Statistica Sinica **16**(2006), 391-409

CHARACTERIZING THE SOLUTION PATH OF MULTICATEGORY SUPPORT VECTOR MACHINES

Yoonkyung Lee and Zhenhuan Cui

The Ohio State University

Abstract: An algorithm for 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) of Efron, Hastie, Johnstone and Tibshirani (2004) and the SVM path of Hastie, Rosset, Tibshirani and Zhu (2004) showcase recent developments of computational algorithms to characterize the entire regularization path in place of a user-dependent grid search for a good solution. These constructive algorithms 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. Different from this approach, there are quite a few widely used algorithms for the SVM such as SMO (Sequential Minimal Optimization) (Platt (1999)), SVM *light* (Joachims (1999)), and LIBSVM (Hsu and Lin (2002)). However, these 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 λ. 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 an 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 λ 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 closely follow the notation and terminology 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 $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, n$. $\boldsymbol{x}_i \in R^p$ represents the covariates of the ith observation and the response $y_i \in \{1, \ldots, k\}$ denotes the class that it falls into. In general, a classification rule $\phi(\boldsymbol{x}) : \mathbb{R}^p \to \{1, \ldots, k\}$ is constructed, based on the training data, that generalizes the relationship between $\boldsymbol{x}_i$ and the class label y_i.

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

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda \|h\|^2.$$

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

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{L}(\boldsymbol{y}_i)^t (\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i)_+ + \frac{\lambda}{2}\sum_{j=1}^{k} \|h^j\|^2, \tag{2.1}$$

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

of the form

$$\hat{f}_\lambda^j(\boldsymbol{x}) = b^j + \sum_{i=1}^n c_i^j K(\boldsymbol{x}_i, \boldsymbol{x}) \quad \text{for } j = 1, \ldots, k, \tag{2.2}$$

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

The main focus of this paper is how to explicitly characterize the coefficient paths of b^j and c_i^j of the solution as a function of λ. For expositions to follow, we briefly discuss the optimization problems associated with (2.1). Let the coefficient vector $\boldsymbol{c}^j = (c_1^j, \ldots, c_n^j)^t$ for $j = 1, \ldots, k$, $\boldsymbol{b} = (b^1, \ldots, b^k)^t$, and $\boldsymbol{C} = (\boldsymbol{c}^1, \ldots, \boldsymbol{c}^k)$. With some abuse of notation, let bold-faced $\boldsymbol{K}$ stand for an n by n matrix with the lmth entry $K(\boldsymbol{x}_l, \boldsymbol{x}_m)$. Also, let $\boldsymbol{L}^j$ denote the jth coordinates of the n misclassification cost vectors, $(L^j_{cat(1)}, \ldots, L^j_{cat(n)})^t$ and $\boldsymbol{y}^j = (y_1^j, \ldots, y_n^j)^t$. Then the MSVM in (2.1) can be rewritten as the problem of finding $(\boldsymbol{b}, \boldsymbol{C})$ to minimize

$$L_P(\boldsymbol{b}, \boldsymbol{C}) = \frac{1}{n} \sum_{j=1}^k (\boldsymbol{L}^j)^t \Big(b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j - \boldsymbol{y}^j \Big)_+ + \frac{\lambda}{2} \sum_{j=1}^k (\boldsymbol{c}^j)^t \boldsymbol{K} \boldsymbol{c}^j \tag{2.3}$$

$$\text{subject to } \textstyle\sum_{j=1}^k \Big(b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j \Big) = \boldsymbol{0}, \tag{2.4}$$

where $\boldsymbol{e}$ is the vector of n ones. To handle the truncate function $(x)_+$ in (2.3), we introduce nonnegative slack variables denoted by $\boldsymbol{\xi}^j = (\xi_1^j, \ldots, \xi_n^j)^t$ for $j = 1, \ldots, k$. Let $\boldsymbol{\xi} = (\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^k)$. By using the slack variables, (2.3) and (2.4) can be reformulated as finding $\boldsymbol{b}$, $\boldsymbol{C}$, and $\boldsymbol{\xi}$ that minimize

$$L_P(\boldsymbol{b}, \boldsymbol{C}, \boldsymbol{\xi}) = \frac{1}{n} \sum_{j=1}^k (\boldsymbol{L}^j)^t \boldsymbol{\xi}^j + \frac{\lambda}{2} \sum_{j=1}^k (\boldsymbol{c}^j)^t \boldsymbol{K} \boldsymbol{c}^j \tag{2.5}$$

$$\text{subject to } b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j - \boldsymbol{y}^j \le \boldsymbol{\xi}^j, \text{ for } j = 1, \ldots, k, \tag{2.6}$$

$$\boldsymbol{\xi}^j \ge \boldsymbol{0}, \text{ for } j = 1, \ldots, k, \text{ and} \tag{2.7}$$

$$\sum_{j=1}^k \Big(b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j \Big) = \boldsymbol{0}. \tag{2.8}$$

For the Lagrangian dual formulation of the problem, we introduce nonnegative Lagrange multipliers $\boldsymbol{\alpha}^j = (\alpha_1^j, \ldots, \alpha_n^j)^t \in R^n$ for (2.6), $\boldsymbol{\gamma}^j \in R^n$ for (2.7), and unconstrained multipliers $\boldsymbol{\delta}_f \in R^n$ for (2.8). Then the dual problem becomes

$$\begin{aligned} \max L_D = & \sum_{j=1}^k (\boldsymbol{L}^j)^t \boldsymbol{\xi}^j + \frac{n\lambda}{2} \sum_{j=1}^k (\boldsymbol{c}^j)^t \boldsymbol{K} \boldsymbol{c}^j + \sum_{j=1}^k (\boldsymbol{\alpha}^j)^t (b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j - \boldsymbol{y}^j - \boldsymbol{\xi}^j) \\ & - \sum_{j=1}^k (\boldsymbol{\gamma}^j)^t \boldsymbol{\xi}^j + \boldsymbol{\delta}_f^t \sum_{j=1}^k \Big(b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j \Big) \end{aligned}$$

$$\text{subject to, for } j = 1, \ldots, k, \quad \frac{\partial L_D}{\partial \boldsymbol{\xi}^j} = \boldsymbol{L}_j - \boldsymbol{\alpha}^j - \boldsymbol{\gamma}^j = \mathbf{0}, \tag{2.9}$$

$$\frac{\partial L_D}{\partial \boldsymbol{c}^j} = n\lambda \boldsymbol{K}\boldsymbol{c}^j + \boldsymbol{K}\boldsymbol{\alpha}^j + \boldsymbol{K}\boldsymbol{\delta}_f = \mathbf{0}, \tag{2.10}$$

$$\frac{\partial L_D}{\partial b^j} = (\boldsymbol{\alpha}^j + \boldsymbol{\delta}_f)^t \boldsymbol{e} = \mathbf{0}, \tag{2.11}$$

$$\boldsymbol{\alpha}^j \geq \mathbf{0} \text{ and } \boldsymbol{\gamma}^j \geq \mathbf{0}.$$

Letting $\bar{\boldsymbol{\alpha}} = (\sum_{j=1}^k \boldsymbol{\alpha}^j)/k$, we have $(\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})^t \boldsymbol{e} = \mathbf{0}$ by taking the unconstrained $\boldsymbol{\delta}_f = -\bar{\boldsymbol{\alpha}}$ in (2.11) and $\boldsymbol{c}^j = -(\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})/(n\lambda)$ from (2.10). Using these relations and (2.9), and denoting $(\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^k)$ by $\boldsymbol{\alpha}$, we have the dual problem of

$$\min L_D(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{j=1}^k (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})^t \boldsymbol{K} (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}}) + n\lambda \sum_{j=1}^k (\boldsymbol{\alpha}^j)^t \boldsymbol{y}^j \tag{2.12}$$

$$\text{subject to} \quad \mathbf{0} \leq \boldsymbol{\alpha}^j \leq \boldsymbol{L}^j \quad \text{for } j = 1, \ldots, k, \tag{2.13}$$

$$(\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})^t \boldsymbol{e} = \mathbf{0} \ \text{ for } j = 1, \ldots, k. \tag{2.14}$$

Note that the α_i^j's corresponding to zero $L_{cat(i)}^j$'s are trivially zero, so the above dual problem involves only $n(k-1)$ Lagrange multipliers. Throughout this paper, we consider only the $n(k-1)$ non-trivial α_i^j. By the Karush-Kuhn-Tucker (KKT) complementarity conditions, the solution satisfies

$$\boldsymbol{\alpha}^j \perp (b^j \boldsymbol{e} + \boldsymbol{K}\boldsymbol{c}^j - \boldsymbol{y}^j - \boldsymbol{\xi}^j) \ \text{ for } j = 1, \ldots, k, \tag{2.15}$$

$$\boldsymbol{\gamma}^j = (\boldsymbol{L}^j - \boldsymbol{\alpha}^j) \perp \boldsymbol{\xi}^j \qquad \text{for } j = 1, \ldots, k, \tag{2.16}$$

where $\perp$ indicates that componentwise product of two vectors is zero. For instance, if $0 < \alpha_i^j < L_i^j$ for some i, then ξ_i^j should be zero from (2.16), and this implies $b^j + \sum_{l=1}^n c_l^j K(\boldsymbol{x}_l, \boldsymbol{x}_i) - y_i^j = 0$ from (2.15). The KKT conditions categorize each component of $\hat{\boldsymbol{f}}_\lambda(\boldsymbol{x}_i)$ 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 $\hat{f}_\lambda^j(\boldsymbol{x}_i)$ as f_i^j, $\mathcal{E} = \{(i,j) |\ f_i^j - y_i^j = 0,\ \xi_i^j = 0,\ 0 \leq \alpha_i^j \leq L_{cat(i)}^j\}$, an elbow set, $\mathcal{U} = \{(i,j) |\ f_i^j - y_i^j > 0,\ \xi_i^j > 0,\ \alpha_i^j = L_{cat(i)}^j\}$, 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 2.1 depicts the jth 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.

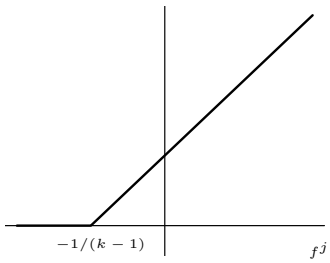

Figure 2.1. MSVM component loss $(f^j - y^j)_+$ where $y^j = -1/(k-1)$.

3. Solution Paths

To describe how MSVM solutions in (2.2) change as a function of the regularization parameter λ, we begin with a very large value of the parameter at which the initial set of dual variables $\boldsymbol{\alpha}^j$ is easily determined. Then a constructive algorithm is laid out for successive update of the dual minimizers as λ decreases. From the relation between the coefficients and the dual variables, $\boldsymbol{c}^j = -(\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})/(n\lambda)$, $\boldsymbol{c}^j = \mathbf{0}$ as λ goes to ∞. So, only b^j and $\boldsymbol{\alpha}^j$ need to be initialized for a sufficiently large λ. For brevity, equal misclassification costs are considered, where $L_j^{j'} = I(j \neq j')$. Generalization of the following to unequal misclassification costs is straightforward.

3.1 Initialization

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

Lemma 1. *Suppose there is only one class j^* with maximum sample size. For a sufficiently large λ, $b^j = 1$ if $j = j^*$, and $-1/(k-1)$ otherwise. $\boldsymbol{\alpha}$ minimizes* $\sum_{j=1}^{k} (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})^t \boldsymbol{K} (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})$ *subject to*

$$0 \leq \alpha_i^j \leq 1 \qquad \text{for } j \neq j^* \text{ and } i \notin \mathcal{I}_j, \tag{3.1}$$

$$\alpha_i^{j^*} = 1 \qquad \text{for } i \notin \mathcal{I}_{j^*}, \tag{3.2}$$

$$\textstyle\sum_i \alpha_i^j = n - n_{j^*} \qquad \text{for } j \neq j^*. \tag{3.3}$$

Proof. For a sufficiently large λ, the minimizer of (2.3) is a constant vector $(b^1, \ldots, b^k)$, for which (2.3) 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 $-1/(k-1)$. Else the objective function can be 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 this is a non-negatively weighted sum of b^j's for $j \neq j^*$, the sum becomes 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 (2.12), (2.13) and (2.14) by observing the following three facts. First, $\xi_i^{j^*} = (b^{j^*} - y_i^{j^*})_+ = k/(k-1)$ for $i \notin \mathcal{I}_{j^*}$, thus $\alpha_i^{j^*} = 1$ satisfies the KKT conditions of (2.15) and (2.16). Second, (2.14) is then restated as $\sum_i \alpha_i^j = n - n_{j^*}$ for all j. Third, as a result, $\sum_{j=1}^k (\boldsymbol{\alpha}^j)^t \boldsymbol{y}^j = -1/(k-1) \sum_{j=1}^k \sum_i \alpha_i^j$ is fixed at $-k/(k-1)(n - n_{j^*})$.

Remark 1. The value of the primal objective function (2.5) for the initial $\boldsymbol{b}$ is $(n - n_{j^*})k/(k-1)$ except for 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 is more than one class in* $\mathcal{M} = \operatorname{argmax} n_j$. *For a sufficiently large* λ, $b^j = -1/(k-1)$ *for* $j \notin \mathcal{M}$, *and* $b^j \geq -1/(k-1)$ *for* $j \in \mathcal{M}$ *with* $\sum_{j \in \mathcal{M}} b^j = (k - |\mathcal{M}|)/(k-1)$. $\boldsymbol{\alpha}$ *minimizes* $\displaystyle\sum_{j=1}^k (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})^t \boldsymbol{K} (\boldsymbol{\alpha}^j - \bar{\boldsymbol{\alpha}})$ *subject to*

$$
\begin{aligned}
0 \leq \alpha_i^j \leq 1 \qquad & \text{for } j \notin \mathcal{M} \text{ and } i \notin \mathcal{I}_j, \\
\alpha_i^j = 1 \qquad & \text{for } j \in \mathcal{M} \text{ and } i \notin \mathcal{I}_j, \qquad (3.4) \\
\textstyle\sum_i \alpha_i^j = n - n_{\mathcal{M}} \qquad & \text{for } j \notin \mathcal{M}. \qquad (3.5)
\end{aligned}
$$

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 (2.15) and (2.16) imply that $\alpha_i^{j^*} = 1$ for $i \notin \mathcal{I}_{j^*}$ since $\xi_i^{j^*} = (b^{j^*} - y_i^{j^*})_+ > 0$. However, by (2.14), $\sum_i \alpha_i^j$ should be the same for all j, which implies that $\alpha_i^j = 1$ for other $j \in \mathcal{M}$ and $i \notin \mathcal{I}_j$ to have the same sum of $n - n_{\mathcal{M}}$. This proves (3.4) and (3.5) in particular, and the rest follows immediately.

Remark 2. The value of (2.5) for the initial $\boldsymbol{b}$ in Lemma 2 is $(n-n_{\mathcal{M}})k/(k-1)$ except for 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 $\alpha_i^j = 1$ for each j and $i \notin \mathcal{I}_j$. Again, Lemma 2 is a generalized version of Lemma 1 for the balanced binary case in Hastie et al. (2004).

We start from λ sufficiently large but indefinite, which determines the limit MSVM solution in the foregoing two lemmas, and decrease λ until non-trivial solutions emerge. To find such a genuine starting value of λ and the corresponding $\boldsymbol{b}$, we consider two possible cases of the limit solution. First, $\alpha_i^j \in \{0, 1\}$ for all j and $i \notin \mathcal{I}_j$. For the completely balanced situation, that is the case. Unbalanced class proportions may also lead to the first case. As λ decreases, $\boldsymbol{\alpha}$ changes, but with the restriction that the equality constraint (2.14) is satisfied. (2.14) states that the sum of the Lagrange multipliers α_i^j is the same across all $j = 1, \ldots, k$. Any change in $\boldsymbol{\alpha}$ is bound to reduce α_i^j with $j \in \mathcal{M}$ from 1 because the values of such α_i^j's in the limit solution are at their maxima. Consequently, this change would reduce some α_i^j with $j \notin \mathcal{M}$ from 1 as well, by (2.14). Hence, some k indices of α_i^j, one from each j, should enter the elbow set $\mathcal{E}$ simultaneously. Note that the lower set of the limit solution is empty. Letting $\mathcal{B}^j = \{i|\ \alpha_i^j = 1\}$ for each j, we choose the data index $i_*^j = \text{argmin}_{i \in \mathcal{B}^j} H_i^j$, where $H_i^j = -\sum_{r=1}^n (\alpha_r^j - \bar{\alpha}_r) K(\boldsymbol{x}_r, \boldsymbol{x}_i)$. These k indices are chosen to satisfy $f^j(\boldsymbol{x}_{i_*^j}) = -1/(k-1)$, yielding k equations that determine the initial λ and $\boldsymbol{b}$.

Second, there could be two or more $0 < \alpha_i^j < 1$ for some $j \notin \mathcal{M}$, by (3.3) and (3.5). By the same logic as in the first case, any change in $\boldsymbol{\alpha}$ would reduce α_i^j from 1 for other component(s) j with $\alpha_i^j = 0$ or 1 only. By Lemmas 1 and 2, there is at least one component $j \in \mathcal{M}$ for which no data index falls into the elbow set of the limit solution. In this case, α_i^j for $(i, j) \in \mathcal{E}$ stays the same until other components j without such index i have a point reaching the margin $f_i^j = y_i^j$. A formal proof of this fact is given in the next section. As a result, α_i^j strictly between 0 and 1 will remain on the elbow set until each of the other component(s) j has a data index in the elbow set. So if there is l such that $0 < \alpha_l^j < 1$ for j, we define $i_*^j = l$, otherwise $i_*^j = \text{argmin}_{i \in \mathcal{B}^j} H_i^j$. Again, $f^j(\boldsymbol{x}_{i_*^j}) = -1/(k-1)$ gives a set of k equations as follows. For $j = 1, \ldots, k$,

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

Solving the equations, we have the initial λ and $\boldsymbol{b}$ in both scenarios:

$$\lambda = \frac{k-1}{kn} \sum_{j=1}^k \sum_{r=1}^n (\alpha_r^j - \bar{\alpha}_r) K(\boldsymbol{x}_r, \boldsymbol{x}_{i_*^j}) \quad \text{and} \tag{3.6}$$

$$b^j = -\frac{1}{k-1} + \frac{1}{n\lambda}\sum_{r=1}^{n}(\alpha_r^j - \bar{\alpha}_r)K(\boldsymbol{x}_r, \boldsymbol{x}_{i_*^j}). \tag{3.7}$$

Since the above initial λ and $\boldsymbol{b}$ depend on i_*^j only through the value of H_i^j, they are uniquely determined regardless of the choice of i_*^j with the minimum H_i^j.

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 $(i,j) \in \mathcal{E}$. Accordingly, this allows us to find some of the Lagrange multipliers fully and determine the coefficients in (2.2). 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 events that can change the elbow set.

1. An index (i,j) leaves from $\mathcal{E}$ to join either $\mathcal{L}$ or $\mathcal{U}$.
2. An index (i,j) from $\mathcal{L}$ enters $\mathcal{E}$.
3. An index (i,j) from $\mathcal{U}$ enters $\mathcal{E}$.

Continuity of the objective function (2.12) in λ implies that the minimizer $\boldsymbol{\alpha}$, as a function of λ, changes continuously between consecutive values of λ at which one of the above three events occurs, and so does the MSVM solution. Consider $\{\lambda_\ell, \ell = 0, 1, \dots\}$, a decreasing sequence of λ starting from the initial value in (3.6) and indicating the values at which some change occurs in $\mathcal{E}$. In fact, the sequence determines the break points of λ at which $\boldsymbol{\alpha}$ can be completely characterized. For λ_ℓ, denote the corresponding elbow, upper and lower sets by $\mathcal{E}_\ell$, $\mathcal{U}_\ell$, and $\mathcal{L}_\ell$ respectively. Letting $\alpha_0^j = n\lambda b^j$, we write

$$\hat{f}_\lambda^j(\boldsymbol{x}) = \frac{1}{n\lambda}\Big(-\sum_{i=1}^{n}(\alpha_i^j - \bar{\alpha}_i)K(\boldsymbol{x}_i, \boldsymbol{x}) + \alpha_0^j\Big).$$

For $\lambda_{\ell+1} < \lambda < \lambda_\ell$, we can express $\hat{f}_\lambda^j(\boldsymbol{x})$ in terms of an incremental change from $\hat{f}_{\lambda_\ell}^j(\boldsymbol{x})$. Denoting α_i^j at λ_ℓ by $\alpha_{i(\ell)}^j$,

$$\begin{aligned}
\hat{f}_\lambda^j(\boldsymbol{x}) &= \Big[\hat{f}_\lambda^j(\boldsymbol{x}) - \frac{\lambda_\ell}{\lambda}\hat{f}_{\lambda_\ell}^j(\boldsymbol{x})\Big] + \frac{\lambda_\ell}{\lambda}\hat{f}_{\lambda_\ell}^j(\boldsymbol{x}) \\
&= \frac{1}{n\lambda}\Big[-\sum_{i=1}^{n}\big((\alpha_i^j - \alpha_{i(\ell)}^j) - (\bar{\alpha}_i - \bar{\alpha}_{i(\ell)})\big)K(\boldsymbol{x}_i, \boldsymbol{x}) + (\alpha_0^j - \alpha_{0(\ell)}^j) + n\lambda_\ell\hat{f}_{\lambda_\ell}^j(\boldsymbol{x})\Big] \\
&= \frac{1}{n\lambda}\Big[-\sum_{i\in\cup_j\mathcal{E}_\ell^j}\big((\alpha_i^j - \alpha_{i(\ell)}^j) - (\bar{\alpha}_i - \bar{\alpha}_{i(\ell)})\big)K(\boldsymbol{x}_i, \boldsymbol{x}) + (\alpha_0^j - \alpha_{0(\ell)}^j) + n\lambda_\ell\hat{f}_{\lambda_\ell}^j(\boldsymbol{x})\Big],
\end{aligned}$$

where $\mathcal{E}_\ell^j = \{i |\ (i,j) \in \mathcal{E}_\ell\}$ for each j. The last equality holds because $\alpha_i^j = 0$ or 1 for all j without any change from $\alpha_{i(\ell)}^j$ if $i \notin \cup_j\mathcal{E}_\ell^j$. For all $(i,j) \in \mathcal{E}_\ell$, $\hat{f}_\lambda^j(\boldsymbol{x}_i) = y_i^j$.

Letting $\delta_0^j = \alpha_0^j - \alpha_{0(\ell)}^j$, $\delta_i^j = -(\alpha_i^j - \alpha_{i(\ell)}^j)$, $\bar{\delta}_i = -(\bar{\alpha}_i - \bar{\alpha}_{i(\ell)})$ for $i \geq 1$, and $\mathcal{E}_\ell^{\mathcal{I}} = \cup_j \mathcal{E}_\ell^j$, we have

$$\hat{f}_\lambda^j(\boldsymbol{x}_i) = \frac{1}{n\lambda}\Big[\sum_{r\in\mathcal{E}_\ell^{\mathcal{I}}} \big(\delta_r^j - \bar{\delta}_r\big)K(\boldsymbol{x}_r, \boldsymbol{x}_i) + \delta_0^j - \frac{n\lambda_\ell}{k-1}\Big] = -\frac{1}{k-1},$$

which gives

$$\sum_{r\in\mathcal{E}_\ell^{\mathcal{I}}} \big(\delta_r^j - \bar{\delta}_r\big)K(\boldsymbol{x}_r, \boldsymbol{x}_i) + \delta_0^j = \frac{n(\lambda_\ell - \lambda)}{k-1}. \tag{3.8}$$

Given any λ, $\sum_{i=1}^n \alpha_i^j$ should be the same for all j and $\sum_{j=1}^k b^j = 0$ by the sum-to-zero constraint. This yields $\sum_{i\in\mathcal{E}_\ell^1} \delta_i^1 = \cdots = \sum_{i\in\mathcal{E}_\ell^k} \delta_i^k$ and $\sum_{j=1}^k \delta_0^j = 0$. As a result, these k constraints and (3.8) 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 (3.8) 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 $\boldsymbol{\delta}_0 = (\delta_0^1, \ldots, \delta_0^k)^t$ and $\boldsymbol{\delta} = (\boldsymbol{\delta}^1, \ldots, \boldsymbol{\delta}^k)^t$ with $\boldsymbol{\delta}^j = (\delta_{i_1^j}, \ldots, \delta_{i_{m_j}^j})$. Note that (3.8) depends only on $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}$ since $\delta_i^j = 0$ for all $(i,j) \notin \mathcal{E}_\ell$. $\boldsymbol{K}_\ell^* = [\boldsymbol{K}_{lj}^*]$ is the square block matrix of $|\mathcal{E}_\ell| = m_1 + \cdots + m_k$ rows and columns, whose ljth block $(l, j = 1, \ldots, k)$ is given by

$$\boldsymbol{K}_{lj}^* = \Big(I(l=j) - \frac{1}{k}\Big) \begin{pmatrix} K(\boldsymbol{x}_{i_1^j}, \boldsymbol{x}_{i_1^l}) & \ldots & K(\boldsymbol{x}_{i_{m_j}^j}, \boldsymbol{x}_{i_1^l}) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_{i_1^j}, \boldsymbol{x}_{i_{m_l}^l}) & \ldots & K(\boldsymbol{x}_{i_{m_j}^j}, \boldsymbol{x}_{i_{m_l}^l}) \end{pmatrix}.$$

Also, define

$$\mathbf{1}_\delta = \begin{pmatrix} \boldsymbol{e}_{m_1}^t & -\boldsymbol{e}_{m_2}^t & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{e}_{m_2}^t & -\boldsymbol{e}_{m_3}^t & & \mathbf{0} \\ \vdots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{e}_{m_{k-1}}^t & -\boldsymbol{e}_{m_k}^t \end{pmatrix} \text{ and } \mathbf{1}_0 = \begin{pmatrix} \boldsymbol{e}_{m_1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{e}_{m_2} & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{e}_{m_k} \end{pmatrix},$$

where $\boldsymbol{e}_m$ is the vector of m ones and $\mathbf{0}$ indicates a vector of zeros of appropriate length. Then, (3.8) and the constraints are succinctly expressed as

$$\begin{pmatrix} \mathbf{0} & \boldsymbol{e}_k^t \\ \boldsymbol{K}_\ell^* & \mathbf{1}_0 \\ \mathbf{1}_\delta & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\delta}_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{n(\lambda_\ell - \lambda)}{k-1} \boldsymbol{e}_{|\mathcal{E}_\ell|} \\ \mathbf{0} \end{pmatrix}. \tag{3.9}$$

Letting $\boldsymbol{A}_\ell$ be the square matrix on the left-hand side of (3.9), and $\boldsymbol{v}_\ell^t = (0, \boldsymbol{e}_{|\mathcal{E}_\ell|}^t, \mathbf{0})$, we solve for $\boldsymbol{\delta}_0$ and $\boldsymbol{\delta}$. If $\boldsymbol{A}_\ell$ is invertible,

$$\begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\delta}_0 \end{pmatrix} = \frac{n(\lambda_\ell - \lambda)}{k-1} \boldsymbol{A}_\ell^{-1} \boldsymbol{v}_\ell.$$

Abbreviating $\boldsymbol{A}_\ell^{-1}\boldsymbol{v}_\ell$ by $\boldsymbol{w}_\ell$, we have

$$\begin{aligned}\boldsymbol{\alpha}_0 - \boldsymbol{\alpha}_{0(\ell)} &= n(\lambda_\ell - \lambda)/(k-1)\boldsymbol{w}_0 \text{ and} \\ \alpha_i^j - \alpha_{i(\ell)}^j &= n(\lambda - \lambda_\ell)/(k-1)w_i^j \text{ for } (i,j) \in \mathcal{E}_\ell, \end{aligned} \tag{3.10}$$

where $\boldsymbol{w}_0$ is the last k elements of $\boldsymbol{w}_\ell$ and w_i^j is the element of $\boldsymbol{w}_\ell$ corresponding to δ_i^j. This shows that the scaled intercepts and the Lagrange multipliers in the elbow set change linearly in λ on the interval $(\lambda_{\ell+1}, \lambda_\ell)$. Rescaling them properly by $n\lambda$ to obtain equations for the coefficients, we have

$$\boldsymbol{b} = \frac{\lambda_\ell}{\lambda}\Big(\boldsymbol{b}_\ell + \frac{1}{k-1}\boldsymbol{w}_0\Big) - \frac{1}{k-1}\boldsymbol{w}_0 \text{ and} \tag{3.11}$$

$$c_i^j = \frac{\lambda_\ell}{\lambda}\Big(c_{i(\ell)}^j + \frac{w_i^j - \bar{w}_i}{k-1}\Big) - \frac{w_i^j - \bar{w}_i}{k-1}. \tag{3.12}$$

Here $\bar{w}_i = (1/k)\sum_j w_i^j$ and the summation is only over j's with $(i,j) \in \mathcal{E}_\ell$. This proves the following theorem concerning piecewise linearity of the paths of the coefficients $\boldsymbol{b}$ and $\boldsymbol{C}$.

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

Likewise, the jth coordinate of the MSVM output has a path linear in $1/\lambda$:

$$\hat{f}_\lambda^j(\boldsymbol{x}) = \frac{\lambda_\ell}{\lambda}\Big(\hat{f}_{\lambda_\ell}^j(\boldsymbol{x}) - \hat{g}_{\lambda_\ell}^j(\boldsymbol{x})\Big) + \hat{g}_{\lambda_\ell}^j(\boldsymbol{x}), \tag{3.13}$$

where $\hat{g}_{\lambda_\ell}^j(\boldsymbol{x}) = -\Big(\sum_{i\in\mathcal{E}_\ell^{\mathcal{I}}}(w_i^j - \bar{w}_i)K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0^j\Big)/(k-1)$. $\hat{g}_{\lambda_\ell}^j(\boldsymbol{x})$ is pivotal to $\hat{f}_\lambda^j(\boldsymbol{x})$ in the sense that $\hat{f}_\lambda^j(\boldsymbol{x})$ can be expressed as a scaled $\hat{f}_{\lambda_\ell}^j(\boldsymbol{x})$, once both are pivoted on $\hat{g}_{\lambda_\ell}^j(\boldsymbol{x})$.

So far, we have discussed how the MSVM solution path is explicitly characterized as a function of λ when the elbow set $\mathcal{E}_\ell^j$ for each j is not empty. If there is at least one empty elbow set $\mathcal{E}_\ell^j$ at $\lambda = \lambda_\ell$, then the constraints used in the previous characterization need to be modified. The constraint that the sum of α_i^j should stay the same for all j now becomes $\sum_{i\in\mathcal{E}_\ell^j}\delta_i^j = 0$ for each non-empty elbow set $\mathcal{E}_\ell^j$. We eliminate the component(s) corresponding to the empty $\mathcal{E}_\ell^j$ from $\boldsymbol{\delta}_0$ and the corresponding column(s) from $\mathbf{1}_0$, and denote the resulting vector and matrix by $\boldsymbol{\delta}_0^*$ and $\mathbf{1}_0^*$, respectively. Then (3.9) is adjusted by taking into account the presence of some empty elbow set(s):

$$\begin{pmatrix} \boldsymbol{K}_\ell^* & \mathbf{1}_0^* \\ \mathbf{1}_0^{*t} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\delta}_0^* \end{pmatrix} = \begin{pmatrix} \frac{n(\lambda_\ell - \lambda)}{k-1}\boldsymbol{e}_{|\mathcal{E}_\ell|} \\ \mathbf{0} \end{pmatrix}. \tag{3.14}$$

Assuming that $\boldsymbol{K}_\ell^*$ is of full rank, we solve for $\boldsymbol{\delta}$ and $\boldsymbol{\delta}_0^*$. Due to the simple structure of the block matrix on the left-hand side, its inverse can be easily written out to give the explicit solution of $\boldsymbol{\delta} = \mathbf{0}$ and $\boldsymbol{\delta}_0^* = n(\lambda_\ell - \lambda)/(k-1)\boldsymbol{e}_{k^*}$, where k^* is the number of non-empty elbow sets $\mathcal{E}_\ell^j$. This yields

$$c_i^j = \frac{\lambda_\ell}{\lambda} c_{i(\ell)}^j \text{ for } (i,j) \in \mathcal{E}_\ell \text{ and}$$
$$b^j = \frac{\lambda_\ell}{\lambda}\Big(b_\ell^j + \frac{1}{k-1}\Big) - \frac{1}{k-1} \text{ for non-empty } \mathcal{E}_\ell^j.$$

When there is more than one empty $\mathcal{E}_\ell^j$, δ_0^j's for the empty sets are not uniquely determined other than that they are constrained to satisfy $\sum_j \delta_0^j = 0$. 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 result give the desired conclusion: the solution path of the MSVM is piecewise linear.

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

4. Computation

We show here how to generate the decreasing sequence of $\{\lambda_\ell, \ell = 0, 1, \dots\}$ that determines the joints of the piecewise linear MSVM solution path. Given λ_ℓ, we find $\lambda_{\ell+1}$ by considering the following possible events.

1. An index (i,j) in $\mathcal{E}_\ell$ leaves the elbow set, and α_i^j $(0 \le \alpha_i^j \le 1)$ becomes either 0 or 1.
2. An index (i,j) in $\mathcal{L}_\ell$ or $\mathcal{U}_\ell$ joins the elbow set, and $\hat{f}^j(\boldsymbol{x}_i)$ is then y_i^j.

When the first type of event happens, a candidate $\lambda_{\ell+1}$ is obtained by setting α_i^j at (3.10) to 0 or 1. For the second type of event, set the left-hand side of (3.13) to y_i^j and consider

$$\lambda_{\ell+1} = \frac{\lambda_\ell(\hat{f}_{\lambda_\ell}^j(\boldsymbol{x}_i) - \hat{g}_{\lambda_\ell}^j(\boldsymbol{x}_i))}{y_i^j - \hat{g}_{\lambda_\ell}^j(\boldsymbol{x}_i)}.$$

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

If there is at least one empty elbow set $\mathcal{E}_\ell^j$ at λ_ℓ, then the α_i^j in non-empty sets stay 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 $\mathcal{E}_\ell^j$ hits the margin $\hat{f}^j(\boldsymbol{x}_i) = y_i^j$

simultaneously. Such a point for each empty elbow set $\mathcal{E}_\ell^j$ is determined via the same argument as in the initialization process. Thus, the corresponding $\lambda_{\ell+1}$ is identified by (3.6).

We stop the process and trace out the solution path if the upper set $\mathcal{U}_\ell$ is empty, since the empirical risk functional at λ_ℓ is zero in this case, completely overfitting the data. As λ gets smaller, the upper sets are bound to become empty in separable problems. For non-separable problems, monitoring change in the solutions at consecutive break points can shorten the procedure. Inspection of (3.13) provides a rule that stops at λ_ℓ if $\max_j (1/n) \sum_{i=1}^n |\hat{f}_{\lambda_\ell}^j(\boldsymbol{x}_i) - \hat{g}_{\lambda_\ell}^j(\boldsymbol{x}_i)| < \epsilon$, where ϵ denotes a prespecified tolerance for declaring 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 λ 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 $|\mathcal{E}_\ell| + k$ unknowns at each λ_ℓ. As λ decreases, the cardinality of the elbow set $\mathcal{E}_\ell$ 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 λ_ℓ. Such a measure helps us judge whether the optimal value of the regularization parameter has been attained or not. If attained, then we stop without completing the entire path.

The computational complexity of the path finding algorithm is proportional to the number of break points λ_ℓ. At each break point, a system of linear equations needs to be solved. In general, solving a system of linear equations with m unknowns takes $O(m^3)$ operations, but it can be reduced to $O(m^2)$ in this case due to an incremental change in the successive linear equations. For determining the next break point, evaluation of $\hat{f}_{\lambda_\ell}^j$ and $\hat{g}_{\lambda_\ell}^j$ at n data points needs to be done, which takes $O((k-1)n|\mathcal{E}_\ell^{\mathcal{I}}|)$ operations. The effect of the class size k is felt throughout the computation in that $|\mathcal{E}_\ell|$, the intermediate function evaluations, and the number of break points tend to increase in proportion to $(k-1)$ when compared to the binary case.

5. A Numerical Example

To illustrate the algorithm characterizing the MSVM solution path, we consider a simulated three-class example. The simulation setting is as follows. For Class 1, two covariates $\boldsymbol{X} = (X_1, X_2)$ are generated from a normal distribution with mean $(3, 0)$. For Class 2, $\boldsymbol{X}$ 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, $\boldsymbol{X}$ has a normal distribution with mean $(-1, 1)$. For all 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 5.2 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 the Gaussian kernel $K(\boldsymbol{s}, \boldsymbol{t}) = \exp(-\gamma\|\boldsymbol{s}-\boldsymbol{t}\|^2)$ was applied to the simulated data, and the entire solution path was traced. Here the additional parameter γ was fixed at 1.

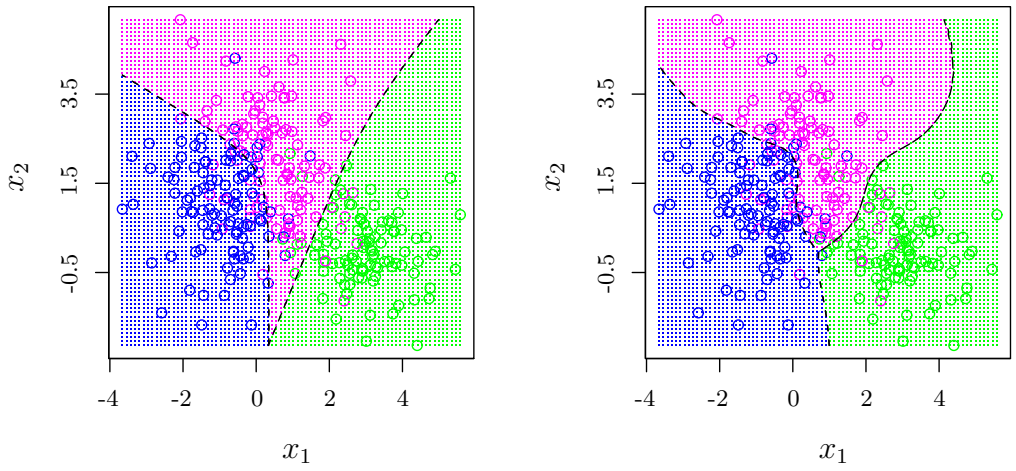

Figure 5.2. Left: the boundaries of the Bayes classification rule in a scatter plot of the training data. Right: the classification boundaries determined by the MSVM at an appropriately chosen λ; Class 1: green, Class 2: magenta, and Class 3: blue.

Figure 5.3 shows how the test error rates, when evaluated over the test set, change as the regularization parameter λ varies. Notice that the x-axis is $\log(1/\lambda)$, so λ decreases in the positive direction of x. As λ 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 λ is critical to the performance of the solution. Note that the test error rate curve in the figure was cropped to show a portion of the full range of λ, since the test error rates rise up sharply as λ 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 can achieve theoretically optimal accuracy if λ is chosen appropriately. The right panel of Figure 5.2 shows the classification boundaries induced by the MSVM solution at this optimal λ.

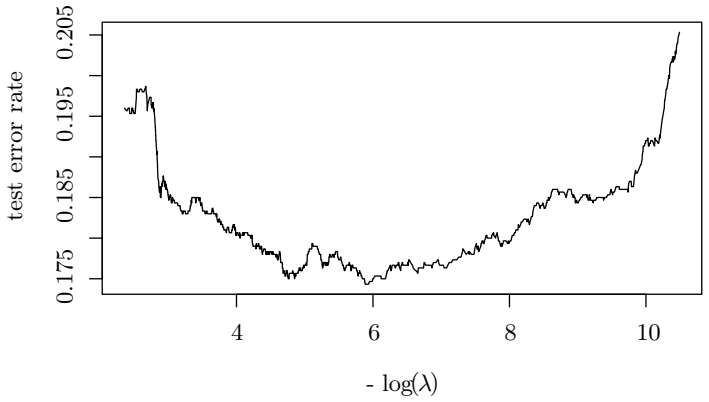

Figure 5.3. Test error rate as a function of λ.

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, and we will not delve into the issue here. General approaches to tuning, equivalently model selection, can be found in, for example, Efron (1986) and Wahba (1990). Just for illustration of the feasibility of tuning, 100 new data sets of size 300 were generated as random replicates of a tuning set, and λ was chosen to minimize the error rate over each tuning set. Figure 5.4 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 5.3. 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 λ, depends on the stopping criterion used. The rule adopted in this implementation of the algorithm does not consider the two stopping criteria discussed in Section 4 only, but also employs a preset lower bound of λ and a 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 λ was reached first before any other criteria for this example, and there were 3014 breaking points of λ.

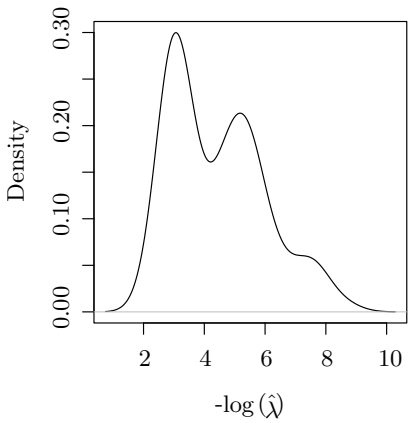

Figure 5.4. An estimated density function of the optimal parameters $\hat{\lambda}$ chosen by 100 tuning sets of sample size 300.

Availability of the entire solution path allows us to visualize the effect of the regularization parameter on the MSVM solution from various angles. For example, Figure 5.5 depicts the complete paths of $\hat{f}^1_\lambda(\boldsymbol{x}_i)$, $\hat{f}^2_\lambda(\boldsymbol{x}_i)$, and $\hat{f}^3_\lambda(\boldsymbol{x}_i)$ for the isolated instance from Class 3 (blue) in the top left region. The paths start off at the initial solution of $(0, 0, 0)$, begin to diverge for a better fit to the data as λ decreases, returning Class 2 as the predicted class for moderate values of λ. As λ further decreases, the solution paths tend to follow the data too closely and $\hat{f}^3_\lambda(\boldsymbol{x}_i)$ emerges as the maximum component. This is a snapshot of the spectrum of solutions ranging from the least to the most complete fit to the data as controlled by λ.

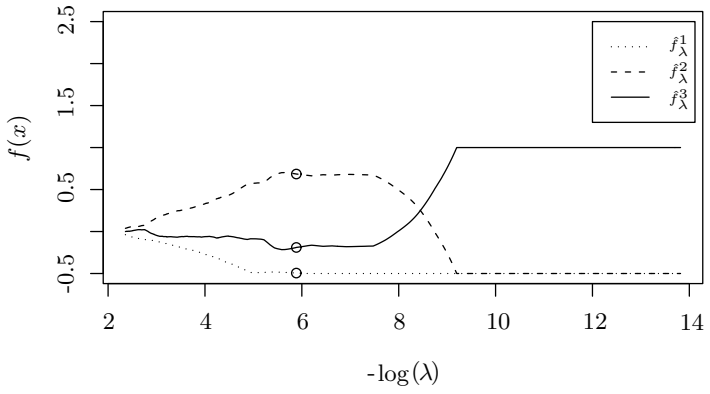

Figure 5.5. The entire paths of $\hat{f}^1_\lambda(\boldsymbol{x}_i)$, $\hat{f}^2_\lambda(\boldsymbol{x}_i)$, and $\hat{f}^3_\lambda(\boldsymbol{x}_i)$ for an outlying instance $\boldsymbol{x}_i$ from Class 3. The circles correspond to λ with the minimum test error rate.

As mentioned before, the computational complexity of the proposed algorithm depends on the size of elbow set at each λ_ℓ, and the elbow size is bound to increase as λ gets smaller. Figure 5.6 shows how the size of elbow set changes as a function λ for each class. The median sizes of elbow sets were 39, 30 and 30 for Classes 1, 2 and 3, respectively, while the maximum elbow sizes were 68, 51 and 61. Around the optimal value of λ, 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 $\mathcal{E}_\ell^j$ is almost constant until λ reaches the optimal value around $\exp(-6)$, and it becomes smaller soon after that, then remains almost constant. It would be desirable to have a theoretical explanation for this, for this might help us devise an early stopping rule.

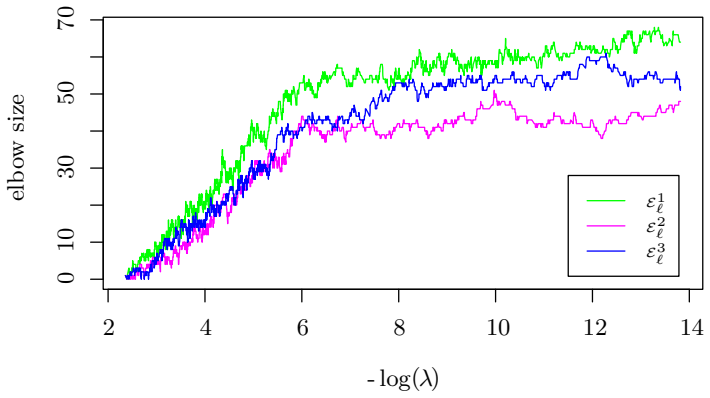

Figure 5.6. The size of elbow set $\mathcal{E}_\ell^j$ for three classes as a function λ.

6. Discussion

It is conceptually attractive to have the whole solution path in perspective, and it is computationally effective to construct the path sequentially with the basic operation of solving a system of linear equations. Another type of a regularization problem, which may well benefit from the idea, is feature selection via an 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 a sparse solution approach to feature selection for the MSVM is described in Lee, Kim, Lee and Koo (2004) by using functional ANOVA decomposition. Besides λ, there is another regularization parameter λ_θ in the setting that governs the sum of non-negative weights θ_ν assigned to each feature x_ν. The sum of θ_ν 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 a model path enables us to understand the data better, and especially it helps us to find out multiple good descriptions of sparse data. We remark here that this approach is closely related to the least angle regression proposed by Efron, Hastie, Johnstone and Tibshirani (2004) for variable selection in multiple linear regression, despite the notable difference between the two settings. We also note in passing that for the special case of an l_1-norm linear MSVM, Wang and Shen (2005) recently studied an algorithm to construct the solution path for simultaneous fitting and feature selection.

Another direction of interest is the investigation of model selection issues that the 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 a 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 directions that we plan to pursue include comparison of a variety of old and new model selection criteria, the 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

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *J. Amer. Statist. Assoc.* **81**, 461-470.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.* **32**, 407-499.

Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004). The entire regularization path for the support vector machine. *J. Mach. Learn. Res.* **5**, 1391-1415.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Networks* **13**, 415-425.

Joachims, T. (1999). Making large-scale SVM learning practical. *In Advances in Kernel Methods - Support Vector Learning* (Edited by B. Schölkopf, C. Burges and A. Smola). MIT Press.

Lee, Y., Kim, Y., Lee, S. and Koo, J.-Y. (2004). Structured Multicategory Support Vector Machine with ANOVA decomposition. Technical Report 743, Department of Statistics, The Ohio State University.

Lee, Y., Lin, Y. and Wahba, G. (2004). Multicategory Support Vector Machines, theory, and application to the classification of microarray data and satellite radiance data. *J. Amer. Statist. Assoc.* **99**, 67-81.

Mangasarian, O. (1994). *Nonlinear Programming*. Classics in Applied Mathematics, Vol. 10. SIAM, Philadelphia.

Platt, J. (1999). Sequential minimal optimization: A fast algorithm for training support vector machines. *In Advances in Kernel Methods: Support Vector Learning* (Edited by B. Schölkopf, C. J. C. Burges and A. J. Smola), 185-208. MIT Press.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, New York.

Wahba, G. (1990). *Spline Models for Observational Data*. Series in Applied Mathematics, Vol. 59. SIAM, Philadelphia.

Wang, L. and Shen, X. (2005). Multi-category Support Vector Machines, feature selection and solution path. Unpublished manuscript.

Department of Statistics, The Ohio State University

E-mail: yklee@stat.ohio-state.edu

Department of Statistics, The Ohio State University

E-mail: zhenhuan@stat.ohio-state.edu

(Received April 2005; accepted October 2005)