BSVM

A Banded Support Vector Machine

Author:

Gautam V. Pendse

gpendse@mclean.harvard.edu

P.A.I.N Group, Brain Imaging Center
McLean Hospital, Harvard Medical School

February 14, 2022
Contents

1 Introduction 3

2 Problem setup 3
  2.1 C-SVM objective function .............................................. 4
  2.2 B-SVM objective function .............................................. 4

3 Solving the B-SVM problem 4
  3.1 The B-SVM dual problem .............................................. 6
  3.2 Kernelifying B-SVM ..................................................... 8
  3.3 Calculation of dual variables ......................................... 9
  3.4 Calculation of primal variables ....................................... 9

4 Toy data 10
  4.1 α-SVs and θ-SVs ........................................................ 14
  4.2 Bounded decision rule .................................................. 14
  4.3 Sensitivity curve ........................................................ 14

5 Discussion and conclusions 15

List of Figures

1 (a) Standard C-SVM like penalty function penalizes $y_i(\beta^T x_i + \beta_0) < \rho_1$. In B-SVM, $\rho_1$ replaces the constant 1 from C-SVM. (b) Novel B-SVM penalty function. This function penalizes $y_i(\beta^T x_i + \beta_0) > \rho_2$. (c) Total penalty function for B-SVM. If $y_i(\beta^T x_i + \beta_0) \in [\rho_1, \rho_2]$ then the total penalty is 0. Choosing $C_2 < C_1$ will impose a milder penalty for values of $y_i(\beta^T x_i + \beta_0) > \rho_2$. ...................................................... 5

2 Figure shows classification obtained for example data using (a) C-SVM and (b) B-SVM. Red and Blue points (.) correspond to class +1 and −1 respectively. Cyan and Orange x-marks (x) show the C-SVM and B-SVM decision rules evaluated at various points. Class 1 membership is indicated in Cyan and class −1 membership is indicated in Orange. The yellow squares in (a) correspond to support points for which $0 < \alpha_i < C$. The cyan squares in (b) correspond to support points for which $0 < \theta_i < C_2$ and the green squares correspond to support points for which $0 < \alpha_i < C_1$. The sparsity of solution is controlled by $\alpha$ in the case of C-SVM and $(\alpha - \theta)$ in the case of B-SVM (c) Shows $\alpha_i$ values for C-SVM. (d) Shows $(\alpha_i - \theta_i)$ values for B-SVM. ................................................................. 11
Figure shows decision rule $g(x)$ for C-SVM (a) and B-SVM (b). Note that in B-SVM the second penalty term $C_2 \sum_{i=1}^{n} [y_i (\beta^T h(x_i) + \beta_0) - \rho_2]_+$ results in most of the $g(x)$ values in the interval $[\rho_1, \rho_2] = [1, 1.5]$. (c) Heat map of the decision rule $g(x)$ for C-SVM (d) Heat map of the decision rule $g(x)$ for B-SVM. In C-SVM the values of decision rule $g(x)$ are unbalanced in Class 1. The central cluster located at $(0,0)$ in Class 1 gets much smaller $g(x)$ values in C-SVM than the rest of the Class 1. In B-SVM however, all clusters in Class 1 including the one centered at $(0,0)$ get similar $g(x)$ values. This is a result of the second penalty term in the B-SVM objective function.

Figure shows the fraction of points classified correctly by both C-SVM (blue curve) and B-SVM (red curve) as a function of the decision rule threshold. The $x$-axis shows the decision rule threshold as a percentage of the maximum absolute value of the decision function $g(x)$ over all training points. The $y$-axis shows the overall classification accuracy or sensitivity of C-SVM and B-SVM.

Abstract

We describe a novel binary classification technique called Banded SVM (B-SVM). In the standard C-SVM formulation of Cortes and Vapnik [1995], the decision rule is encouraged to lie in the interval $[1, \infty]$. The new B-SVM objective function contains a penalty term that encourages the decision rule to lie in a user specified range $[\rho_1, \rho_2]$. In addition to the standard set of support vectors (SVs) near the class boundaries, B-SVM results in a second set of SVs in the interior of each class.

Notation

 Scalars and functions will be denoted in a non-bold font (e.g., $\beta_0, C, g$). Vectors and vector functions will be denoted in a bold font using lower case letters (e.g., $x, \beta, h$). Matrices will be denoted in bold font using upper case letters (e.g., $B, H$). The transpose of a matrix $A$ will be denoted by $A^T$ and its inverse will be denoted by $A^{-1}$. $I_p$ will denote the $p \times p$ identity matrix and $0$ will denote a vector or matrix of all zeros whose size should be clear from context.

 $|x|$ will denote the absolute value of $x$ and $I(x > a)$ is an indicator function that returns 1 if $x > a$ and 0 otherwise.

 The $j$th component of vector $t$ will be denoted by $t_j$. The element $(i, j)$ of matrix $G$ will be denoted by $G(i, j)$ or $G_{ij}$. The 2-norm of a $p \times 1$ vector $x$ will be denoted by $||x||_2 = \sqrt{\sum_{i=1}^{p} x_i^2}$. Probability distribution of a random vector $x$ will be denoted by $P_x(x)$. $E[f(s, \eta)]$ denotes the expectation of $f(s, \eta)$ with respect to both random variables $s$ and $\eta$. 

2
1 Introduction

We consider the standard binary classification problem. Suppose \( y_i \) is the class membership label (+1 for class +1 and −1 for class −1) associated with a feature vector \( x_i \). Given \( n \) such \((x_i, y_i)\) pairs, we would like to learn a linear decision rule \( g(x) \) that can be used to accurately predict the class label \( y \) associated with feature vector \( x \).

In C-SVM [Vapnik and Lerner, 1963, Boser et al., 1992, Cortes and Vapnik, 1995], one can think of the linear decision rule \( g \) as a means of measuring membership in a particular class. Given a feature vector \( x \), C-SVM encourages the function \( g(x) \) to be positive if \( x \in \text{class +1} \) and negative if \( x \in \text{class } -1 \).

We motivate the development of B-SVM in the following way. Suppose that vector \( x \) comes from an arbitrary probability distribution \( P_x(x) \) with mean \( E[x] = \mu \) and finite co-variance \( \text{Cov}[x] = \Sigma \). Consider the linear decision rule \( g(x) = \beta^T x + \beta_0 \). It is easy to see that \( g(x) \) has mean \( E[g(x)] = \beta^T \mu + \beta_0 \) and covariance \( \text{Cov}[g(x)] = \beta^T \Sigma \beta \). By Chebyshev’s inequality, there exists a high probability band around \( E[g(x)] \) where \( g(x) \) is expected to lie when \( x \) comes from \( P_x(x) \).

Hence, for every probability distribution of vectors \( x \) from class +1 and class −1 with finite co-variance, \( g(x) \) is expected to lie in a certain high probability band. In B-SVM, we choose \( g(x) \) to encourage:

- \( y g(x) > 0 \) same condition as C-SVM
- \( y g(x) \in \text{certain high probability band} \) new B-SVM condition

Both of the above conditions can be satisfied if we encourage:

\[
\begin{align*}
  y g(x) & \in [\rho_1, \rho_2] & \text{with } \rho_2 > \rho_1 > 0
\end{align*}
\]

(1.1)

Since non-linear decision rules in C-SVM are simply linear decision rules operating in a high dimensional space via the kernel trick [Boser et al., 1992], the B-SVM band formation argument holds for non-linear decision rules as well.

2 Problem setup

As per standard SVM terminology, assume that we are given \( n \) data-label pairs \((x_i, y_i)\) where \( x_i \) are \( m \times 1 \) vectors and the data labels \( y_i \in \{ -1, 1 \} \). First, we consider only the linear case and afterwards transform to the general case via the kernel trick. Let \( m \times 1 \) vector \( \beta \) and scalar \( \beta_0 \) be parameters of a linear decision rule \( g(x) = \beta^T x + \beta_0 = 0 \) separating class +1 and −1 such that \( g(x) > 0 \) if \( x \) belongs to class +1 and vice versa.
2.1 C-SVM objective function

The C-SVM objective function [Cortes and Vapnik, 1995] to be minimized can be written as:

\[
f_{\text{CSVM}}(\beta, \beta_0) = \frac{1}{2} ||\beta||^2 + C \sum_{i=1}^{n} [1 - y_i(\beta^T x_i + \beta_0)]_+ \tag{2.1}
\]

where \([t]_+\) is the positive part of \(t\):

\[
[t]_+ = \begin{cases} 
0 & \text{if } t \leq 0, \\
t & \text{if } t > 0.
\end{cases}
\tag{2.2}
\]

and \(C\) governs the regularity of the solution. The C-SVM objective function penalizes signed decisions \(y_i(\beta^T x_i + \beta_0)\) whenever their value is below 1. This is the only penalty in C-SVM.

2.2 B-SVM objective function

We present below the novel B-SVM objective function that we wish to minimize:

\[
f_{\text{BSVM}}(\beta, \beta_0) = \frac{1}{2} ||\beta||^2 + C_1 \sum_{i=1}^{n} [\rho_1 - y_i(\beta^T x_i + \beta_0)]_+ + C_2 \sum_{i=1}^{n} [y_i(\beta^T x_i + \beta_0) - \rho_2]_+ \tag{2.3}
\]

where \(\rho_2 > \rho_1 > 0\) are margin parameters specified by the user and \(C_1\) and \(C_2\) are regularization constants. This objective function has two penalty terms:

\(\Rightarrow\) The first penalty term is similar to C-SVM. It penalizes signed decisions \(y_i(\beta^T x_i + \beta_0)\) whenever their values are below \(\rho_1\) (as opposed to 1 in C-SVM).

\(\Rightarrow\) The second penalty term is novel. It penalizes signed decisions \(y_i(\beta^T x_i + \beta_0)\) when their values are above \(\rho_2\).

The net effect of these penalty terms is to encourage \(y_i(\beta^T x_i + \beta_0)\) to lie in the interval \([\rho_1, \rho_2]\). Please see Figure 1 for a sketch of the two penalty terms in B-SVM.

3 Solving the B-SVM problem

We derive the B-SVM dual problem in order to maximize a lower bound on the B-SVM primal objective function in equation 2.3. This dual problem will be simpler to solve compared to the primal form 2.3. We proceed as follows:
Figure 1: (a) Standard C-SVM like penalty function penalizes \( y_i(\beta^T x_i + \beta_0) < \rho_1 \). In B-SVM, \( \rho_1 \) replaces the constant 1 from C-SVM. (b) Novel B-SVM penalty function. This function penalizes \( y_i(\beta^T x_i + \beta_0) > \rho_2 \). (c) Total penalty function for B-SVM. If \( y_i(\beta^T x_i + \beta_0) \in [\rho_1, \rho_2] \) then the total penalty is 0. Choosing \( C_2 < C_1 \) will impose a milder penalty for values of \( y_i(\beta^T x_i + \beta_0) > \rho_2 \).
As shown in 3.2, the primal problem in 2.3 can be modified into a strictly convex objective function with linear inequality constraints using slack variables. Consequently, strong duality holds and the maximum value of the B-SVM dual objective function is equal to the minimum value of the B-SVM primal objective function in 2.3.

For more details on convex duality, please see Nocedal and Wright [2006].

3.1 The B-SVM dual problem

We introduce slack variables:

\[ \xi_i = \left[ \rho_1 - y_i(\beta^T x_i + \beta_0) \right]_+ \]

\[ \eta_i = \left[ y_i(\beta^T x_i + \beta_0) - \rho_2 \right]_+ \]

into the primal objective function in 2.3. The modified optimization problem can be written as:

\[
\min_{\beta, \beta_0, \xi, \eta} f_{BSVM}(\beta, \beta_0, \xi, \eta) = \frac{1}{2} ||\beta||^2 + C_1 \sum_{i=1}^n \xi_i + C_2 \sum_{i=1}^n \eta_i
\]

\[ \xi_i \geq 0 \quad \text{Lagrange multiplier } \mu_i \]

\[ \eta_i \geq 0 \quad \text{Lagrange multiplier } \psi_i \]

\[ \xi_i \geq \rho_1 - y_i(\beta^T x_i + \beta_0) \quad \text{Lagrange multiplier } \alpha_i \]

\[ \eta_i \geq -\rho_2 + y_i(\beta^T x_i + \beta_0) \quad \text{Lagrange multiplier } \theta_i \]

After introducing Lagrange multipliers for each inequality constraint as shown in 3.2, the Lagrangian function for problem 3.2 can be written as:

\[
L(\beta, \beta_0, \xi, \eta, \alpha, \theta, \mu, \psi) = \frac{1}{2} ||\beta||^2 + C_1 \sum_{i=1}^n \xi_i + C_2 \sum_{i=1}^n \eta_i - \sum_{i=1}^n \alpha_i \{\xi_i - \rho_1 + y_i(\beta^T x_i + \beta_0)\}
\]

\[ - \sum_{i=1}^n \theta_i \{\eta_i + \rho_2 - y_i(\beta^T x_i + \beta_0)\} - \sum_{i=1}^n \mu_i \xi_i - \sum_{i=1}^n \psi_i \eta_i \]

where

\[ \alpha_i, \theta_i, \mu_i, \psi_i \geq 0 \]

Next, we solve for primal variables \( \beta, \beta_0, \xi, \eta \) in terms of the dual variables \( \alpha, \theta, \mu, \psi \) by minimizing \( L(\beta, \beta_0, \xi, \eta, \alpha, \theta, \mu, \psi) \) with respect to the primal variables. Since the Lagrangian in 3.3 is a convex function of the primal variables, its unique global minimum can be obtained using the first order Karush Kuhn Tucker (KKT) conditions given in 3.5 - 3.8:

\[
\frac{\partial L}{\partial \beta} = \beta - \sum_{i=1}^n \alpha_i y_i x_i + \sum_{i=1}^n \theta_i y_i x_i = 0 \quad (3.5)
\]
After 3.5, the vector \( \beta \) is given by:

\[
\beta = \sum_{i=1}^{n} (\alpha_i - \theta_i) y_i x_i
\]  

(3.9)

From 3.6, vectors \( \alpha \) and \( \theta \) satisfy the equality constraint:

\[
\sum_{i=1}^{n} (\alpha_i - \theta_i) y_i = 0
\]  

(3.10)

Combining 3.7, 3.8 and 3.4, the elements of \( \alpha \) must satisfy:

\[
0 \leq \alpha_i \leq C_1
\]  

(3.11)

and elements of \( \theta \) satisfy:

\[
0 \leq \theta_i \leq C_2
\]  

(3.12)

Let \( B \) be a \( n \times n \) matrix with entries:

\[
B_{ij} = y_i y_j x_i^T x_j
\]  

(3.13)

and \( e_n \) be a \( n \times 1 \) vector of \( n \) ones (in MATLAB notation: \( e_n = \text{ones}(n,1) \)). Substituting \( \beta \) from 3.9 in 3.3 and noting the constraints 3.7, 3.8 and 3.10, we get the B-SVM dual problem:

\[
\max_{\alpha,\theta} L_D(\alpha, \theta) = \rho_1 e_n^T \alpha - \rho_2 e_n^T \theta - \frac{1}{2} (\alpha - \theta)^T B (\alpha - \theta)
\]  

(3.14)

\[
\begin{align*}
0 & \leq \alpha \leq C_1 e_n \\
0 & \leq \theta \leq C_2 e_n \\
(\alpha - \theta)^T y & = 0
\end{align*}
\]
3.2 Kernelifying B-SVM

Let $h$ be a non-linear vector function that takes inputs $x_i$ into a high dimensional space. Then we recover kernel B-SVM by doing linear B-SVM on the data-label pairs $(h(x_i), y_i)$ instead of the original pairs $(x_i, y_i)$. In practice, we do not need $h(x)$ explicitly but only the dot products through a kernel matrix $K$ with elements:

$$K_{ij} = K(x_i, x_j) = h(x_i)^T h(x_j) \tag{3.15}$$

This is the so-called kernel trick. From 3.13, elements of matrix $B$ for transformed feature vectors $h(x)$ are given by:

$$B_{ij} = y_i y_j K(x_i, x_j) = y_i y_j K_{ij} \tag{3.16}$$

For a new point $x$, the decision rule is then given by:

$$g(x) = \beta^T h(x) + \beta_0 \tag{3.17}$$

and $x$ is classified into class $+1$ if $g(x) > 0$ and into class $-1$ if $g(x) < 0$. From 3.9, for the transformed feature vectors $h(x_i)$, we have:

$$\beta = \sum_{i=1}^{n} (\alpha_i - \theta_i) y_i h(x_i) \tag{3.18}$$

Using the kernel trick, calculation of $g(x)$ does not need $h(x)$ explicitly as we can write:

$$g(x) = \beta^T h(x) + \beta_0 = \sum_{i=1}^{n} (\alpha_i - \theta_i) y_i K(x_i, x) + \beta_0 \tag{3.19}$$

Proposition 3.1. The B-SVM dual objective function $L_D(\alpha, \theta)$ in 3.14 is a concave function of $\alpha$ and $\theta$.

Proof. Since $B$ is symmetric, the Hessian of $L_D$ with respect to the vector $(\alpha, \theta)$ is given by:

$$H = \begin{pmatrix} -B & B \\ B & -B \end{pmatrix} \tag{3.20}$$

If $c$ and $d$ are arbitrary $n \times 1$ vectors,

$$(c^T \ d^T) H \begin{pmatrix} c \\ d \end{pmatrix} = c^T (-Bc + Bd) + d^T (Bc - Bd) = -(c - d)^T B (c - d) \tag{3.21}$$

From 3.16,

$$(c - d)^T B (c - d) = \sum_{i=1}^{n} \sum_{j=1}^{n} (c - d)_i (y_i y_j K(x_i, x_j)) (c - d)_j = \sum_{i=1}^{n} \sum_{j=1}^{n} ((c - d)_i y_i) K(x_i, x_j) ((c - d)_j y_j) \tag{3.22}$$
If $\circ$ is an element-wise multiplication operator then:

$$(c - d)^T B (c - d) = ((c - d) \circ y)^T K ((c - d) \circ y) \geq 0$$

(3.23)

where the last inequality holds since $K$ is a kernel matrix which is positive definite by 3.15. Therefore, from 3.21 and 3.23:

$$(c^T d^T) H (c d) \leq 0$$

(3.24)

for all vectors $c$ and $d$. Thus $L_D(\alpha, \theta)$ is a concave function of $(\alpha, \theta)$. $\Box$ $\Box$

It immediately follows that problem 3.14 attempts to maximize a concave function under linear constraints and thus has a unique solution [Nocedal and Wright, 2006].

### 3.3 Calculation of dual variables

Dual variables $\alpha$, $\theta$, $\mu$, $\psi$ can be calculated as follows:

- Calculation of $\alpha$, $\theta$ requires the solution of a concave maximization problem 3.14 where the elements of $B$ are chosen using a suitable kernel $K(x_i, x_j)$. This can be accomplished using an sequential minimal optimization (SMO) type active set technique [Platt, 1998] or a projected conjugate gradient (PCG) technique [Nocedal and Wright, 2006].

- Once $\alpha$ and $\theta$ are known, equations 3.7 and 3.8 give

  $\mu = C_1 e_n - \alpha$ and $\psi = C_2 e_n - \theta$.

### 3.4 Calculation of primal variables

Primal variables $\beta$, $\beta_0$, $\xi$, $\eta$ can be calculated as follows:

- $\beta$ is given by equation 3.18.

- Calculation of $\beta_0$, $\xi$, $\eta$ is accomplished by considering the inequality constraints and the KKT complementarity constraints for the problem 3.2:

  $\xi_i \geq 0, \eta_i \geq 0$  
  $\xi_i \geq \rho_1 - y_i (\beta^T h(x_i) + \beta_0)$  
  $\eta_i \geq -\rho_2 + y_i (\beta^T h(x_i) + \beta_0)$  
  $\alpha_i \{\xi_i - \rho_1 + y_i (\beta^T h(x_i) + \beta_0)\} = 0$  
  $\theta_i \{\eta_i + \rho_2 - y_i (\beta^T h(x_i) + \beta_0)\} = 0$  
  $\mu_i \xi_i = (C_1 - \alpha_i) \xi_i = 0$  
  $\psi_i \eta_i = (C_2 - \theta_i) \eta_i = 0$
Given the positivity constraints 3.4 and the bound constraints 3.11 and 3.12, we consider the following cases:

- If $\alpha_i < C_1$ then $\xi_i = 0$ and similarly if $\theta_i < C_2$ then $\eta_i = 0$.
- If $0 < \alpha_i < C_1$ then we have $\xi_i = 0$ and $\{\xi_i - \rho_1 + y_i (\beta^T x_i + \beta_0)\} = 0$ which can be used to solve for $\beta_0$.
- If $0 < \theta_i < C_2$ then we have $\eta_i = 0$ and $\{\eta_i + \rho_2 - y_i (\beta^T h(x_i) + \beta_0)\} = 0$ which can be used to solve for $\beta_0$.
- Similar to C-SVM, for stability purposes we can average the estimate of $\beta_0$ over all points where $0 < \alpha_i < C_1$ and $0 < \theta_i < C_2$.
- We can calculate $\xi_i$ for those points for which $\alpha_i = C_1$ using $\xi_i = \rho_1 - y_i (\beta^T h(x_i) + \beta_0)$.
- Similarly, if $\theta_i = C_2$ then $\eta_i = y_i (\beta^T h(x_i) + \beta_0) - \rho_2$.

4 Toy data

In order to illustrate the differences between C-SVM and B-SVM we generated artificial data in 2 dimensions as follows:

- Class 1 consisted of 5 bivariate Normal clusters centered at $(0, 0)$, $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$, $(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, and $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$ and covariance $\sigma_1^2 I_2$ with $\sigma_1 = 0.2$.
- Class $-1$ consisted of 4 bivariate Normal clusters centered at $(1, 0)$, $(0, 1)$, $(-1, 0)$ and $(0, -1)$ with covariance $\sigma_2^2 I_2$ with $\sigma_2 = 0.2$.

A radial basis function (RBF) kernel was chosen for computations. For the RBF kernel, the elements of $K$ are given by:

\[ K(x_i, x_j) = K_{ij} = \exp \left\{ -\gamma (x_i - x_j)^T (x_i - x_j) \right\} \quad (4.1) \]

Our parameter settings were as follows:

- For both C-SVM and B-SVM we used the same kernel parameter $\gamma = 1$.
- For C-SVM was used $C = 10$.
- For B-SVM we chose $\rho_1 = 1$ and $C_1 = 10$ (same as $C$ for C-SVM). Thus the parameters of the common penalty term $C_1 \sum_{i=1}^n [\rho_1 - y_i (\beta^T h(x_i) + \beta_0)]_+$ are chosen to be identical for C-SVM and B-SVM.
- The parameters of the second penalty term for B-SVM were chosen as $C_2 = 100$ and $\rho_2 = 1.5$. Thus B-SVM will encourage $g(x)$ to lie in the interval $[\rho_1, \rho_2] = [1, 1.5]$. 

10
Figure 2: Figure shows classification obtained for example data using (a) C-SVM and (b) B-SVM. Red and Blue points (.) correspond to class +1 and −1 respectively. Cyan and Orange x-marks (x) show the C-SVM and B-SVM decision rules evaluated at various points. Class 1 membership is indicated in Cyan and class −1 membership is indicated in Orange. The yellow squares in (a) correspond to support points for which $0 < \alpha_i < C$. The cyan squares in (b) correspond to support points for which $0 < \theta_i < C_2$ and the green squares correspond to support points for which $0 < \alpha_i < C_1$. The sparsity of solution is controlled by $\alpha$ in the case of C-SVM and $(\alpha - \theta)$ in the case of B-SVM. (c) Shows $\alpha_i$ values for C-SVM. (d) Shows $(\alpha_i - \theta_i)$ values for B-SVM.
Figure 3: Figure shows decision rule $g(x)$ for C-SVM (a) and B-SVM (b). Note that in B-SVM the second penalty term $C_2 \sum_{i=1}^{n} [g_i(\beta^T h(x_i) + \beta_0) - \rho_2]_+$ results in most of the $g(x)$ values in the interval $[\rho_1, \rho_2] = [1, 1.5]$. (c) Heat map of the decision rule $g(x)$ for C-SVM (d) Heat map of the decision rule $g(x)$ for B-SVM. In C-SVM the values of decision rule $g(x)$ are unbalanced in Class 1. The central cluster located at (0,0) in Class 1 gets much smaller $g(x)$ values in C-SVM than the rest of the Class 1. In B-SVM however, all clusters in Class 1 including the one centered at (0,0) get similar $g(x)$ values. This is a result of the second penalty term in the B-SVM objective function.
Figure 4: Figure shows the fraction of points classified correctly by both C-SVM (blue curve) and B-SVM (red curve) as a function of the decision rule threshold. The $x$-axis shows the decision rule threshold as a percentage of the maximum absolute value of the decision function $g(x)$ over all training points. The $y$-axis shows the overall classification accuracy or sensitivity of C-SVM and B-SVM.
Both C-SVM and B-SVM were fitted to the toy data described above. The following differences in the two solutions are noteworthy:

### 4.1 α-SVs and θ-SVs

The B-SVM dual problem 3.14 contains two variables $\alpha$ and $\theta$. Both $\alpha_i$ and $\theta_i$ are positive and satisfy the bound constraints given in 3.14. Therefore, similar to C-SVM, we define 2 types of support vectors (SVs) in B-SVM:

- Points $i$ for which $\theta_i > 0$ are called the $\theta$-SVs — new SVs that arise in B-SVM
- Points $i$ for which $\alpha_i > 0$ are called the $\alpha$-SVs — standard C-SVM like SVs

Figures 2(a) and 2(b) show the C-SVM and B-SVM induced classification respectively for this example problem. Figure 2(b) shows $\alpha$-SVs for which $0 < \alpha_i < C_1$ and $\theta$-SVs for which $0 < \theta_i < C_2$. It is clear from 3.19 that the sparsity of a B-SVM decision rule depends on the quantities $(\alpha_i - \theta_i)$. Figures 2(c) and 2(d) show a plot of $\alpha_i$ for C-SVM and $(\alpha_i - \theta_i)$ for B-SVM respectively.

### 4.2 Bounded decision rule

Figures 3(a) and 3(b) show the decision rule values $g(x)$ over all training points for C-SVM and B-SVM. Recall that C-SVM does not enforce an upper limit on $g(x)$ whereas B-SVM attempts to encourage $g(x)$ to lie in $[\rho_1, \rho_2]$. It can be seen in Figure 3(b) that B-SVM was successful in limiting the absolute value of $g(x)$ to be $< \rho_2 = 1.5$ with $C_2 = 100$. Figures 3(c) and 3(d) show a heat map of the decision rule for C-SVM and B-SVM respectively evaluated over a 2-D grid containing the training points. It can be seen that:

- The C-SVM decision rule values are unbalanced in class +1 as the central cluster in class +1 gets lower $g(x)$ values compared to other clusters in class +1.
- The decision rule values are balanced in class +1 for B-SVM.

### 4.3 Sensitivity curve

We calculate the quantity:

$$S(t) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}[y_i g(x_i) \geq t]$$

which is simply the fraction of correctly classified points (or sensitivity) using decision rule $g(x)$ at threshold $t$. To illustrate the variation in sensitivity of C-SVM and B-SVM decision rules:
For both C-SVM and B-SVM, we divide the range of \( g(x) \) into 50 equally spaced points as follows (in MATLAB notation):

\[
t = \text{linspace}(0, \max_x |g(x)|, 50)
\]  

Then we plot \( 100 \times \left( \frac{t_j}{\max_x |g(x)|} \right) \) versus \( S(t_j) \).

Figure 4 shows this sensitivity curve. It can be seen that for the same percentage threshold on the decision rule range:

- B-SVM has higher classification accuracy (or is more sensitive) than C-SVM.
- This effect is because of the balanced nature of decision rule values in B-SVM compared to C-SVM (see Figure 3(c) and 3(d)).

5 Discussion and conclusions

In this work, we considered the binary classification problem when the feature vectors in individual classes have finite co-variance. We showed that B-SVM is a natural generalization to C-SVM in this situation. It turns out that the B-SVM dual maximization problem 3.14 retains the concavity property of its C-SVM counterpart and C-SVM turns out to be a special case of B-SVM when \( C_2 = 0 \). Two types of SVs arise in B-SVM, the \( \alpha \)-SVs which are similar to the standard SVs in C-SVM and \( \theta \)-SVs which arise due to the novel B-SVM objective function penalty 2.3. The B-SVM decision rule is more balanced than the C-SVM decision rule since it assigns \( g(x) \) values that are comparable in magnitude to different sub-classes (or clusters) of class +1 and class −1. In addition, B-SVM retains higher classification accuracy compared to C-SVM as the decision rule threshold is varied from 0 to \( \max_x |g(x)| \). For a training set of size \( n \), B-SVM results in a dual optimization problem of size \( 2n \) compared to a C-SVM dual problem of size \( n \). Hence it is computationally more expensive to solve a B-SVM problem.

In summary, B-SVM can be used to enforce balanced decision rules in binary classification. It is anticipated that the C-SVM leave one out error bounds for the bias free case given in Jaakkola and Haussler [1999] will continue to hold in a similar form for bias free B-SVM as well.
References

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT ’92, pages 144–152, 1992.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

T. S. Jaakkola and D. Haussler. Probabilistic kernel regression models. In *Proceedings of the 1999 Conference on AI and Statistics*, 1999.

J. Nocedal and S. J Wright. *Numerical Optimization, 2nd Edition*. New York-Springer, 2006.

J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.