Nonlinear Support Vector Machines through Iterative Majorization and I-Splines

P.J.F. Groenen* G. Nalbantov† J.C. Bioch*

July 19, 2006

Econometric Institute Report EI 2006-25

Abstract

To minimize the primal support vector machine (SVM) problem, we propose to use iterative majorization. To do so, we propose to use iterative majorization. To allow for nonlinearity of the predictors, we use (non)monotone spline transformations. An advantage over the usual kernel approach in the dual problem is that the variables can be easily interpreted. We illustrate this with an example from the literature.

Keywords: Support vector machines, Iterative majorization, I-Splines.

1 Introduction

In recent years, support vector machines (SVMs) have become a popular technique to predict two groups out of a set of predictor variables (Vapnik, 2000). This data analysis problem is not new and such data can also be analyzed through alternative techniques such as linear and quadratic discriminant analysis, neural networks, and logistic regression. However, SVMs seem to compare favorably in their prediction quality with respect to competing models. Also, their optimization problem is well defined and can be solved through a quadratic program. Furthermore, the classification rule derived from an SVM is relatively simple and it can be readily applied to new, unseen samples. At the downside, the interpretation in terms of the predictor variables in nonlinear SVM is not always possible. In addition, the usual formulation of an SVM is not easy to grasp.

In this paper, we offer a different way of looking at SVMs that makes the interpretation much easier. First of all, we stick to the primal problem and

*Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands groenen@few.eur.nl, bioch@few.eur.nl

†ERIM, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands nalbantov@few.eur.nl

formulate the SVM in terms of a loss function that is regularized by a penalty term. From this formulation, it can be seen that SVMs use robustified errors. Then, we propose a new majorization algorithm that minimizes the loss. Finally, we show how nonlinearity can be imposed by using I-Spline transformations.

2 The SVM Loss Function

In many ways, an SVM resembles regression quite closely. Let us first introduce some notation. Let $\mathbf{X}$ be the $n \times m$ matrix of predictor variables of n objects and m variables. The $n \times 1$ vector $\mathbf{y}$ contains the grouping of the objects into two classes, that is, $y_i = 1$ if object i belongs to class 1 and $y_i = -1$ if object i belongs to class -1. Obviously, the labels -1 and 1 to distinguish the classes are unimportant. Let $\mathbf{w}$ be the $m \times 1$ vector with weights used to make a linear combination of the predictor variables. Then, the predicted value q_i for object i is

$$q_i = c + \mathbf{x}_i'\mathbf{w}, \tag{1}$$

where $\mathbf{x}_i'$ is row i of $\mathbf{X}$ and c is an intercept. Consider the example in Figure 1a where for two predictor variables, each row i is represented by a point labelled '+' for the class 1 and 'o' for class -1. Every combination of w_1 and w_2 defines a direction in this scatter plot. Then, each point i can be projected onto this line. The idea of the SVM is to choose this line in such a way that the projections of the class 1 points are well separated from those of class -1 points. The line of separation is orthogonal to the line with projections and the intercept c determines where exactly it occurs. Note that if $\mathbf{w}$ has length 1, that is, $\|\mathbf{w}\| = (\mathbf{w}'\mathbf{w})^{1/2} = 1$, then Figure 1a explains fully the linear combination (1). If $\mathbf{w}$ has not length 1, then the scale values along the projection line should be multiplied by $\|\mathbf{w}\|$. The dotted lines in Figure 1a show all those points that project to the lines at $q_i = -1$ and $q_i = 1$. These dotted lines are called the margin lines in SVMs. Note that if there are more than two variables the margin lines become hyperplanes. Summarizing, the SVM has three sets of parameters that determines its solution: (1) the regression weights, normalized to have length 1, that is, $\mathbf{w}/\|\mathbf{w}\|$, (2) the length of $\mathbf{w}$, that is, $\|\mathbf{w}\|$, and (3) the intercept c.

SVMs count an error as follows. Every object i from class 1 that projects such that $q_i \geq 1$ yields a zero error. However, if $q_i < 1$, then the error is linear with $1 - q_i$. Similarly, objects in class -1 with $q_i \leq -1$ do not contribute to the error, but those with $q_i > -1$ contribute linearly with $q_i + 1$. In other words, objects that project on the wrong side of their margin contribute to the error, whereas objects that project on the correct side of their margin yield zero error. Figure 1b shows the error functions for the two classes.

As the length of $\mathbf{w}$ controls how close the margin lines are to each other, it can be beneficial for the number of errors to choose the largest $\|\mathbf{w}\|$ possible, so that fewer points contribute to the error. To control the $\|\mathbf{w}\|$, a penalty term

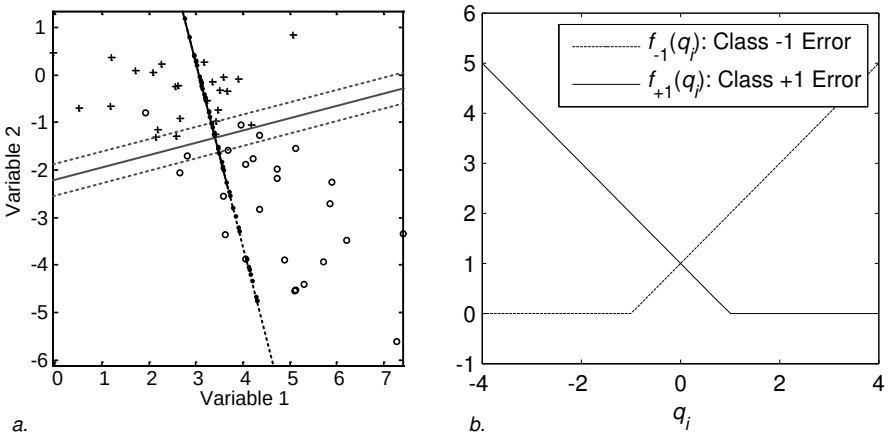

Figure 1: Panel a Projections of the observations in groups 1 (+) and −1 (o) onto the line given by w_1 and w_2. Panel b shows the error function $f_1(q_i)$ for class 1 objects (solid line) and $f_{-1}(q_i)$ for class −1 objects (dashed line).

that is dependent on $\|\mathbf{w}\|$ is added to the loss function. The penalty term also avoids overfitting of the data.

Let G_1 and G_{-1} denote the sets of class 1 and −1 objects. Then, the SVM loss function can be written as

$$\begin{array}{lcccccl} L_{\mathrm{SVM}}(c,\mathbf{w}) & & & & & & \\ & = & \sum_{i\in G_1}\max(0,1-q_i) & + & \sum_{i\in G_{-1}}\max(0,q_i+1) & + & \lambda\mathbf{w}'\mathbf{w} \\ & = & \sum_{i\in G_1} f_1(q_i) & + & \sum_{i\in G_{-1}} f_{-1}(q_i) & + & \lambda\mathbf{w}'\mathbf{w} \\ & = & \text{Class 1 errors} & + & \text{Class } -1 \text{ errors} & + & \text{Penalty for nonzero } \mathbf{w} \end{array}, \quad (2)$$

see, for similar expressions, Hastie, Tibshirani, and Friedman (2000) and Vapnik (2000).

Assume that a solution has been found. All the objects i that project on the correct side of their margin, contribute with zero error to the loss. As a consequence, these objects could be removed from the analysis without changing the solution. Therefore, all the objects i that project at the wrong side of their margin and thus induce error or if an object falls exactly on the margin, then these objects determine the solution. Such objects are called support vectors as they form the fundament of the SVM solution. Note that these objects (the support vectors) are not known in advance so that the analysis needs to be carried out with all n objects present in the analysis.

What can be seen from (2) is that any error is punished linearly, not quadratically. Thus, SVMs are more robust against outliers than a least-squares loss function. The idea of introducing robustness by absolute errors is not new. For

more information on robust multivariate analysis, we refer to Huber (1981), Vapnik (2000), and Rousseeuw and Leroy (2003).

The SVM literature usually presents the SVM loss function as follows (Burges, 1998):

$$\begin{aligned} L_{\text{SVMClas}}(c, \mathbf{w}, \xi) &= C \sum_{i \in G_1} \xi_i + C \sum_{i \in G_2} \xi_i + \frac{1}{2}\mathbf{w}'\mathbf{w}, && (3) \\ \text{subject to} \quad & 1 + (c + \mathbf{w}'\mathbf{x}_i) \leq \xi_i \text{ for } i \in G_{-1} && (4) \\ & 1 - (c + \mathbf{w}'\mathbf{x}_i) \leq \xi_i \text{ for } i \in G_1 && (5) \\ & \xi_i \geq 0, && (6) \end{aligned}$$

where C is a nonnegative parameter set by the user to weight the importance of the errors represented by the so-called slack variables ξ_i. Suppose that object i in G_1 projects at the right side of its margin, that is, $q_i = c + \mathbf{w}'\mathbf{x}_i \geq 1$. As a consequence, $1 - (c + \mathbf{w}'\mathbf{x}_i) \leq 0$ so that the corresponding ξ_i can be chosen as 0. If i projects on the wrong side of its margin, then $q_i = c + \mathbf{w}'\mathbf{x}_i < 1$ so that $1 - (c + \mathbf{w}'\mathbf{x}_i) > 0$. Choosing $\xi_i = 1 - (c + \mathbf{w}'\mathbf{x}_i)$ gives the smallest ξ_i satisfying the restrictions in (4), (5), and (6). As a consequence, $\xi_i = \max(0, 1 - q_i)$ and is a measure of error. A similar derivation can be made for class -1 objects. Choosing $C = (2\lambda)^{-1}$ gives

$$\begin{aligned} & L_{\text{SVMClas}}(c, \mathbf{w}, \xi) \\ &= (2\lambda)^{-1} \left(\sum_{i \in G_1} \xi_i + \sum_{i \in G_{-1}} \xi_i + 2\lambda \frac{1}{2}\mathbf{w}'\mathbf{w} \right) \\ &= (2\lambda)^{-1} \left(\sum_{i \in G_1} \max(0, 1 - q_i) + \sum_{i \in G_{-1}} \max(0, q_i + 1) + \lambda \mathbf{w}'\mathbf{w} \right) \\ &= (2\lambda)^{-1} L_{\text{SVM}}(c, \mathbf{w}). \end{aligned}$$

showing that the two formulations (2) and (3) are exactly the same up to a scaling factor $(2\lambda)^{-1}$ and yield the same solution. However, the advantage of (2) is that it can be interpreted as a (robust) error function with a penalty. The quadratic penalty term is used for regularization much in the same way as in ridge regression, that is, to force the w_j to be close to zero. The penalty is particularly useful to avoid overfitting. Furthermore, it can be easily seen that $L_{\text{SVM}}(c, \mathbf{w})$ is a convex function in c and $\mathbf{w}$ because all three terms are convex in c and $\mathbf{w}$. As the function is also bounded below by zero and it is convex, the minimum of $L_{\text{SVM}}(c, \mathbf{w})$ is a global one. In fact, (3) allows the problem to be treated as a quadratic program. However, in the next section, we optimize (2) directly by the method of iterative majorization.

3 A Majorizing Algorithm for SVM

In the SVM literature, the dual of (3) is reexpressed as a quadratic program and is solved by special quadratic program solvers. A disadvantage of these solvers is that they may become computationally slow for large number of objects n (although fast specialized solvers exist). However, here we derive an iterative majorization (IM) algorithm. An advantage of IM algorithms is that each iteration reduces (2). As this function is convex and IM is a guaranteed descent algorithm, the IM algorithm will stop when the estimates are sufficiently close to the global minimum.

Let $f(\mathbf{q})$ be the function to be minimized. Iterative majorization operates on an auxiliary function, called the majorizing function $g(\mathbf{q}, \overline{\mathbf{q}})$, that is dependent on $\mathbf{q}$ and the previous (known) estimate $\overline{\mathbf{q}}$. The majorizing function $g(\mathbf{q}, \overline{\mathbf{q}})$ has to fulfill several requirements: (1) it should touch f at the supporting point $\mathbf{y}$, that is, $f(\overline{\mathbf{q}}) = g(\overline{\mathbf{q}}, \overline{\mathbf{q}})$, (2) it should never be below f, that is, $f(\mathbf{q}) \leq g(\mathbf{q}, \overline{\mathbf{q}})$, and (3) $g(\mathbf{q}, \overline{\mathbf{q}})$ should be simple, preferably linear or quadratic in $\mathbf{q}$. Let $\mathbf{q}^*$ be such that $g(\mathbf{q}^*, \overline{\mathbf{q}}) \leq g(\overline{\mathbf{q}}, \overline{\mathbf{q}})$, for example, by finding the minimum of $g(\mathbf{q}, \overline{\mathbf{q}})$. Because the majorizing function is never below the original function, we obtain the so called sandwich inequality

$$f(\mathbf{q}^*) \leq g(\mathbf{q}^*, \overline{\mathbf{q}}) \leq g(\overline{\mathbf{q}}, \overline{\mathbf{q}}) = f(\overline{\mathbf{q}})$$

showing that the update $\mathbf{q}^*$ obtained by minimizing the majorizing function never increases f and usually decreases it. More information on iterative majorization can be found in De Leeuw (1994), Heiser (1995), Lange, Hunter, and Yang (2000), Kiers (2002), and Hunter and Lange (2004) and an introduction in Borg and Groenen (2005).

To find an algorithm, we need to find a majorizing function for (2). First, we derive a quadratic majorizing function for each individual error term. Then, we combine the results for all terms and come up with the total majorizing function that is quadratic in c and $\mathbf{w}$ so that an update can be readily derived. At the end of this section, we provide the majorization results.

Consider the term $f_{-1}(q) = \max(0, q + 1)$. For notational convenience, we drop the subscript i for the moment. The solid line in Figure 2 shows $f_{-1}(q)$. Because of its shape of a hinge, this function is sometimes referred to as the hinge function. Let $\overline{q}$ be the known error q of the previous iteration. Then, a majorizing function for $f_{-1}(q)$ is given by $g_{-1}(q, \overline{q})$ at the supporting point $\overline{q} = 2$. For notational convenience, we refer in the sequel to the majorizing function as $g_{-1}(q)$ without the implicit argument $\overline{q}$. We want $g_{-1}(q)$ to be quadratic so that it is of the form $g_{-1}(q) = a_{-1}q^2 - 2b_{-1}q + c_{-1}$. To find a_{-1}, b_{-1}, and c_{-1}, we impose two supporting points, one at $\overline{q}$ and the other at $-2 - \overline{q}$. These two supporting points are located symmetrically around -1. Note that the hinge function is linear at both supporting points, albeit with different gradients. Because $g_{-1}(q)$ is quadratic, the additional requirement that $f_{-1}(q) \leq g_{-1}(q)$ is satisfied if $a_{-1} > 0$ and the derivatives at the two supporting points of $f_{-1}(q)$ and $g_{-1}(q)$ are the same. More formally, the requirements are

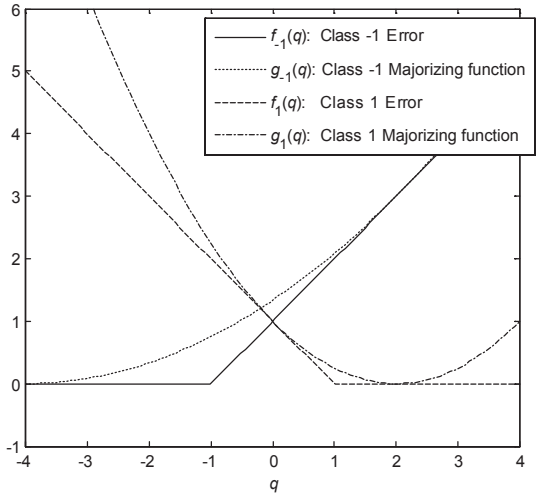

Figure 2: The error functions of Groups -1 and 1 and their majorizing functions. The supporting point is $\overline{q} = 2$. Note that the majorizing function for Group -1 also touches at $\overline{q} = -4$ and that of Group 1 also at 0.

that

$$\begin{aligned} f_{-1}(\overline{q}) &= g_{-1}(\overline{q}), \\ f'_{-1}(\overline{q}) &= g'_{-1}(\overline{q}), \\ f_{-1}(-2-\overline{q}) &= g_{-1}(-2-\overline{q}), \\ f'_{-1}(-2-\overline{q}) &= g'_{-1}(-2-\overline{q}), \\ f_{-1}(q) &\leq g_{-1}(q). \end{aligned}$$

It can be verified that the choice of

$$a_{-1} = \tfrac{1}{4}|\overline{q}+1|^{-1}, \tag{7}$$

$$b_{-1} = -a_{-1} - \tfrac{1}{4}, \tag{8}$$

$$c_{-1} = a_{-1} + \tfrac{1}{2} + \tfrac{1}{4}|\overline{q}+1|, \tag{9}$$

satisfies all these requirements. Figure 2 shows the majorizing function $g_{-1}(q)$ with supporting points $\overline{q} = 2$ or $\overline{q} = -4$ as the dotted line.

For Group 1, a similar majorizing function can be found for $f_1(q) = \max(0, 1-q)$. However, in this case, we require equal function values and first derivative at $\overline{q}$ and at $2-\overline{q}$, that is, symmetric around 1. The requirements are

$$\begin{aligned} f_1(\overline{q}) &= g_1(\overline{q}), \\ f'_1(\overline{q}) &= g'_1(\overline{q}), \\ f_1(2-\overline{q}) &= g_1(2-\overline{q}), \\ f'_1(2-\overline{q}) &= g'_1(2-\overline{q}), \\ f_1(q) &\leq g_1(q). \end{aligned}$$

Choosing

$$\begin{aligned} a_1 &= \tfrac{1}{4}|1-\overline{q}|^{-1} \\ b_1 &= a_1 + \tfrac{1}{4} \\ c_1 &= a_1 + \tfrac{1}{2} + \tfrac{1}{4}|1-\overline{q}| \end{aligned}$$

satisfies these requirements. The functions $f_1(q)$ and $g_1(q)$ with supporting points $\overline{q} = 2$ or $\overline{q} = 0$ are plotted in Figure 2.

Note that a_{-1} is not defined if $\overline{q} = -1$. In that case, we choose a_{-1} as a small positive constant δ that is smaller than the convergence criterion ϵ (introduced below). Strictly speaking, the majorization requirements are violated. However, by choosing δ small enough, the monotone convergence of the sequence of $L_{\text{SVM}}(\mathbf{w})$ will be no problem. The same holds for a_1 if $\overline{q} = 1$.

Let

$$a_i = \begin{cases} \max(\delta, a_{-1i}) & \text{if } i \in G_{-1}, \\ \max(\delta, a_{1i}) & \text{if } i \in G_1, \end{cases} \tag{10}$$

$$b_i = \begin{cases} b_{-1i} & \text{if } i \in G_{-1}, \\ b_{1i} & \text{if } i \in G_1, \end{cases} \tag{11}$$

$$c_i = \begin{cases} c_{-1i} & \text{if } i \in G_{-1}, \\ c_{1i} & \text{if } i \in G_1. \end{cases} \tag{12}$$

Then, summing all the individual terms leads to the majorization inequality

$$L_{\text{SVM}}(c, \mathbf{w}) \leq \sum_{i=1}^{n} a_i q_i^2 - 2\sum_{i=1}^{n} b_i q_i + \sum_{i=1}^{n} c_i + \lambda \sum_{j=1}^{m} w_j^2. \tag{13}$$

Because $q_i = c + \mathbf{x}_i'\mathbf{w}_i$, it is useful to add an extra column of ones as the first column of $\mathbf{X}$ so that $\mathbf{X}$ becomes $n \times (m+1)$. Let $\mathbf{v}' = [c\ \mathbf{w}']$ so that $\mathbf{q} = \mathbf{X}\mathbf{v}$. Now, (2) can be majorized as

$$\begin{aligned} L_{\text{SVM}}(\mathbf{v}) &\leq \sum_{i=1}^{n} a_i (\mathbf{x}_i'\mathbf{v})^2 - 2\sum_{i=1}^{n} b_i \mathbf{x}_i'\mathbf{v} + \sum_{i=1}^{n} c_i + \lambda \sum_{j=2}^{m+1} v_j^2 \\ &= \mathbf{v}'\mathbf{X}'\mathbf{A}\mathbf{X}\mathbf{v} - 2\mathbf{v}'\mathbf{X}'\mathbf{b} + c_m + \lambda \mathbf{v}'\mathbf{K}\mathbf{v} \\ &= \mathbf{v}'(\mathbf{X}'\mathbf{A}\mathbf{X} + \lambda\mathbf{K})\mathbf{v} - 2\mathbf{v}'\mathbf{X}'\mathbf{b} + c_m, \end{aligned} \tag{14}$$

where $\mathbf{A}$ is a diagonal matrix with elements a_i on the diagonal, $\mathbf{b}$ is a vector with elements b_i, and $c_m = \sum_{i=1}^{n} c_i$, and $\mathbf{K}$ is the identity matrix except for element $k_{11} = 0$. Differentiation the last line of (14) with respect to $\mathbf{v}$ yields the system of equalities linear in $\mathbf{v}$

$$(\mathbf{X}'\mathbf{A}\mathbf{X} + \lambda\mathbf{K})\mathbf{v} = \mathbf{X}'\mathbf{b}.$$

The update $\mathbf{v}^+$ solves this set of linear equalities, for example, by Gaussian elimination, or, somewhat less efficiently, by

$$\mathbf{v}^+ = (\mathbf{X}'\mathbf{A}\mathbf{X} + \lambda\mathbf{K})^{-1}\mathbf{X}'\mathbf{b}. \tag{15}$$

$t = 0$;
Set ϵ to a small positive value;
Set $\mathbf{w}_0$ and c_0 to a random initial values;
Compute $L_{\text{SVM}}(c_0, \mathbf{w}_0)$ according to (2);
Set $L_{-1} = L_{\text{SVM}}(c_0, \mathbf{w}_0) + 2\epsilon$;
while $L_{t-1} - L_{\text{SVM}}(c_t, \mathbf{w}_t) > \epsilon$ **do**
    $t = t + 1$;
    $L_{t-1} = L_{\text{SVM}}(c_{t-1}, \mathbf{w}_{t-1})$;
    Compute the diagonal matrix $\mathbf{A}$ with elements defined by (10);
    Compute the $\mathbf{b}$ with elements defined by (11);
    Find $\mathbf{v}^+$ by solving (15);
    Set $c_t^+ = v_1$ and $w_{tj}^+ = v_{j+1}^+$ for $j = 1, \ldots, m$;
end

Figure 3: The SVM majorization algorithm.

Because of the substitution $\mathbf{v}' = [c\ \mathbf{w}']$, the update of the intercept is $c^+ = v_1$ and $w_j^+ = v_{j+1}^+$ for $j = 1, \ldots, m$. The update $\mathbf{v}^+$ forms the heart of the majorization algorithm for SVMs.

The majorizing algorithm for minimizing the standard SVM in (2) is summarized in Figure 3. This algorithm has several advantages. First, it iteratively approaches the global minimum closer in each iteration. In contrast, quadratic programming of the dual problem need to solve the dual problem completely to have the global minimum of the original primal problem. Secondly, the progress can be monitored, for example, in terms of the changes in the number of misclassified objects. Thirdly, to reduce the computational time, smart initial estimates of c and $\mathbf{w}$ can be given if they are available, for example, from a previous cross validation run.

An illustration of the iterative majorization algorithm is given in Figure 4. Here, c is fixed at its optimal value and the minimization is only over $\mathbf{w}$, that is, over w_1 and w_2. Each point in the horizontal plane represents a combination of w_1 and w_2. The majorization function is indeed located above the original function and touches it at the dotted line. The w_1 and w_2 where this majorization function finds its minimum, $L_{\text{SVM}}(c, \mathbf{w})$ is lower than at the previous estimate, so $L_{\text{SVM}}(c, \mathbf{w})$ has decreased. Note that the separation line and the margins corresponding to the current estimates of w_1 and w_2 are given together with the class 1 points represented as open circles and the class -1 points as closed circles.

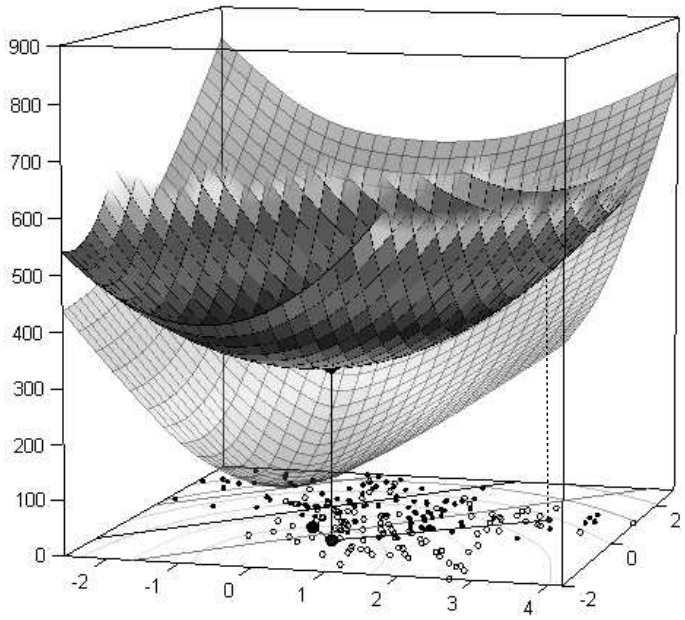

Figure 4: Example of the iterative majorization algorithm for SVMs in action where c is fixed and w_1 and w_2 are being optimized. The majorization function touches $L_{\text{SVM}}(c, \mathbf{w})$ at the previous estimates of $\mathbf{w}$ (the dotted line) and a solid line is lowered at the minimum of the majorizing function showing a decrease in $L_{\text{SVM}}(c, \mathbf{w})$ as well.

4 Nonlinear SVM

The SVM described so far tries to find a linear combination $\mathbf{q} = \mathbf{Xb}$ such that negative values are classified into class -1 and positive values into class 1. As a consequence, there is a separation hyperplane of all the points that project such that $q = 0$. Therefore, the standard SVM has a linear separation hyperplane. To allow for a nonlinear separation plane, the the classical approach is to turn to the dual problem and introduce kernels. By doing so, the relation with the primal problem $L_{\text{SVM}}(c, \mathbf{w})$ is lost and the interpretation in terms of the original variables is not always possible anymore.

To cover nonlinearity, we use the optimal scaling ideas from Gifi (1990). In particular, each predictor variable is being transformed. A powerful class of transformations is formed by spline transformation. The advantage of splines

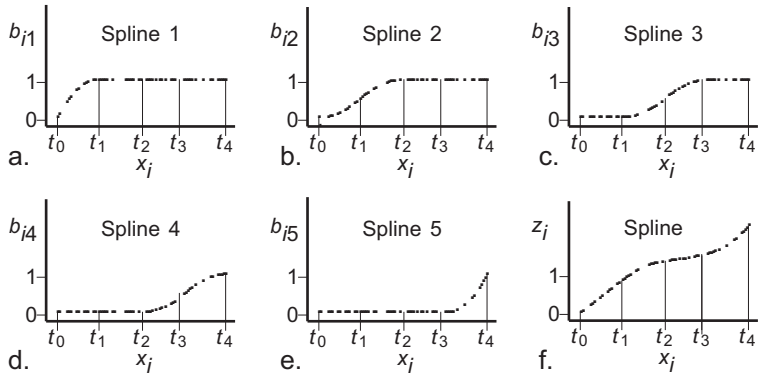

Figure 5: Panels a to e give an example of the columns of the spline basis $\mathbf{B}$. It consists of five columns given the degree $d = 2$ and the number of interior knots $k = 3$. Panel f shows a linear sum of these bases that is the resulting I-Spline transformation $\mathbf{z}$ for a single predictor variable $\mathbf{x}$.

is that they yield transformations that are piecewise polynomial. In addition, the transformation is smooth. Because the resulting transformation consists of polynomials whose coefficients are known, the spline basis values can also be computed for unobserved points. Therefore, the transformed value of test points in can be easily computed.

There are various sorts of spline transformations, but here we choose the I-Spline transformations (see Ramsay, 1988). An example of such a transformation is given in Figure 5f. In this case, the piecewise polynomial consists of four intervals. The boundary points between subsequent intervals are called interior knots t_k. The interior knots are chosen such that the number of observations is about equal in each interval. The degree of the polynomial d in the I-Spline transformation of Figure 5f is 2 so that each piece is quadratic in the original predictor variable $\mathbf{x}_j$. Once the number of interior knots k and the degree d are fixed, each I-Spline transformation can be expressed as $\mathbf{z}_j = \mathbf{B}_j\mathbf{w}_j$ where $\mathbf{B}_j$ is the so called spline basis of $n \times (d+k)$. For the example transformation in Figure 5f, the columns of $\mathbf{B}_j$ are visualized in Figures 5a to 5e. One of the properties of the I-Spline transformation is that if the weights $\mathbf{w}_j$ are all positive, then the transformation is monotone increasing as in our example as in Figure 5f. This property is of use to interpret the solution.

To estimate the transformations in the SVM problem, we simply replace $\mathbf{X}$ by the matrix $\mathbf{B} = [\mathbf{B}_1\ \mathbf{B}_2\ \ldots\ \mathbf{B}_m]$, that is, by concatenating the spline bases $\mathbf{B}_j$, one for each original predictor variable $\mathbf{x}_j$. In our example, we have $m = 2$ variables ($\mathbf{x}_1$ and $\mathbf{x}_2$), $d = 2$, and $k = 3$, so that $\mathbf{B}_1$ and $\mathbf{B}_2$ are both matrices of size $n \times (d+k) = n \times 5$ and $\mathbf{B}$ is of size $n \times m(d+k) = n \times 10$. Then, the vector of weights $\mathbf{w}' = [\mathbf{w}_1'\ \mathbf{w}_2'\ \ldots\ \mathbf{w}_m']$ is of size $m(d+k) \times 1$ and the transformation $\mathbf{z}_j$ of a single variable $\mathbf{x}_j$ is given by $\mathbf{z}_j = \mathbf{B}_j\mathbf{w}_j$. Thus, to model the nonlinearity

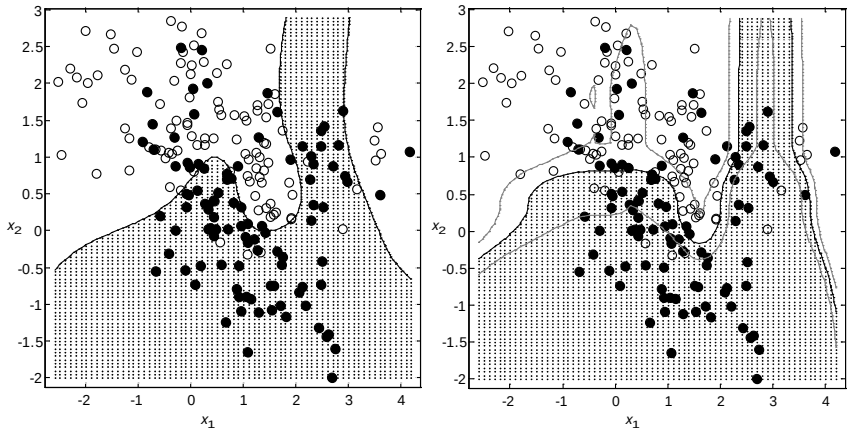

Figure 6: The left panel shows samples from a mixture distribution of two groups (Hastie et al., 2000) and a sample of points of these groups. The line is the optimal Bayes decision boundary. The right panel shows the SVM solution using spline transformations of degree 2 and 5 interior knots, $\lambda = .00316$, with an training error rate of .21.

in the decision boundary, we extend the space of predictor variables from $\mathbf{X}$ to the space of the spline bases of all predictor variables and then search through the SVM for a linear separation in this high dimensional space.

Consider the example of a mixture of 20 distributions for two groups given by Hastie et al. (2000) on two variables. The left panel of Figure 6 shows a sample 200 points with 100 in each class. It also shows the optimal Bayes decision boundary. The right panel of Figure 6 shows the results of the SVM with I-Spline transformations of the two predictor variables using $k = 5$ and $d = 2$. After cross validation, the best preforming $\lambda = .00316$ yielding a training error rate of .21.

Once the SVM is estimated and $\mathbf{w}$ is known, the transformations $\mathbf{z}_j = \mathbf{B}_j\mathbf{w}_j$ are determined. Thus, each interval of the transformation is in our example with $d = 2$ a quadratic function in $\mathbf{x}_j$ for which the polynomial coefficients can be derived. As test points, we use a grid in the space of the two predictor variables. Because the polynomial coefficients are known for each interval, we can derive the transformed (interpolated) value z_{ij} of test point i for each j and the value $q_i = c + \sum_j z_{ij}$ where c is the intercept. If q_i for the test point is positive, we classify the test point i in class 1, if $q_i < 0$ in class -1, and if $q_i = 0$ it is on the decision boundary. This classification is done for all the test points in the grid, resulting in the reconstructed boundary in the right panel of Figure 6.

I-Splines have the property that for nonnegative $\mathbf{w}$ the transformation $\mathbf{z}_j$ is monotone increasing with $\mathbf{x}_j$. Let $\mathbf{w}_j^+ = \max(\mathbf{0}, \mathbf{w}_j)$ and $\mathbf{w}_j^- = \min(\mathbf{0}, \mathbf{w}_j)$ so that $\mathbf{w}_j = \mathbf{w}_j^+ + \mathbf{w}_j^-$. Then, the transformation $\mathbf{z}_j$ can be split in a monotone

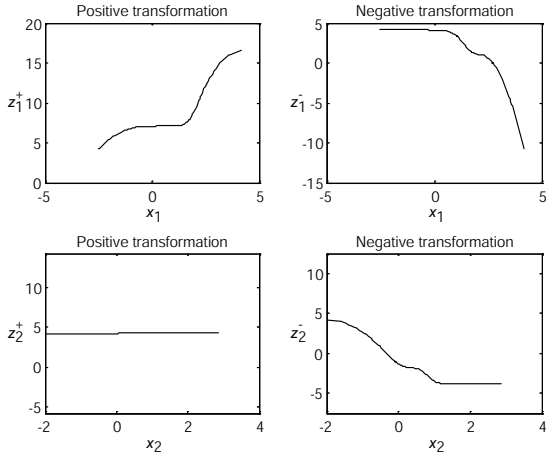

Figure 7: Spline transformations of the two predictor variables used in the SVM solution in the right panel of Figure 6.

increasing part $\mathbf{z}_j^+ = \mathbf{B}_j \mathbf{w}_j^+$ and a monotone decreasing part $\mathbf{z}_j^- = \mathbf{B}_j \mathbf{w}_j^+$. For the mixture example, these transformations are shown in Figure 7 for each of the two predictor variables. From this figure, we see that for $\mathbf{x}_1$ the nonlinearity is caused by the steep transformations of values for $x_1 > 1$ both for the positive as for the negative part. For $\mathbf{x}_2$, the nonlinearity seems to be caused by only by the negative transformation for $x_2 < 1$.

5 Conclusions

We have discussed how SVM can be viewed as a the minimization of a robust error function with a regularization penalty. Nonlinearity was introduced by mapping the space of each predictor variable into a higher dimensional space using I-Spline basis. The regularization is needed to avoid overfitting in the case when the number of predictor variables increases or the their respective spline bases become of high rank. The use of I-Spline transformations are useful to allow interpreting the nonlinearity in the prediction. We also provided a new majorization algorithm for the minimization of the primal SVM problem.

There are several open issues and possible extensions. A disadvantage of the I-Spline transformation over the usual kernel approach is that the degree of the spline d and the number of interior knots k need to be set whereas most standard kernels just have a single parameter. We need more numerical experience to study what good ranges for these parameters are.

The present approach can be extended to other error functions as well. Also, there seems to be close relations with the optimal scaling approach taken in multidimensional scaling and by the work of Gifi (1990). We intend to study

these issues in subsequent publications.

References

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications (2nd edition).* New York: Springer.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, *2*, 121–167.

De Leeuw, J. (1994). Block relaxation algorithms in statistics. In H.-H. Bock, W. Lenski, & M. M. Richter (Eds.), *Information systems and data analysis* (pp. 308–324). Berlin: Springer.

Gifi, A. (1990). *Nonlinear multivariate analysis.* Chichester: Wiley.

Hastie, T., Tibshirani, R., & Friedman, J. (2000). *The elements of statistical learning.* New York: Springer.

Heiser, W. J. (1995). Convergent computation by iterative majorization: Theory and applications in multidimensional data analysis. In W. J. Krzanowski (Ed.), *Recent advances in descriptive multivariate analysis* (pp. 157–189). Oxford: Oxford University Press.

Huber, P. J. (1981). *Robust statistics.* New York: Wiley.

Hunter, D. R., & Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, *39*, 30–37.

Kiers, H. A. L. (2002). Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems. *Computational Statistics and Data Analysis*, *41*, 157–170.

Lange, K., Hunter, D. R., & Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, *9*, 1–20.

Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, *3*(4), 425–461.

Rousseeuw, P. J., & Leroy, A. M. (2003). *Robust regression and outlier detection.* New York: Wiley.

Vapnik, V. N. (2000). *The nature of statistical learning theory.* New York: Springer.