(W) + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0, \\ & 1^T|W|1 \leq r\mathrm{tr}(W), \quad W = ww^T, \end{aligned}$$

which is still non-convex because of the quadratic equality constraint, $W = ww^T$. Since we are minimizing $\mathrm{tr}(W)$, the quadratic equality constraint is equivalent to (Lemaréchal & Oustry, 1999)

$$W = ww^T \quad \Leftrightarrow \quad \begin{array}{l} W - ww^T \succeq 0, \\ \mathrm{rank}(W) = 1. \end{array} \tag{4}$$

Finally, relaxing the constraint (4) by simply dropping the rank constraint leads to a convex relaxation of the sparse SVM as a semidefinite program (SDP-SSVM):

Problem 11 (SDP-SSVM)

$$\begin{aligned} \min_{W,w,b,\xi} \quad & \frac{1}{2}\mathrm{tr}(W) + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1,\ldots,N, \\ & \xi_i \geq 0, \\ & 1^T|W|1 \leq r\mathrm{tr}(W), \quad W - ww^T \succeq 0. \end{aligned}$$

The dual of the SDP-SSVM is also an SDP (again see (Chan et al., 2007) for derivation):

Problem 12 (SDP-SSVM dual)

$$\begin{aligned} \max_{\alpha,\mu,\Lambda,\nu} \quad & \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T \Lambda^{-1} x_j \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1,\ldots,N, \\ & \Lambda = (1-\mu r)I + \nu \succeq 0, \\ & \mu \geq 0, \quad -\mu \leq \nu_{j,k} \leq \mu, \quad j,k = 1,\ldots,d. \end{aligned}$$

The hyperplane w is computed from the optimal dual variables as $w = \Lambda^{-1}\sum_{i=1}^{N}\alpha_i y_i x_i$. The SDP-SSVM dual is similar to the SVM dual (Problem 2), but in the SDP dual, the inner product between x_i and x_j is replaced by a more general, weighted inner product $\langle x_i, x_j\rangle_\Lambda = x_i^T\Lambda^{-1}x_j$. The weighting matrix Λ is regularized by μ and r, which controls the amount of rotation and scaling. Ideally, setting $\Lambda^{-1} = 0$ will maximize the quadratic term in the objective function. However, this is not possible since the entries of Λ are bounded by μ. Instead, the optimal weighting matrix is one that increases the influence of the relevant features (i.e. scales up those features), while demoting the less relevant features. Hence, the SDP-SSVM learns a weighting on the inner product (i.e., learns a kernel) such that the separating hyperplane in the feature space is sparse.

4. Experiments

We compare the performance of the two proposed sparse SVMs, QCQP-SSVM and SDP-SSVM, with the standard C-SVM and LP-SVM, on a synthetic problem and on fifteen UCI data sets.

4.1. Data Sets

The synthetic problem is a binary classification problem, similar to (Weston et al., 2000), where only the first six dimensions of the feature space are relevant for classification. With probability 0.7, the first three features $\{x_1, x_2, x_3\}$ are drawn as $x_i \sim y\mathcal{N}(i, 1)$ and the second triplet $\{x_4, x_5, x_6\}$ as $x_i \sim \mathcal{N}(0, 1)$. Otherwise, the first three features are drawn as $x_i \sim \mathcal{N}(0, 1)$, and the second triplet as $x_i \sim y\mathcal{N}(i-3, 1)$. The remaining features are noise $x_i \sim \mathcal{N}(0, 20)$ for $i = 7,\ldots,30$. One thousand data points are sampled, with equal probability for each class $y \in \{-1, 1\}$.

The remaining experiments were performed on the fifteen UCI data sets listed in Table 2. Most of these are straightforward binary or multi-class classification problems. For the Wisconsin prognostic breast cancer data set (`wpbc`) two classes were formed by selecting examples with recurrence before 24 months, and examples with non-recurrence after 24 months (`wpbc 24`). The data set was also split using 60 month recurrence (`wpbc 60`). The Cleveland `heart-disease` data set was split into two classes based on the disease level (> 2 and ≤ 2). All data sets are available from (Newman et al., 1998), and the `brown yeast` data set is available from (Weston et al., 2003).

4.2. Experimental Setup

For each data set, each dimension of the data is normalized to zero mean and unit variance. The data is split randomly with 80% of the data used for training and cross-validation, and 20% held-out for testing. Two standard SVMs, the C-SVM (Problem 1) and the LP-SVM (Problem 3), and two sparse SVMs,

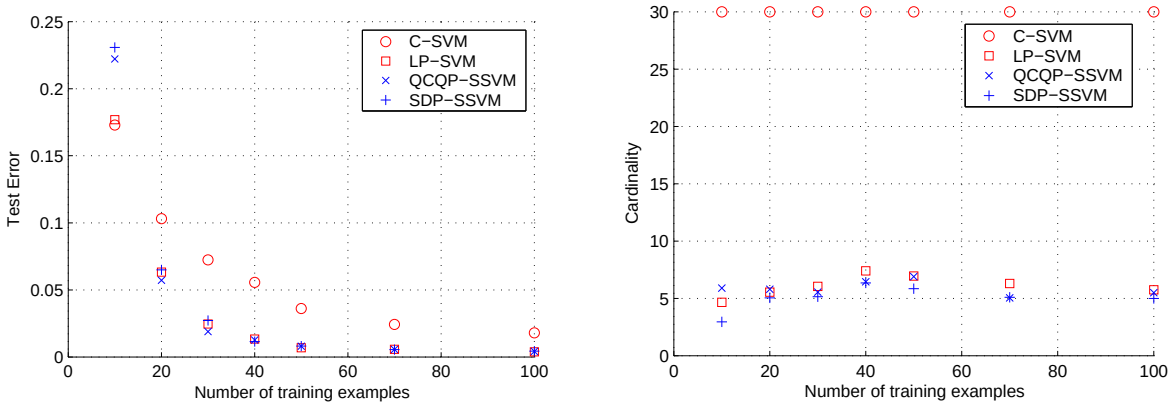

Figure 3. Results on the toy experiment: (left) test error and (right) cardinality of the SVM hyperplane versus the number of training examples.

Table 1. Test error and sparsity results averaged over the 15 UCI data sets.

	C-SVM	LP-SVM	QCQP-SSVM	SDP-SSVM
average change in error rate w.r.t. C-SVM	0.000%	0.337%	0.091%	0.449%
average sparsity (cardinality / dimension)	0.980	0.658	0.591	0.611
average sparsity w.r.t. LP-SVM	1.81	1.000	0.897	0.940

the QCQP-SSVM (Problem 6) and the SDP-SSVM (Problem 11), are learned from the training data. The sparsity parameters are set to $r_{qcqp} = 0.01$ for the QCQP-SSVM, and $r_{sdp} = 1$ for the SDP-SSVM. Although the value of r could be selected using cross-validation, in these experiments we select a low value of r to maximize the sparsity of the SSVM[1]. For each SVM, the optimal C parameter is selected using 5-fold cross-validation (80% training, 20% validation) over the range $C = \{2^{-10}, 2^{-9}, \cdots, 2^{10}, \infty\}$. The C value yielding the best average accuracy on the validation sets is used to train the SVM on all training data. Finally, the cardinality of the hyperplane w is computed as the number of weights w_j with large relative magnitude, i.e. the number of weights with $|w_j|/\max_i(|w_i|) \geq 10^{-4}$. The weights that do not fit this criteria are set to zero.

Each SVM is tested on the held-out data, and the final results reported are averaged over 20 trials of different random splits of the data. For the three multi-class datasets (`wine`, `image`, and `brown yeast`), a 1-vs-all classification problem is generated for each of the classes. The reported results are averaged over all classes. Because of their large size, the `image` and `spambase` data sets are split with 50% for training and 50% for testing. All SVMs are trained using standard convex optimization toolboxes. In particular, SeDuMi (Sturm, 1999) or CSDP (Borchers, 1999) is used to solve SDP-SSVM, and MOSEK for the other SVM.

[1]Investigation of the trade-off between sparsity and accuracy using r is also interesting, but is not presented here.

4.3. Results

Figure 3 (left) shows the test error on the synthetic data set versus the number examples used to train each of the four SVMs. The results suggest that the accuracy of all SVMs depends on the latter. When trained with at least 20 examples, the test errors of LP-SVM, QCQP-SSVM, and SDP-SSVM are similar and consistently smaller than the test error of C-SVM. On the other hand, the performance of the sparse SVMs is worse than C-SVM and LP-SVM when trained with too few examples (e.g., 10). Figure 3 (right) shows the cardinality of the classifiers. C-SVM has full cardinality (30), while LP-SVM has an average cardinality of 6.1. The QCQP-SSVM selects slightly fewer features (5.9), while SDP-SSVM selects the lowest number of features (5.1).

The overall experimental results on the 15 UCI data sets are shown in Table 1. In these experiments, the test errors obtained with the C-SVM and the sparse SVMs are roughly identical, e.g. on average the QCQP-SSVM error rate only increases by 0.091% over the C-SVM error rate. However, while C-SVM typically uses all the dimensions of the feature space, LP-SVM, QCQP-SSVM, and SDP-SSVM use much fewer. The QCQP-SSVM used the fewest features, with average sparsity (i.e. the ratio between the cardinality and the dimension of the feature space) of 0.591. In contrast, the SDP-SSVM and LP-SVM had an average sparsity of 0.611 and 0.658, respectively.

Table 2 shows the cardinality and the change in test

Table 2. Results on 15 UCI data sets: d is the dimension of the feature space, $\|w\|_0$ is the average cardinality of SVM hyperplane, "err" is the average test error, and "Δ err" is the average change in test error with respect to the C-SVM test error. The lowest cardinality and best test errors among the LP-SVM, QCQP-SSVM, and SDP-SSVM are highlighted.

		C-SVM		LP-SVM		QCQP-SSVM		SDP-SSVM	
UCI Data Set	d	$\|w\|_0$	err	$\|w\|_0$	Δ err	$\|w\|_0$	Δ err	$\|w\|_0$	Δ err
1. Pima Indians Diabetes	8	7.9	22.6%	7.2	−0.03%	7.2	**−0.16%**	**6.8**	0.07%
2. Breast Cancer (Wisc.)	9	9.0	2.9%	8.5	0.33%	8.6	**0.29%**	**8.3**	0.37%
3. Wine	12	12.0	3.5%	8.1	−0.27%	**8.0**	**−0.64%**	**8.0**	−0.45%
4. Heart Disease (Cleve.)	13	13.0	14.5%	11.4	0.42%	10.9	**0.17%**	**10.5**	0.50%
5. Image Segm.	19	17.6	2.7%	8.6	**−0.05%**	**8.0**	0.04%	8.5	−0.01%
6. SPECT	22	21.4	17.2%	14.0	**−0.09%**	12.4	0.19%	**12.1**	**−0.09%**
7. Breast Cancer (wdbc)	30	30.0	3.0%	14.4	0.70%	**12.9**	**0.57%**	13.4	0.70%
8. Breast Cancer (wpbc 24)	32	32.0	20.2%	27.7	0.00%	**16.1**	−0.16%	16.3	**−0.63%**
9. Breast Cancer (wpbc 60)	32	32.0	34.1%	16.3	3.04%	**12.1**	**1.52%**	13.9	3.26%
10. Ionosphere	34	28.2	16.2%	18.8	−1.83%	**17.2**	**−4.30%**	21.2	−4.16%
11. SPECTF	44	44.0	19.1%	39.3	0.43%	**34.0**	**0.14%**	36.1	0.50%
12. spambase	57	55.6	7.3%	53.6	**−0.12%**	**52.1**	0.04%	53.7	3.72%
13. sonar	60	60.0	22.7%	29.4	**1.05%**	**24.2**	1.51%	28.8	1.40%
14. brown yeast	79	79.0	2.5%	13.3	−0.07%	**13.0**	0.03%	13.4	**−0.14%**
15. musk	166	166.0	16.2%	83.5	**1.56%**	**74.7**	2.14%	83.6	1.72%

error for each of the UCI datasets, and Figure 4 plots the sparsities. Note that the SSVMs have the best sparsity for all of the datasets, with the improvement tending to increase with the dimensionality. In some data sets, the test error drops significantly when using SSVM (e.g. `ionosphere`, −4.30%), which indicates that some features in these data sets are noise (similar to the toy experiment). In others, the error remains the same while the sparsity increases (e.g. `brown yeast`), which suggests that these datasets contain redundant features. In both cases, the SSVM ignores the noise or redundant features and achieves a sparse solution. Finally, there are some cases where the SSVM is too aggressive and the added sparsity introduces a slight increase in error (e.g. `sonar`). This is perhaps an indication that the data set is not sparse, although the SSVM still finds a sparse classifier.

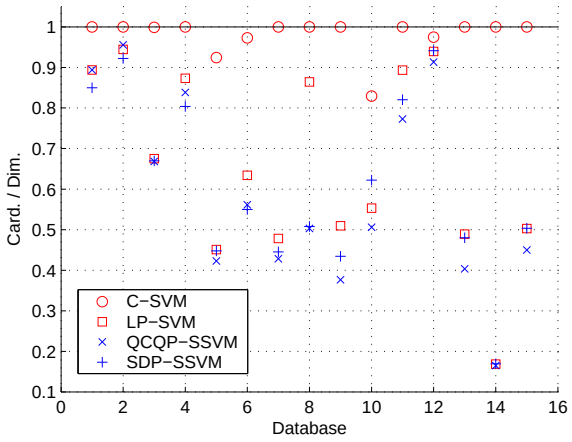

Figure 4. UCI results: sparsity of the SVM hyperplane.

5. Conclusions and Future Work

In this paper, we have formulated the sparse SVM as a standard SVM with an explicit cardinality constraint on the weight vector. Relaxing the cardinality constraint yields two convex optimization problems, a QCQP and an SDP, that approximately solve the original sparse SVM formulation. An interpretation of the QCQP formulation is that it applies an adaptive soft-threshold on the hyperplane weights to achieve sparsity. On the other hand, the SDP formulation learns an inner-product weighting (i.e. a kernel) that results in a sparse hyperplane. Experimental results on fifteen UCI data sets show that both sparse SVM achieve a test error similar to standard C-SVM, while using fewer features than both C-SVM and LP-SVM.

One interesting property of SDP-SSVM, which is missing for LP-SVM, is that its dual problem depends on an inner product $(x_i^T \Lambda^{-1} x_j)$, which suggests the possibility of kernelizing SDP-SSVM. In this case, the sparsity of the weight vector in the high-dimensional feature space (induced by the kernel) may lead to better generalization of the classifier. The main obstacle, however, is that the weighting matrix Λ^{-1} lives in the feature space. Hence, further study is needed on the properties of the matrix and whether it can be computed using the kernel.

Finally, the implementation of SDP-SSVM using the

off-the-shelf SDP optimizers (SeDuMi and CSDP) is quite slow for high-dimensional data. Future work will be directed at developing a customized solver that will make the SDP-SSVM amenable for larger and higher dimensional datasets.

Acknowledgments

The authors thank the anonymous reviewers for insightful comments, and Sameer Agarwal for helpful discussions. This work was partially supported by NSF award IIS-0448609, NSF grant DMS-MSPA 0625409, and NSF IGERT award DGE-0333451.

References

Bennett, K. P., & Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optim. Methods Softw.*, *1*, 23–34.

Bi, J., Bennett, K. P., Embrechts, M., Breneman, C. M., & Song, M. (2003). Dimensionality reduction via sparse support vector machines. *J. Mach. Learn. Res.*, *3*, 1229–1243.

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*, 245–271.

Borchers, B. (1999). CSDP, a C library for semidefinite programming. *Optim. Methods Softw.*, *11*, 613–623.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. *Intl. Conf. on Machine Learning*.

Bradley, P. S., & Mangasarian, O. L. (2000). Massive data discrimination via linear support vector machines. *Optim. Methods Softw.*, *13(1)*, 1–10.

Chan, A. B., Vasconcelos, N., & Lanckriet, G. R. G. (2007). *Duals of the QCQP and SDP sparse SVM* (Technical Report SVCL-TR-2007-02). University of California, San Diego. `http://www.svcl.ucsd.edu`.

Fung, G. M., & Mangasarian, O. L. (2004). A feature selection Newton method for support vector machine classification. *Computational Optimization and Applications*, *28*, 185–202.

Grandvalet, Y., & Canu, S. (2003). Adaptive scaling for feature selection in SVMs. *Neural Information Processing Systems*.

Grate, L. R., Bhattacharyya, C., Jordan, M. I., & Mian, I. S. (2002). Simultaneous relevant feature identification and classification in high-dimensional spaces. *Lecture Notes in Computer Science*.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, *3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Mach. Learn.*, *46*, 389–422.

Lemaréchal, C., & Oustry, F. (1999). *Semidefinite relaxations and Lagrangian duality with application to combinatorial optimization* (Technical Report 3710). INRIA.

MOSEK (2006). *MOSEK optimization software* (Technical Report). `http://www.mosek.com/`.

Neumann, J., Schnörr, C., & Steidl, G. (2005). Combined SVM-based feature selection and classification. *Mach. Learn.*, *61*, 129–150.

Newman, D. J., Hettich, S., Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases* (Technical Report). `http://www.ics.uci.edu/~mlearn`.

Peleg, D., & Meir, R. (2004). A feature selection algorithm based on the global minimization of a generalization error bound. *Neural Information Processing Systems*.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *J. Mach. Learn. Res.*, *3*, 1357–1370.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, *11*, 625–653.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag.

Weston, J., Elisseeff, A., Scholkopf, B., & Tipping, M. (2003). Use of zero-norm with linear models and kernel methods. *J. Mach. Learn. Res.*, *3*, 1439–1461.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for SVMs. *Neural Information Processing Systems*.

Zhu, J., Rossett, S., Hastie, T., & Tibshirani, R. (2003). 1-norm support vector machines. *Neural Information Processing Systems*.