Characterizing the Solution Path of Multicategory Support Vector Machines

Yoonkyung Lee, The Ohio State University
Zhenhuan Cui, The Ohio State University

Technical Report No. 754

April, 2005

Department of Statistics
The Ohio State University
1958 Neil Avenue
Columbus, OH 43210-1247
CHARACTERIZING THE SOLUTION PATH OF MULTICATEGORY SUPPORT VECTOR MACHINES

Yoonkyung Lee and Zhenhuan Cui

Department of Statistics, The Ohio State University

April 24, 2005

Abstract

The algorithm of fitting the entire regularization path of the support vector machine (SVM) was recently proposed by Hastie et al. (2004). It allows effective computation of solutions and greatly facilitates the choice of the regularization parameter that balances a trade-off between complexity of a solution and its fit to data. Extending the idea to more general setting of the multiclass case, we characterize the coefficient path of the multicategory SVM via the complementarity conditions for optimality. The extended algorithm provides a computational shortcut to attain the entire spectrum of solutions from the most regularized to the completely overfitted ones.

Key words and phrases: Classification, coefficient paths, Karush-Kuhn-Tucker condition, multicategory support vector machine.

1 Introduction

Regularization methods are widely used in statistics and machine learning for data analysis. A few examples include smoothing splines (Wahba 1990), penalized logistic regression, and support vector machines (Vapnik 1998). The effectiveness of a regularization method often largely depends on the choice of a regularization parameter (or tuning parameter) which controls model elaboration in a continuous fashion. The LAR (least angle regression) by Efron et al. (2004) and the SVM path by Hastie et al. (2004) showcase recent developments
of computational algorithms to characterize the entire regularization path in place of a user-dependent grid search of a good solution. These constructive algorithms do not only enable efficient computation of solutions along the path, but also provide a bird’s-eye view of the spectrum of solutions from the least to the most complex fit to given data.

This paper focuses on construction of the support vector machine (SVM) solution path for classification. As illustrated in Hastie et al. (2004), capability of solving a system of linear equations is sufficient to find the complete solution path of the binary SVM as a function of its tuning parameter. In other words, the whole range of SVM solutions can be obtained without resorting to an external quadratic programming solver except for one-time initialization if the two classes are unbalanced. Differently from this approach, there are quite a few popularly used algorithms for the SVM such as SMO (Sequential Minimal Optimization) (Platt 1999), SVM light (Joachims 1999), and LIBSVM (Hsu and Lin 2002). However, they are only tailored for scalable computation of the SVM solution at a single value of the tuning parameter, proper specification of which would require a non-trivial inspection. Moreover, Hastie et al. (2004) empirically demonstrated that the computational cost of obtaining the entire regularization path could be almost the same as getting a single solution by other methods, while misspecification of the tuning parameter can be readily avoided through the SVM path.

Motivated by the idea of sequentially finding the SVM path for the binary case, we extend the algorithm to the multiclass case for general treatment of classification. This extension is for the Multicategory SVM (MSVM) in Lee, Lin and Wahba (2004) that subsumes the binary SVM as a special case and retains the same problem structure. The Karush-Kuhn-Tucker optimality conditions (Mangasarian 1994) for the corresponding optimization problems play an important role in fully determining solution paths as a function of the regularization parameter $\lambda$. Hastie et al. (2004) cleverly utilized the conditions to show that the SVM coefficient path is piecewise linear in $1/\lambda$. In this paper, we draw a parallel to this idea and necessary derivations for the multiclass case. It is established that the MSVM coefficient path is also piecewise linear in $1/\lambda$ with additional number of joints roughly proportional to the number of classes. The joints of the piecewise linear solution path are identified as the values of $\lambda$ at which any of data points on the margin of MSVM coordinates changes. The entire coefficient path is then constructed sequentially, and it provides a computational shortcut to simultaneous fitting and tuning. The extended algorithm of finding the MSVM coefficient path seamlessly encompasses that for the binary SVM, contributing further to our general understanding of the structure of the SVM formulation. When developing the analogous
algorithm for the multiclass case, we will closely follow the notations and terminologies used in Hastie et al. (2004) for the binary case, in order to illuminate the connection between them.

This paper is organized as follows. Section 2 briefly reviews the optimization problem of the MSVM and states the optimality conditions for the solution. Section 3 characterizes each coefficient path as a piecewise linear function via the optimality conditions. Section 4 discusses how to find the joints that determine the solution path and other computational issues. Section 5 presents a numerical example that illustrates the constructive algorithm when applied to simulated data. Concluding remarks and future directions are given at the end.

2 Multicategory SVM and Optimality Conditions

In the classification problem, we are given a training data set of \( n \) pairs of covariates and a known class label \((x_i, y_i)\) for \( i = 1, \ldots, n \). \( x_i \in \mathbb{R}^p \) represents the covariates of the \( i \)th observation and the response \( y_i \in \{1, \ldots, k\} \) denotes the class that it falls into. \( y_i \) can be one of \( k \) predefined categories. In general, a classification rule \( \phi(x) : \mathbb{R}^p \rightarrow \{1, \ldots, k\} \) is constructed based on the training data such that it can generalize the relationship between \( x_i \) and the class label \( y_i \) to novel cases.

The Multicategory SVM proposed by Lee, Lin and Wahba (2004) is a general classification method, which extends good theoretical properties of the binary SVM to the multiclass case. It follows the general scheme of finding a \( k \)-tuple of functions \( f(x) = (f_1(x), \ldots, f_k(x)) \), which induces a classifier \( \phi(x) = \text{argmax}_{j=1,\ldots,k} f_j(x) \) via the maximum component. Note that superscripts are used to indicate coordinates. We consider each component \( f_j(x) \) as an element of a reproducing kernel Hilbert space (RKHS), \( \mathcal{H} = \{1\} \oplus \mathcal{H} \). Then, each coordinate \( f_j(x) \) can be expressed as \( b_j + h_j(x) \) with \( h_j \in \mathcal{H} \). A vector valued class code \( y_i \) is to be used in place of the nominal class label \( y_i \). If the \( i \)th observation is from class \( j \), \( y_i = (y_i^1, \ldots, y_i^k) \) is defined as a vector of contrast with \( y_i^j = 1 \) and \(-1/(k-1)\) elsewhere. Generally, a regularization method in an RKHS can be cast as a problem of finding \( f(x) = b + h(x) \in \mathcal{H} \) in the RKHS that minimizes

\[
\frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda \|h\|^2.
\]

\( L(y_i, f(x_i)) \) is a loss function measuring goodness of fit, \( \| \cdot \| \) is the norm defined on the RKHS \( \mathcal{H} \), and \( \lambda \) is a tunable regularization parameter which balances the empirical risk.
and the penalty associated with \( f \). In this regularization framework, the MSVM solution \( \hat{f}_\lambda(x) = (\hat{f}^1_\lambda(x), \ldots, \hat{f}^k_\lambda(x)) \) given \( \lambda \) is defined as the minimizer of

\[
\frac{1}{n} \sum_{i=1}^{n} L(y_i)^t(f(x_i) - y_i)_{+} + \frac{\lambda}{2} \sum_{j=1}^{k} \|h^j\|^2,
\]

(1)

with the sum-to-zero constraint \( \sum_{j=1}^{k} f^j(x) = 0 \) for any \( x \in \mathbb{R}^p \). To explain the loss function, let \( \text{cat}(i) \) be the category of \( y_i \), and \( L^j_i \) be the cost of misclassifying \( j \) as \( j' \), and define the misclassification cost vector \( L(y_i) = (L^1_{\text{cat}(i)}, \ldots, L^k_{\text{cat}(i)})^t \). Then, the so-called hinge loss function is \( \mathcal{L}(y_i, f(x_i)) = L(y_i)^t(f(x_i) - y_i)_{+} \). It can be written explicitly as \( \mathcal{L}(y_i, f(x_i)) = \sum_{j=1}^{k} L^j_{\text{cat}(i)}(f^j(x_i) - y^j_i)_{+} \), where \( (x)_{+} = \max(x, 0) \). For the equal misclassification costs, that is, \( L^j_i = I(j \neq j') \), it is simplified as \( \mathcal{L}(y_i, f(x_i)) = \sum_{j \neq \text{cat}(i)}(f^j(x_i) + 1/(k - 1))_{+} \). By the representer theorem, \( \hat{f}_\lambda = (\hat{f}^1_\lambda, \ldots, \hat{f}^k_\lambda) \) is of the form

\[
\hat{f}^j_\lambda(x) = b^j + \sum_{i=1}^{n} c^j_i K(x_i, x) \quad \text{for} \quad j = 1, \ldots, k,
\]

(2)

where \( K(s, t) \) is the reproducing kernel of \( \mathcal{H} \).

The main focus of this paper is how to explicitly characterize the coefficient paths of \( b^j \) and \( c^j_i \) of the solution as a function of \( \lambda \). For expositions to follow, we briefly discuss the optimization problems associated with (1). Let the coefficient vector \( c^j = (c^j_1, \ldots, c^j_n)^t \) for \( j = 1, \ldots, k \), \( b = (b^1, \ldots, b^k)^t \), and \( C = (c^1, \ldots, c^k) \). With some abuse of notations, let the bold-faced \( K \) stand for \( n \times n \) matrix with the \( lm \)th entry \( K(x_i, x_m) \). Also, let \( L^j \) denote the \( j \)th coordinates of the \( n \) misclassification cost vectors, \( (L^j_{\text{cat}(1)}, \ldots, L^j_{\text{cat}(n)})^t \) and \( y^j = (y^j_1, \ldots, y^j_n)^t \). Then the MSVM in (1) can be rewritten as a problem of finding \( (b, C) \) that minimizes

\[
L_P(b, C) = \frac{1}{n} \sum_{j=1}^{k} (L^j)^t(b^j e + Kc^j - y^j)_{+} + \frac{\lambda}{2} \sum_{j=1}^{k} (c^j)^t Kc^j
\]

(3)

subject to \( \sum_{j=1}^{k} (b^j e + Kc^j) = 0 \),

(4)

where \( e \) is the vector of \( n \) ones. To handle the truncate function \( (x)_{+} \) in (3), we introduce non-negative slack variables denoted by \( \xi^j = (\xi^j_1, \ldots, \xi^j_n)^t \) for \( j = 1, \ldots, k \). Let \( \xi = (\xi^1, \ldots, \xi^k) \). By using the slack variables, (3) and (4) can be reformulated as finding \( b, C, \) and \( \xi \) that minimize

\[
L_P(b, C, \xi) = \frac{1}{n} \sum_{j=1}^{k} (L^j)^t \xi^j + \frac{\lambda}{2} \sum_{j=1}^{k} (c^j)^t Kc^j
\]

(5)
subject to \[ b^j e + K c^j - y^j \leq \xi^j, \quad \text{for } j = 1, \ldots, k, \] \[ \xi^j \geq 0, \quad \text{for } j = 1, \ldots, k, \] and \[ \sum_{j=1}^{k} (b^j e + K c^j) = 0. \] (8)

The minimizers are determined by solving the dual quadratic programming problem related with the above reformulation.

For Lagrangian dual formulation of the problem, we introduce nonnegative Lagrange multipliers \( \alpha^j = (\alpha^j_1, \ldots, \alpha^j_n)^t \in \mathbb{R}^n \) for (6), \( \gamma^j \in \mathbb{R}^n \) for (7), and unconstrained Lagrange multipliers \( \delta_f \in \mathbb{R}^n \) for (8). Then, the dual problem becomes a problem of maximizing

\[
L_D = \sum_{j=1}^{k} (L^j)^t \xi^j + \frac{n \lambda}{2} \sum_{j=1}^{k} (c^j)^t K c^j + \sum_{j=1}^{k} (\alpha^j)^t (b^j e + K c^j - y^j - \xi^j) - \sum_{j=1}^{k} (\gamma^j)^t \xi^j + \delta_f^t \sum_{j=1}^{k} (b^j e + K c^j)
\]

subject to for \( j = 1, \ldots, k \)

\[
\frac{\partial L_D}{\partial \xi^j} = L^j - \alpha^j - \gamma^j = 0 \quad (9)
\]

\[
\frac{\partial L_D}{\partial c^j} = n \lambda K c^j + K \alpha^j + K \delta_f = 0 \quad (10)
\]

\[
\frac{\partial L_D}{\partial b^j} = (\alpha^j + \delta_f)^t e = 0 \quad (11)
\]

\[
\alpha^j \geq 0 \quad (12)
\]

\[
\gamma^j \geq 0 \quad (13)
\]

Letting \( \bar{\alpha} = (\sum_{j=1}^{k} \alpha^j)/k \), we have \( (\alpha^j - \bar{\alpha})^t e = 0 \) by taking the unconstrained \( \delta_f = -\bar{\alpha} \) in (11) and \( c^j = -(\alpha^j - \bar{\alpha})/(n \lambda) \) from (10). Using these relations and (9), and denoting \((\alpha^1, \ldots, \alpha^k)\) by \( \alpha \), we have the following dual quadratic programming problem.

\[
\min L_D(\alpha) = \frac{1}{2} \sum_{j=1}^{k} (\alpha^j - \bar{\alpha})^t K (\alpha^j - \bar{\alpha}) + n \lambda \sum_{j=1}^{k} (\alpha^j)^t y^j \quad (14)
\]

subject to \( 0 \leq \alpha^j \leq L^j \) for \( j = 1, \ldots, k \)

\[
(\alpha^j - \bar{\alpha})^t e = 0 \quad \text{for } j = 1, \ldots, k. \quad (16)
\]

Note that the \( \alpha^j \)'s corresponding to zero \( L^j_{\text{cat}(i)} \)'s are trivially zero, so the above dual problem involves only \( n(k - 1) \) Lagrange multipliers. Throughout this paper, we consider only the
Figure 1: MSVM component loss \((f^j - y^j)_+\) where \(y^j = -1/(k - 1)\).

\(n(k - 1)\) non-trivial \(\alpha_i^j\). By the Karush-Kuhn-Tucker (KKT) complementarity conditions, the solution satisfies

\[
\alpha^j \perp (b^j e + Kc^j - y^j - \xi^j) \quad \text{for } j = 1, \ldots, k \tag{17}
\]

\[
\gamma^j = (L^j - \alpha^j) \perp \xi^j \quad \text{for } j = 1, \ldots, k \tag{18}
\]

where \(\perp\) indicates that componentwise products of two vectors are zero. For instance, if \(0 < \alpha_i^j < L_i^j\) for some \(i\), then \(\xi_i^j\) should be zero from (18), and this implies that \(b^j + \sum_{i=1}^n c_i^j K(x_l, x_i) - y_i^j = 0\) from (17).

The KKT conditions in (17) and (18) categorize each component of \(\hat{f}_\lambda(x_i)\) as one of three types defining three different sets. To refer to the three sets of indices, we borrow the names from Hastie et al. (2004) but slightly modify them as follows. Abbreviating \(\hat{f}_\lambda^j(x_i)\) as \(f_i^j\),

- \(\mathcal{E} = \{(i, j) | f_i^j - y_i^j = 0, \xi_i^j = 0, 0 \leq \alpha_i^j \leq L_{\text{cat}(i)}^j\}\) an elbow set,
- \(\mathcal{U} = \{(i, j) | f_i^j - y_i^j > 0, \xi_i^j > 0, \alpha_i^j = L_{\text{cat}(i)}^j\}\) an upper set of the elbow, and
- \(\mathcal{L} = \{(i, j) | f_i^j - y_i^j < 0, \xi_i^j = 0, \alpha_i^j = 0\}\) a lower set of the elbow.

Figure 1 depicts the \(j\)th component of the MSVM hinge loss \((f^j - y^j)_+\) as a function of \(f^j\) with \(y^j = -1/(k - 1)\). The elbow set \(\mathcal{E}\) consists of indices of data points falling on the soft margin of the MSVM solution while the lower set \(\mathcal{L}\) is associated with non-support vectors.

3 Solution Paths

To describe how MSVM solutions in (2) change as a function of the regularization parameter \(\lambda\), we begin with a very large value of the parameter at which the initial set of dual variables
Lemma 1 is easily determined. Then a constructive algorithm is laid out for successive update of the dual minimizers as \( \lambda \) decreases. From the relation between the coefficients and the dual variables, \( c^j = -(\alpha^j - \bar{\alpha})/(n\lambda), \) \( c^j = 0 \) as \( \lambda \) goes to \( \infty \). So, only \( b^j \) and \( \alpha^j \) need to be initialized for a sufficiently large \( \lambda \). For brevity of explanation, the equal misclassification costs are considered, where \( L_j' = 1(j \neq j') \). Generalization of the following results to unequal misclassification costs is straightforward.

### 3.1 Initialization

Let \( I_j \) be the index set of observations in class \( j \) and \( n_j = |I_j| \), the number of instances in class \( j \) for \( j = 1, \ldots, k \). Since initialization of \( b^j \) and \( \alpha^j \) depends on which class is the largest in terms of the sample size, we define \( M = \arg\max n_j \) and \( n_M = \max n_j \) first. Lemma 1 below is concerned with initialization when \( |M| = 1 \), that is, there is a unique class with the maximum sample size while Lemma 2 is for \( |M| > 1 \). The results presented here subsume Lemma 1 and 2 in Hastie et al. (2004) for the binary case.

**Lemma 1** Suppose that there is only one class \( j^* \) with the maximum sample size. For a sufficiently large \( \lambda \), \( b^i = 1 \) if \( i = j^* \), and \(-1/(k-1)\) otherwise. \( \alpha \) minimizes

\[
\begin{align*}
\sum_{j=1}^{k}(\alpha^j - \bar{\alpha})^t K(\alpha^j - \bar{\alpha}) \\
\text{subject to} & \quad 0 \leq \alpha^j_i \leq 1 \quad \text{for } j \neq j^* \text{ and } i \notin I_j \quad (19) \\
& \quad \alpha_i^{j^*} = 1 \quad \text{for } i \notin I_j. \quad (20) \\
& \quad \sum_i \alpha^j_i = n - n_j \quad \text{for } j \neq j^*. \quad (21)
\end{align*}
\]

**Proof.**

For a sufficiently large \( \lambda \), the minimizer of (3) is a constant vector \((b^1, \ldots, b^k)\), for which (3) is reduced to \( \sum_{j=1}^{k}(n - n_j)(b^j + 1/(k-1))_+ \) up to a multiplicative constant. To minimize the objective function, note that all the \( b^j \)'s need to be at least \(-1/(k-1)\). Otherwise, the objective function can be always made smaller. Thus, it amounts to finding \((b^1, \ldots, b^k)\) minimizing \( \sum_{j=1}^{k}(n - n_j)(b^j + 1/(k-1)) = \sum_{j \neq j^*}(n_j - n_j)b^j + n \). Since it is a non negatively weighted sum of \( b^j \)'s for \( j \neq j^* \), the sum becomes the smallest when \( b^j = -1/(k-1) \) for \( j \neq j^* \), and consequently \( b^{j^*} = 1 \) by the sum-to-zero constraint. The rest follows from (14), (15), and (16) by observing the following three facts. First, \( \xi_i^{j^*} = (b^{j^*} - y_i^{j^*})_+ = k/(k-1) \) for \( i \notin I_j \), thus \( \alpha_i^{j^*} = 1 \) to satisfy the KKT conditions of (17) and (18). Second, (16) is then restated as \( \sum_i \alpha_i^j = n - n_j \) for all \( j \). Third, as a result, \( \sum_{j=1}^{k}(\alpha^j)^t y^j = -1/(k-1) \sum_{j=1}^{k} \sum_i \alpha_i^j \) is fixed at \(-k/(k-1)(n - n_{j^*})\). \( \square \)
Remark 1 The value of the primal objective function (5) for the initial $b$ is

$$\sum_{j=1}^{k} (L^j)^t \xi^j = (n - n_{j^*})k/(k - 1),$$

except the multiplicative constant $1/n$. Lemma 1 is a generalized version of Lemma 2 for the unbalanced binary case in Hastie et al. (2004).

Lemma 2 Suppose that there are more than one class in $\mathcal{M} = \arg\max n_j$. For a sufficiently large $\lambda$, $b^j = -1/(k - 1)$ for $j \notin \mathcal{M}$, and $b^j \geq -1/(k - 1)$ for $j \in \mathcal{M}$ with $\sum_{j \in \mathcal{M}} b^j = (k - |\mathcal{M}|)/(k - 1)$. $\alpha$ minimizes

$$\sum_{j=1}^{k} (\alpha^j - \bar{\alpha})^t K(\alpha^j - \bar{\alpha})$$

subject to

$$0 \leq \alpha^j_i \leq 1 \quad \text{for } j \notin \mathcal{M} \text{ and } i \notin I_j$$

$$\alpha^j_i = 1 \quad \text{for } j \in \mathcal{M} \text{ and } i \notin I_j$$

(22)

$$\sum_i \alpha^j_i = n - n_{\mathcal{M}} \quad \text{for } j \notin \mathcal{M}.$$  (23)

Proof.

By the same arguments used in the proof of Lemma 1, finding the minimizer $(b^1, \ldots, b^k)$ of the primal objective function leads to searching $(b^1, \ldots, b^k)$ that minimizes $\sum_{j \notin \mathcal{M}} (n_{\mathcal{M}} - n_j)b^j + n$ with $b^j \geq -1/(k - 1)$. Hence, $b^j = -1/(k - 1)$ for $j \notin \mathcal{M}$ and the remaining $b^j$'s are arbitrary except that they satisfy $\sum_{j \in \mathcal{M}} b^j = (k - |\mathcal{M}|)/(k - 1)$ by the sum-to-zero constraint. From the equality constraint on $b^j$'s with $j \in \mathcal{M}$, we infer that there is at least one $j^* \in \mathcal{M}$ such that $b^{j^*} > -1/(k - 1)$. Then the KKT conditions of (17) and (18) imply that $\alpha^{j^*}_i = 1$ for $i \notin I_{j^*}$ since $\xi^{j^*}_i = (b^{j^*} - y_i^{j^*})_+ > 0$. However, by (16) $\sum_i \alpha^{j^*}_i$ should be the same for all $j$, which implies that $\alpha^{j^*}_i = 1$ for other $j \in \mathcal{M}$ and $i \notin I_j$ to have the same sum of $n - n_{\mathcal{M}}$. This proves (22) and (23), in particular, and the rest follows immediately. $\square$

Remark 2 The value of (5) for the initial $b$ in Lemma 2 is

$$\sum_{j=1}^{k} (L^j)^t \xi^j = (n - n_{\mathcal{M}})k/(k - 1),$$

except the multiplicative constant $1/n$. In fact, Lemma 1 is a special case of Lemma 2. Each $b^j$ with $j \in \mathcal{M}$ can be chosen to be $(k/|\mathcal{M}| - 1)/(k - 1)$ for computational ease. If $k$ classes are completely balanced, that is, $\mathcal{M} = \{1, \ldots, k\}$, then $\alpha^j_i = 1$ for each $j$ and $i \notin I_j$. Again, Lemma 2 is a generalized version of Lemma 1 for the balanced binary case in Hastie et al. (2004).
We start from $\lambda$ sufficiently large but indefinite, which determines the limit MSVM solution in the foregoing two lemmas, and decrease $\lambda$ until non-trivial solutions emerge. To find such genuine starting value of $\lambda$ and the corresponding $b$, we consider two possible cases of the limit solution. First, $\alpha_i^j \in \{0, 1\}$ for all $j$ and $i \notin I_j$. For the completely balanced situation, that is the case. Unbalanced class proportions may also lead to the first case. As $\lambda$ decreases, $\alpha$ changes, however, with the restriction that the equality constraint (16) is always satisfied. Since (16) states that the sum of the Lagrange multipliers $\alpha_i^j$ is the same across all $j = 1, \ldots, k$, any change in $\alpha$, which is bound to reduce $\alpha_i^j$ with $j \in M$ from 1, would reduce some $\alpha_i^j = 1$ with $j \notin M$. Hence, some $k$ indices of $\alpha_i^j$, one from each $j$, should enter the elbow set $E$ simultaneously. Letting $B^j = \{i | \alpha_i^j = 1\}$ for each $j$, we choose the data index $i^j_* = \arg\min_{i \in B^j} H_i^j$, where $H_i^j = -\sum_{r=1}^n (\alpha_i^j - \bar{\alpha}_r)K(x_r, x_i)$. These $k$ indices are chosen to satisfy $f^j(x_{i_*}) = -1/(k-1)$ yielding $k$ equations that determine the initial $\lambda$ and $b$.

Second, there could be two or more $0 < \alpha_i^j < 1$ for some $j$ by (21) and (23). In this case, $\alpha_i^j$ strictly between 0 and 1 will remain on the elbow set until other components $j$ without such index $i$ have a point reaching the margin $f_i^j = y_i^j$. So, if there is $l$ such that $0 < \alpha_l^j < 1$ for $j$, we define $i^j_* = l$. Otherwise, $i^j_* = \arg\min_{i \in B^j} H_i^j$. Again, $f^j(x_{i_*}) = -1/(k-1)$ gives a set of $k$ equations as follows. For $j = 1, \ldots, k$,

$$b^j - \frac{1}{n\lambda} \sum_{r=1}^n (\alpha_r^j - \bar{\alpha}_r)K(x_r, x_{i_*}) = -\frac{1}{k-1} \quad \text{and} \quad \sum_{j=1}^k b^j = 0.$$

Solving the equations, we have the initial $\lambda$ and $b$ in both of the scenarios:

$$\lambda = \frac{k-1}{kn} \sum_{j=1}^k \sum_{r=1}^n (\alpha_r^j - \bar{\alpha}_r)K(x_r, x_{i_*}) \quad \text{and} \quad (24)$$

$$b^j = -\frac{1}{k-1} + \frac{1}{n\lambda} \sum_{r=1}^n (\alpha_r^j - \bar{\alpha}_r)K(x_r, x_{i_*}). \quad (25)$$

### 3.2 Characterizing Coefficient Paths

As seen in the previous initialization step, the elbow set permits explicit equations of $f_i^j = y_i^j$ for $(i, j) \in E$. Accordingly, it allows us to find some of the Lagrange multipliers fully, which determine the coefficients in (2). As a result, our strategy for constructing the coefficient paths is to keep track of changes in the elbow set. There are three types of different events which can change the elbow set.
1. An index \((i, j)\) leaves from \(E\) to join either \(L\) or \(U\).

2. An index \((i, j)\) from \(L\) enters \(E\).

3. An index \((i, j)\) from \(U\) enters \(E\).

Continuity of the objective function (14) in \(\lambda\) implies that the minimizer \(\alpha\) as a function of \(\lambda\) changes continuously between the consecutive values of \(\lambda\) at which one of the above three events occurs, and so does the MSVM solution. Consider \(\{\lambda_\ell, \ell = 0, 1, \ldots\}\), a decreasing sequence of \(\lambda\) starting from the initial value in (24) and indicating the values at which some change occurs in \(E\). In fact, the sequence determines the break points of \(\lambda\) at which \(\alpha\) can be completely characterized. For \(\lambda_\ell\), denote the corresponding elbow, upper and lower sets by \(E_\ell, U_\ell,\) and \(L_\ell\) respectively. Letting \(\alpha_0^j = n\lambda b^j\), we write

\[
\hat{f}^j_\lambda(x) = \frac{1}{n\lambda} \left( -\sum_{i=1}^n (\alpha_i^j - \bar{\alpha}_i) K(x_i, x) + \alpha_0^j \right).
\]

For \(\lambda_{\ell+1} < \lambda < \lambda_\ell\), we can express \(\hat{f}^j_\lambda(x)\) in terms of an incremental change from \(\hat{f}^j_{\lambda_\ell}(x)\). Denoting \(\alpha_i^j\) at \(\lambda_\ell\) by \(\alpha_i^{j(\ell)}\),

\[
\hat{f}^j_\lambda(x) = \left[ \hat{f}^j_{\lambda_\ell}(x) - \frac{\lambda_\ell}{\lambda} \hat{f}^j_{\lambda_\ell}(x) \right] + \frac{\lambda_\ell}{\lambda} \hat{f}^j_{\lambda_\ell}(x)
\]

\[
= \frac{1}{n\lambda} \left[ -\sum_{i=1}^n ((\alpha_i^j - \alpha_i^{j(\ell)}) - (\bar{\alpha}_i - \bar{\alpha}_i^{(\ell)})) K(x_i, x) + (\alpha_0^j - \alpha^{j(\ell)}_0) + n\lambda_\ell \hat{f}^j_{\lambda_\ell}(x) \right]
\]

\[
= \frac{1}{n\lambda} \left[ -\sum_{i \in \cup_{j} \mathcal{E}^j_\ell} ((\alpha_i^j - \alpha_i^{j(\ell)}) - (\bar{\alpha}_i - \bar{\alpha}_i^{(\ell)})) K(x_i, x) + (\alpha_0^j - \alpha^{j(\ell)}_0) + n\lambda_\ell \hat{f}^j_{\lambda_\ell}(x) \right],
\]

where \(\mathcal{E}^j_\ell = \{i| (i, j) \in \mathcal{E}_\ell \}\) for each \(j\). The last equality holds because \(\alpha_i^j = 0\) or 1 for all \(j\) without any change from \(\alpha_i^{j(\ell)}\) if \(i \notin \cup_{j} \mathcal{E}^j_\ell\). For all \((i, j) \in \mathcal{E}_\ell, \hat{f}^j_{\lambda}(x_i) = y_i^j\). Letting \(\delta_0^j = \alpha_0^j - \alpha^{j(\ell)}_0, \delta_i^j = -\alpha_i^j + \alpha_i^{j(\ell)}\), \(\bar{\delta}_i = -(\bar{\alpha}_i - \bar{\alpha}_i^{(\ell)})\) for \(i \geq 1\), and \(\mathcal{E}^T_\ell = \cup_{j} \mathcal{E}^j_\ell\), we have

\[
\hat{f}^j_{\lambda}(x_i) = \frac{1}{n\lambda} \left[ \sum_{r \in \mathcal{E}^T_\ell} (\delta_i^j - \bar{\delta}_r) K(x_r, x_i) + \delta_0^j - \frac{n\lambda_\ell}{k-1} \right] = -\frac{1}{k-1},
\]

which gives

\[
\sum_{r \in \mathcal{E}^T_\ell} (\delta_i^j - \bar{\delta}_r) K(x_r, x_i) + \delta_0^j = \frac{n(\lambda_\ell - \lambda)}{k-1}.
\]

Given any \(\lambda\), \(\sum_{i=1}^n \alpha_i^j\) should be the same for all \(j\) and \(\sum_{j=1}^k b^j = 0\) by the sum-to-zero constraint. This yields \(\sum_{i \in \mathcal{E}^j_\ell} \delta_i^j = \cdots = \sum_{i \in \mathcal{E}^j_\ell} \delta_i^k\) and \(\sum_{j=1}^k \delta_0^j = 0\). As a result, these \(k\)
constraints and (26) provide a set of \(|\mathcal{E}_\ell| + k\) equations to solve for \(|\mathcal{E}_\ell| + k\) unknowns if all the \(k\) elbow sets \(\mathcal{E}_\ell^j\) are non-empty. To re-express (26) conveniently in a vector notation, let \(m_j = |\mathcal{E}_\ell^j|\) and \(i_1^j, \ldots, i_{m_j}^j\) denote the \(m_j\) data indices in \(\mathcal{E}_\ell^j\). Now, we define \(\delta_0 = (\delta_0^1, \ldots, \delta_0^k)^t\) and \(\delta = (\delta^1, \ldots, \delta^k)^t\) with \(\delta^j = (\delta_{i_1}^j, \ldots, \delta_{i_{m_j}}^j)\). Note that (26) depends only on \(\delta_0\) and \(\delta\) since \(\delta^j_i = 0\) for all \((i, j) \notin \mathcal{E}_\ell\). \(K_\ell^* = [K_\ell^*_{ij}]\) is the square block matrix of \(|\mathcal{E}_\ell| = m_1 + \cdots + m_k\) rows and columns, whose \(lj\)th block \((l, j = 1, \ldots, k)\) is given by

\[
K_{lj}^* = \left( I(l = j) - \frac{1}{k} \right) \begin{pmatrix}
K(x_{i_1^l}, x_{i_1^l}^l) & \cdots & K(x_{i_{m_j}^j}, x_{i_{m_j}^j}^l)
\vdots & \ddots & \vdots
K(x_{i_1^j}, x_{i_{m_j}}^j) & \cdots & K(x_{i_{m_j}}^j, x_{i_{m_j}}^j)
\end{pmatrix}
\]

Also, define

\[
1_\delta = \begin{pmatrix}
\epsilon_{m_1}^t & -\epsilon_{m_2}^t & 0 & \ldots & 0 \\
0 & \epsilon_{m_2}^t & -\epsilon_{m_3}^t & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & \epsilon_{m_{k-1}}^t & -\epsilon_{m_k}^t
\end{pmatrix}
\quad \text{and} \quad
1_0 = \begin{pmatrix}
\epsilon_{m_1} & 0 & \ldots & 0 \\
0 & \epsilon_{m_2} & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & \epsilon_{m_k}
\end{pmatrix}
\]

where \(\epsilon_m\) is the vector of \(m\) ones and \(0\) indicates a vector of appropriately defined number of zeros. Then, (26) and the constraints are succinctly expressed as

\[
\begin{pmatrix}
0 & e_{k-1}^l \\
K_\ell^* & 1_0 \\
1_\delta & 0
\end{pmatrix}
\begin{pmatrix}
\delta \\
\delta_0
\end{pmatrix}
= \begin{pmatrix}
0 \\
\frac{n(\lambda_{\ell} - \lambda)}{k-1} e_{|\mathcal{E}_\ell|} \\\n0
\end{pmatrix}
\]

(27)

Letting \(A_\ell\) be the square matrix in the left-hand side of (27) and \(v_\ell^t = (0, e_{|\mathcal{E}_\ell|}^t, 0)\), we solve for \(\delta_0\) and \(\delta\). If \(A_\ell\) is invertible,

\[
\begin{pmatrix}
\delta \\
\delta_0
\end{pmatrix}
= \frac{n(\lambda_{\ell} - \lambda)}{k-1} A_\ell^{-1} v_\ell.
\]

Abbreviating \(A_\ell^{-1} v_\ell\) by \(w_\ell\), we have

\[
\alpha_0 - \alpha_{0(\ell)} = n(\lambda_{\ell} - \lambda)/(k-1) w_0 \quad \text{and} \quad
\alpha_i^j - \alpha_{i(\ell)}^j = n(\lambda - \lambda_{\ell})/(k-1) w_i^j \quad \text{for } (i, j) \in \mathcal{E}_\ell,
\]

(28)

where \(w_0\) is the last \(k\) elements of \(w_\ell\) and \(w_i^j\) is the element of \(w_\ell\) corresponding to \(\delta_i^j\). This shows that the scaled intercepts and the Lagrange multipliers in the elbow set change
linearly in $\lambda$ on the interval $(\lambda_{t+1}, \lambda_{t})$. Rescaling them properly by $n\lambda$ to obtain equations for the coefficients, we have

$$b = \frac{\lambda_t}{\lambda}(b_{e} + \frac{1}{k-1}w_{0}) - \frac{1}{k-1}w_{0} \text{ and }$$

$$c_{i}^{j} = \frac{\lambda_t}{\lambda}(c_{i}^{j} + \frac{w_{i}^{j} - \bar{w}_{i}}{k-1}) - \frac{w_{i}^{j} - \bar{w}_{i}}{k-1}.$$  

(29)

(30)

$\bar{w}_{i} = (1/k)\sum_{j} w_{i}^{j}$ and the summation is only over $j$'s with $(i, j) \in \mathcal{E}_{t}$. This proves the following theorem of piecewise linearity of the paths of the coefficients $b$ and $C$.

**Theorem 1** If there is at least one data index in the elbow set $\mathcal{E}_{t}$ at $\lambda_{t}$ for each $j$, then the coefficient path of the MSVM is linear in $1/\lambda$ on the interval $(\lambda_{t+1}, \lambda_{t})$.

Likewise, the $j$th coordinate of the MSVM output has a path linear in $1/\lambda$ as follows.

$$\hat{f}_{\lambda}^{j}(x) = \frac{\lambda_t}{\lambda}(\hat{f}_{\lambda}^{j}(x) - \hat{g}_{\lambda}^{j}(x)) + \hat{g}_{\lambda}^{j}(x),$$

(31)

where $\hat{g}_{\lambda}^{j}(x) = -\left(\sum_{i \in \mathcal{E}_{t}}(w_{i}^{j} - \bar{w}_{i})K(x, x_{i}) + w_{0}^{j}\right)/(k-1)$. $\hat{g}_{\lambda}^{j}(x)$ is pivotal to $\hat{f}_{\lambda}^{j}(x)$ in the sense that $\hat{f}_{\lambda}^{j}(x)$ can be expressed as a scaled $\hat{g}_{\lambda}^{j}(x)$ once both are pivoted on $\hat{g}_{\lambda}^{j}(x)$.

So far, we have discussed how the MSVM solution path is explicitly characterized as a function of $\lambda$ when the elbow set $\mathcal{E}_{t}^{j}$ for each $j$ is not empty. If there is at least one empty elbow set $\mathcal{E}_{t}^{j}$ at $\lambda = \lambda_{t}$, then the constraints used in the previous characterization needs to be modified. The constraint that the sum of $\alpha_{i}^{j}$ should stay the same for all $j$, now becomes $\sum_{i \in \mathcal{E}_{t}} \delta_{i}^{j} = 0$ for each non-empty elbow set $\mathcal{E}_{t}^{j}$. We eliminate the component(s) corresponding to the empty $\mathcal{E}_{t}^{j}$ from $\delta_{0}$ and the corresponding column(s) from $1_{0}$, and denote the resulting vector and matrix by $\delta_{0}^{*}$ and $1_{0}^{*}$, respectively. Then (27) is adjusted as follows by taking into account the presence of some empty elbow set(s).

$$
\left(\begin{array}{cc}
K_{t}^{*} & 1_{0}^{*} \\
1_{0}^{*} & 0
\end{array}\right)
\left(\begin{array}{c}
\delta \\
\delta_{0}^{*}
\end{array}\right) = \frac{n(\lambda_{t}-\lambda)}{k-1}e_{|\mathcal{E}_{t}|}. $$

(32)

Assuming that $K_{t}^{*}$ is of full rank, we solve for $\delta$ and $\delta_{0}^{*}$. Due to the simple structure of the block matrix in the left-hand side, its inverse can be easily written out to give the explicit solution of $\delta = 0$ and $\delta_{0}^{*} = n(\lambda_{t}-\lambda)/(k-1)e_{k^{*}}$, where $k^{*}$ is the number of non-empty elbow set $\mathcal{E}_{t}^{j}$. This yields

$$c_{i}^{j} = \frac{\lambda_t}{\lambda}c_{i}^{j}(\ell) \text{ for } (i, j) \in \mathcal{E}_{t} \text{ and }$$

$$b^{j} = \frac{\lambda_t}{\lambda}(b_{e}^{j} + \frac{1}{k-1}) - \frac{1}{k-1} \text{ for non-empty } \mathcal{E}_{t}^{j}.$$
When there are more than one empty \( E_j^\ell \), \( \delta_j^\ell \)'s for the empty sets are not uniquely determined other than that they are constrained to satisfy \( \sum_j \delta_j^0 = 0 \). In practice, a linear path can be chosen for such \( b^j \) corresponding to the empty elbow sets for convenience. Theorem 1 and the following theorem of summary complete the conclusion that the solution path of the MSVM is piecewise linear.

**Theorem 2** If there is only one empty elbow set \( E_j^\ell \) at \( \lambda_\ell \), then the coefficient path of the MSVM is linear in \( 1/\lambda \) on the interval \( (\lambda_{\ell+1}, \lambda_\ell) \). If there are more than one empty elbow sets at \( \lambda_\ell \), then the coefficient path is still linear in \( 1/\lambda \) except that the path of \( b^j \) for the empty \( E_j^\ell \) can be arbitrary.

## 4 Computation

### 4.1 Generating the break points

This section regards how to generate the decreasing sequence of \( \{\lambda_\ell, \ell = 0, 1, \ldots\} \) that determines the joints of the piecewise linear MSVM solution path. Given \( \lambda_\ell \), we find \( \lambda_{\ell+1} \) by considering the following possible events.

1. An index \((i, j)\) in \( E_\ell \) leaves the elbow set, and \( \alpha_j^i \) (0 \( \leq \) \( \alpha_j^i \) \( \leq \) 1) becomes either 0 or 1.
2. An index \((i, j)\) in \( L_\ell \) or \( U_\ell \) joins the elbow set, and \( \hat{f}^j(x_i) \) becomes equal to \( y_j^i \).

When the first type of event happens, a candidate \( \lambda_{\ell+1} \) is obtained by setting \( \alpha_j^i \) in the left-hand side of (28) to 0 or 1. For the second type of event, we set the left-hand side of (31) to \( y_j^i \), which proposes another tentative \( \lambda_{\ell+1} \) given by

\[
\lambda_{\ell+1} = \frac{\lambda_\ell (\hat{f}_{x_i}^j(x_i) - \hat{g}_{x_i}^j(x_i))}{y_j^i - \hat{g}_{x_i}^j(x_i)}.
\]

The next break point \( \lambda_{\ell+1} \) is then determined by the largest \( \lambda < \lambda_\ell \) among the potential candidate values.

If there is at least one empty elbow set \( E_j^\ell \) at \( \lambda_\ell \), then \( \alpha_j^i \) in non-empty sets stays the same until the next event occurs as discussed in Section 3. The ensuing event that changes the elbow set is of the second type in this case, and specifically it is the event that a point for each \( j \) with empty \( E_j^\ell \) hits the margin \( \hat{f}^j(x_i) = y_j^i \) simultaneously. Such a point for each empty elbow set \( E_j^\ell \) is determined via the same argument as in the initialization process. Thus, the corresponding \( \lambda_{\ell+1} \) is identified by (24).
4.2 Stopping rule

We stop finding the next $\lambda_{\ell+1}$ and tracing out the solution path if the upper set $U_\ell$ is empty, since this implies that the empirical risk functional at $\lambda_\ell$ is driven to zero completely overfitting the data. As $\lambda$ gets smaller, the upper sets are bound to become empty for separable problems. For non separable problems, monitoring change in the solutions at two consecutive break points would admit a more time-saving stopping criterion. Inspection of (31) provides a rule of stopping at $\lambda_\ell$ if $\max_j (1/n) \sum_{i=1}^n |f_{\lambda_\ell}^j(x_i) - g_{\lambda_\ell}^j(x_i)| < \epsilon$, where $\epsilon$ denotes a prespecified tolerance to declare no change between successive solutions.

From a practical point of view, early stopping may be desired if one does not attempt to find the entire solution path, but wishes to keep track of solutions only until $\lambda$ gets small enough to be in the vicinity of theoretically optimal values of the least error rate. As a related issue, it is worth noting that the computational cost of characterizing the MSVM solution path essentially lies in solving a system of equations with at most $|E_\ell| + k$ unknowns at each $\lambda_\ell$. As $\lambda$ decreases, the cardinality of the elbow set $E_\ell$ tends to increase, an example of which is to be shown shortly. This is another motivation for devising an early stopping rule. The idea of early stopping presupposes a reasonable data-driven measure of predictive accuracy of the solution at $\lambda_\ell$. Such measure will guide us to judge whether the optimal value of the regularization parameter has been already attained or not. If attained, then we may stop without completing the entire path.

5 A Numerical Example

To illustrate the algorithm of characterizing the MSVM solution path, we consider a simulated three-class example. The simulation setting is as follows. For class 1, two covariates $X = (X_1, X_2)$ are generated from a normal distribution with mean $(3,0)$. For class 2, $X$ comes from a mixture of two normal distributions with mean $(0,3)$ and $(1,1)$, respectively. The mixing proportion is 0.5. For class 3, $X$ has a normal distribution with mean $(-1,1)$. For all the three classes, $X_1$ and $X_2$ are independent and have variance 1. A training data set of size $n = 300$ was generated from the specified distributions with $n_j = 100$ for each class. The left panel of Figure 2 depicts the training data denoted by circles (class 1 in green, 2 in magenta, and 3 in blue) in a scatter plot as well as the theoretically optimal classification boundaries. A Monte Carlo estimate of the Bayes error rate of this example was approximately 0.1773 with standard error 0.007 based on a test data set of size 3,000 with 1,000 cases from each class. The MSVM with Gaussian kernel of $K(s, t) = \exp(-\gamma \|s - t\|^2)$ was
applied to the simulated data, and the entire solution path was traced. Here the additional parameter $\gamma$ was fixed at 1.

Figure 2: Left: the boundaries of the Bayes classification rule in a scatter plot of the training data. Right: the classification boundaries determined by the MSVM at an appropriately chosen $\lambda$. Class 1: green, class 2: magenta, and class 3: blue.

Figure 3 shows how the test error rates, when evaluated over the test set, change as the regularization parameter $\lambda$ varies. Notice that the $x$-axis is $\log(1/\lambda)$, so $\lambda$ decreases in the positive direction of $x$. As $\lambda$ decreases from the initial value, the test error rates get smaller, reach the minimum around $\log(\lambda) = -6$, and begin to increase soon after that, clearly demonstrating that the choice of $\lambda$ is critical to the performance of the solution. We note here that for better illustration, the test error rate curve in the figure was cropped to show part of the full path over the whole range of $\lambda$ since the test error rates rise up sharply as $\lambda$ approaches the smaller end. The minimum error rate achieved by the MSVM classifier was about 0.1743 at $\log(\lambda) \approx -5.886$. This illustrates that the MSVM equipped with a flexible kernel could achieve the theoretically optimal accuracy if $\lambda$ were chosen appropriately. The right panel of Figure 2 shows the classification boundaries induced by the MSVM solution at this optimal $\lambda$.

In practice, we have to face the problem of choosing the regularization parameter in the absence of test cases. Data-driven tuning is in itself a long standing research topic, complete treatment of which even in this specific context of the MSVM would require a separate venue.
Figure 3: Test error rate curve as a function of $\lambda$.

We will not delve into the issue in this paper. General approaches to tuning or equivalently model selection can be found in, for example, Efron (1986) and Wahba (1990). Just for illustration of feasibility of tuning, 100 new data sets of size 300 were generated as random replicates of a tuning set, and $\lambda$ was chosen to minimize the error rate over each tuning set. Figure 4 shows an estimated density curve of such $\hat{\lambda}$’s. Bimodality of the distribution is discernible perhaps due to the presence of multiple dips in the test error rate plot of Figure 3. The mean of the test error rates at tuned $\hat{\lambda}$’s values was 0.1815 with standard deviation of 0.00525. The mean is slightly larger than the Bayes error rate of about 0.1773, but reasonably close to it.

The length of the solution path we monitor or simply the number of break points of $\lambda$ depends on the stopping criterion used. The rule adopted in our current implementation of the algorithm does not consider the two stopping criteria discussed in Section 4 only, but also a preset lower bound of $\lambda$ and the maximum number of break points, which could be rather arbitrary but nonetheless useful in practice. Whichever criterion is met first, the algorithm stops. Based on the current working rule, the preset lower bound of $\lambda$ was reached first before any other criteria for this example and there were 3014 breaking points of $\lambda$. Availability of the entire solution path allows us to visualize the effect of the regularization parameter on the MSVM solution at various angles. For example, Figure 5 depicts the complete paths of $\hat{f}^1_\lambda(x_i)$, $\hat{f}^2_\lambda(x_i)$, and $\hat{f}^3_\lambda(x_i)$ for the isolated instance from class 3 (blue) in the top left region. The paths start off the initial solution of (0, 0, 0), begin to diverge for better fit to the data as $\lambda$ decreases, returning class 2 as the predicted class for moderate values of $\lambda$. As $\lambda$ further
Figure 4: An estimated density function of the optimal parameters $\hat{\lambda}$ chosen by 100 tuning sets of sample size 300.

decreases, the solution paths tend to follow the data too closely and $\hat{f}_i^3(x_i)$ emerges to be the maximum component. This is a snapshot of the spectrum of solutions ranging from the least to the complete fit to the data controlled by $\lambda$.

As mentioned before, the computational complexity of the proposed algorithm depends on the size of elbow set at each $\lambda_\ell$, and the elbow size is bound to increase as $\lambda$ gets smaller. Figure 5 shows how the size of elbow set changes as a function $\lambda$ for each class. The median sizes of elbow sets were 39, 30, and 30 for class 1, 2, and 3, respectively, while the maximum elbow sizes were 68, 51, and 61. Around the optimal value of $\lambda$, roughly 40 to 50% of data points were in the elbow set for this example. It is interesting to observe that the rate of increase in the size of each elbow set $E_\ell^j$ is almost constant until $\lambda$ reaches the optimal value around $\exp(-6)$, and it becomes smaller soon after that and remains almost constant. If possible, it is desirable to have theoretical explanation for this, which may help devise an early stopping rule based on this observation.

6 Discussion

It is conceptually attractive to have the whole solution path in perspective, and computationally efficient to construct the path sequentially with the basic operation of solving a
Figure 5: The entire paths of $\hat{f}_1^\lambda(x_i)$, $\hat{f}_2^\lambda(x_i)$, and $\hat{f}_3^\lambda(x_i)$ for an outlying instance $x_i$ from class 3. The circles correspond to $\lambda$ with the minimum test error rate.

Figure 6: The size of elbow set $\mathcal{E}_i^j$ for three classes as a function $\lambda$. 

18
system of linear equations. Another type of a regularization problem, which may well benefit from the idea is feature selection via $l_1$ type penalty imposed on rescaling factors of features, in addition to the squared norm penalty for the maximal geometric margin. For example, such sparse solution approach to feature selection for the MSVM is described in Lee, Kim, Lee and Koo (2004) by using functional ANOVA decomposition. Besides $\lambda$, there is another regularization parameter $\lambda_\theta$ in the setting that governs the sum of non negative weights $\theta_\nu$ assigned to each feature $x_\nu$, which is closely tied to the model complexity of the number of features present in each model. In this case, the entire spectrum of models embraces the simplest model of a constant function to the least regularized one with all the features regardless of their relevance to prediction of $y$. It is particularly enticing to overlook how the features get in and out of the current model along the path in conjunction with the prediction accuracy of the the current model. This complete picture of model path would enable us to understand the data better, and especially help assess the multiplicity of good descriptions of sparse data. We remark here that this approach is closely related to the least angle regression proposed by Efron et al. (2004) for variable selection in multiple linear regression, despite the notable difference between the two settings.

Another direction of interest is investigation of model selection issues that generation of the solution path entails. As mentioned before, it would be ideal to incorporate a reliable estimate of the performance of the current solution over unseen data with the fitting algorithm, and instantly evaluate the generalization ability of each solution at hand as the path evolves. Monitoring such measure would facilitate early stopping if desired. Given a solution, a model selection criterion with little added computational cost would be much preferable in this case. Related future directions that we pursue include comparison of a variety of old and new model selection criteria, design of working rules for early stopping without sacrificing the optimal performance, approximation of solutions by basis thinning, and further streamlining of the fitting process. For careful examination of these issues, a comprehensive numerical study is to be carried out, completion of which would result in a useful and effective data analysis tool for practitioners.

References


