Support Vector Machines for Classification: A Statistical Portrait

YOONKYUNG LEE

Department of Statistics, The Ohio State University, Columbus, Ohio 43210

Author:

Yoonkyung Lee

Department of Statistics

The Ohio State University

1958 Neil Ave

Columbus, OH 43210

Phone: 614-292-9495

Fax: 614-292-2096

E-mail: yklee@stat.osu.edu

Abstract

The support vector machine is a supervised learning technique for classification increasingly used in many applications of data mining, engineering, and bioinformatics. This chapter aims to provide an introduction to the method, covering from the basic concept of the optimal separating hyperplane to its nonlinear generalization through kernels. A general framework of kernel methods that encompass the support vector machine as a special case is outlined. In addition, statistical properties that illuminate both advantage and limitation of the method due to its specific mechanism for classification are briefly discussed. For illustration of the method and related practical issues, an application to real data with high dimensional features is presented.

Key Words: Classification, Machine learning, Kernel methods, Regularization, Support vector machine

1 Introduction

Classification is a type of statistical problem where we want to predict a predefined class membership accurately based on features of an individual. For instance, pathologists wish to diagnose a patient either healthy or diseased, based on some measurements from the patient's tissue sample. In general, the foremost goal of classification is to learn the discrimination rule attaining the minimum error rate over novel cases. In the statistics literature, Fisher's linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are classical examples of a discriminant rule, and modern statistical tools include classification trees, logistic regression, neural networks, and kernel density based methods. For reference to classification in general, see (1–3).

This chapter introduces the support vector machine (SVM), a classification method which has drawn tremendous attention in machine learning, a thriving area of computer science, for the last decade or so. It has been successfully used in many applications of data mining, engineering, and bioinformatics; for instance, hand-written digit recognition, text categorization, and tumor classification with genomic profiles. Motivated by the statistical learning theory (4) that Vapnik and Chervonenkis developed, Vapnik and his collaborators (5) proposed the optimal separating hyperplane and its nonlinear generalization for pattern recognition in the early 90's, which is now known as the SVM. For complete treatment of the subject, see (4, 6, 7) and references therein.

The method has gained its popularity in part due to simple geometric interpretation, competitive classification accuracy in practice, and an elegant theory behind. In addition, it has such operational characteristics as sparsity and duality, and they render the method an appealing data-analytic tool. The sparsity of the SVM solution leads to efficient data reduction for massive data at the testing stage, and the mathematical duality allows coherent handling of high dimensional data. The latter property, in particular, seems to be fitting for modern data analysis, as nowadays data with high dimensional features are quite prevalent due to technological advances in many areas of science and industry. In fact, the aforementioned successful applications all involve high dimensional data.

On the statistical side, a salient aspect of the SVM as a classification rule is its mechanism to directly focus on the decision boundary. One of the earlier references of the SVM (8) begins by noting how quickly the number of parameters to estimate increases in Fisher's normal discriminant paradigm as the dimension of the feature space increases. Instead of probability model parameters, the SVM aims at classification boundary directly by a hyperplane with maximum margin, amending the non uniqueness of Rosenblatt's perceptron (9) (an earlier attempt to find a hyperplane for discrimination). This 'hard' classification approach departs from more traditional approach of 'soft' classification through estimation of the underlying probability model that generates data. Whether the latter is more appropriate than the former depends largely on the context of applications, and their relative efficiency still remains to be a subject of controversy.

Contrasting hard classification with soft classification, this chapter provides an overview of the SVM with more focus on conceptual understanding than technical details, for beginners in the field of statistical learning. Geometric formulation of the method, related computation, and the resulting operational characteristics are outlined. Various aspects of the method are examined with more emphasis on its statistical properties and connection than other tutorials, for instance, (10–12). By doing so, its advantages as well as limitation are highlighted. For illustration of the method, a data example is provided with discussion of some practical issues arising in its applications.

2 Method

Consider a classification problem where multivariate attributes such as expression levels of genes are measured for each subject in a data set as potential molecular markers for a biological or clinical outcome of interest (e.g. the status of a disease or its progression). Let $X = (X_1, \dots, X_p) \in \mathcal{X} = \mathbb{R}^p$ denote the predictors and Y the variable for the categorical outcome which takes one of, say, k nominal *class labels*, $\mathcal{Y} = \{1, \dots, k\}$. Then the so-called *training data* are given as a set of n observation pairs, $\mathcal{D}_n = \{(x_i, y_i), i = 1, \dots, n\}$, where (x_i, y_i)'s are viewed as independent and identically distributed random outcomes of (X, Y) from some unknown distribution $P_{X,Y}$.

The ultimate goal of classification is to understand informative patterns that exist in the predictors in relation to their corresponding class labels. For that reason, classification is known as pattern recognition in the computer science literature. Formally, it aims to find a map (*classification rule*), $\phi : \mathcal{X} \to \mathcal{Y}$ based on the training data which can be generalized to future cases from the same distribution $P_{X,Y}$.

For simplicity, this section is focused on classification with binary outcomes only ($k = 2$). Typically, construction of such a rule ϕ is done by finding a real-valued discriminant function f first and taking either the indicator $\phi(x) = I(f(x) \geq 0)$ if two classes are labeled as 0 or 1, or its sign $\phi(x) = sgn(f(x))$ if they are symmetrically labeled as ± 1. In the latter, the classification boundary is determined by the zero level set of f, i.e. $\{x : f(x) = 0\}$.

2·1 Linearly Separable Case

With $\mathcal{Y} = \{-1, 1\}$, first consider a simple scenario as depicted in Figure 1, where two classes in the training data are linearly separable, so a linear discriminant function, $f(x) = \beta' x + \beta_0$, could be adequate for classification. Fisher's LDA is a standard example of linear classifiers in statistics, which is proven to be optimal in minimizing the misclassification rate under

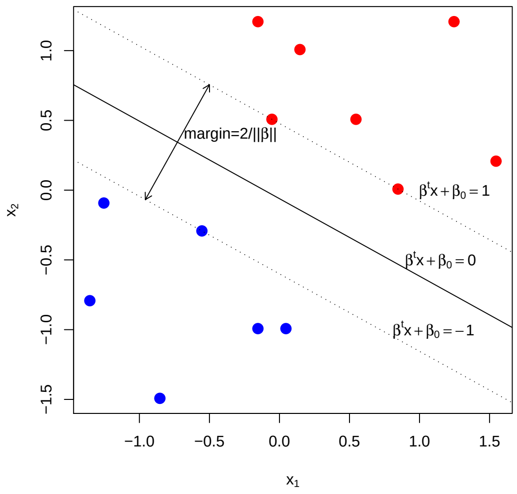

Figure 1: This toy example illustrates training data ith two predictors and binary class labels (red: 1 and blue: -1). The solid line indicates the optimal separating hyperplane.

the normality and equal covariance assumptions on the distributions of predictors for two classes.

In contrast to the LDA, without such distributional assumptions, the discriminant function of the linear SVM is determined directly through the corresponding hyperplane, $\beta' x + \beta_0 = 0$, or the classification boundary itself. The perceptron algorithm (9) is a precursor of the SVM in the sense that both search for a hyperplane for discrimination. However, in the situations illustrated in Figure 1, there are infinitely many separating hyperplanes, and the former intends to find just one by sequentially updating β and β_0 while the latter looks for the hyperplane with the maximum margin between two classes, which is uniquely determined. The margin is defined as the distance between the two convex hulls formed by x_i's with class labels 1 and -1, respectively, and it can be mathematically characterized as follows. When the training data are linearly separable, there exist $\delta > 0$, β_0, and β such that

$$\begin{aligned} \beta' x_i + \beta_0 &\geq \delta \quad \text{for } y_i = 1 \text{ and} \\ \beta' x_i + \beta_0 &\leq -\delta \quad \text{for } y_i = -1. \end{aligned}$$

Then, without loss of generality, δ can be set to 1 by normalizing β_0 and β. This leads to the following *separability condition*:

$$y_i(\beta' x_i + \beta_0) \geq 1 \quad \text{for all } i = 1, \ldots, n. \tag{1}$$

So, the margin between the two classes is the same as the sum of the distances from the nearest x_i's with $y_i = \pm 1$ to the hyperplane $\beta' x + \beta_0 = 0$. Since the distance of a point $x_0 \in \mathbb{R}^p$ from a hyperplane $\beta' x + \beta_0 = 0$ is given by $|\beta' x_0 + \beta_0|/\|\beta\|$, under the specified normalization of the separating hyperplane, the margin is given as $2/\|\beta\|$. Maximizing the margin is mathematically equivalent to minimizing its reciprocal or a monotonically decreasing function of it in general, for example, $\|\beta\|^2/2$. For the optimal hyperplane, the SVM finds β_0 and β minimizing

$$\frac{1}{2}\|\beta\|^2 \;\; \text{subject to } y_i(\beta' x_i + \beta_0) \geq 1 \text{ for all } i = 1, \ldots, n. \tag{2}$$

Once the minimizer $(\hat{\beta}_0, \hat{\beta})$ is obtained, the induced SVM classifier is given as

$$\phi_{SVM}(x) = sgn(\hat{\beta}' x + \hat{\beta}_0).$$

The solid line in Figure 1 indicates the boundary of the linear SVM classifier with maximal margin for the toy example, and the dotted lines are 1 and -1 level sets of the discriminant function.

Although the formulation of the SVM discussed so far pertains to a linearly separable case only, which may be fairly restrictive, it serves as a prototype for its extension to more general cases with possible overlap between classes and nonlinear boundary. The extension to the nonseparable case will follow in the next section.

Two distinctive aspects of the formulation for the SVM are noted here. First, by targeting classification boundary, it takes direct aim at prediction of labels given attributes bypassing modeling or estimation of the probabilistic mechanism that generates the labels. In a decision theoretic view, the SVM is categorized as a procedure that directly minimizes the error rate under the 0-1 loss. Differently from data modeling strategy in statistics, this approach of risk minimization is commonly employed in machine learning for supervised-learning problems as encapsulated in the empirical risk minimization principle. Yet, as a result of error rate minimization, the SVM classifier is inevitably limited in inference on the underlying probability distribution, which is to be discussed later in detail. Second, although the margin may well be justified as a simple geometric notion to determine a unique separating hyperplane in the separable case, the rationale for a large margin is, in fact, deeply rooted in Vapnik's statistical learning theory (4). The theory shows that a notion of the capacity of a family of linear classifiers is inversely related to the margin size, and large margin classifiers can be expected to give lower test error rates. Clearly, maximizing the margin is a form of regularization akin to penalization of regression coefficients in ridge regression (13) in order to control model complexity for stability and accuracy in estimation.

2·2 Case with Overlapping Classes

When the training data are not linearly separable, the separability condition (Eq. 1) can not be met. To relax the condition, a set of non-negative variables ξ_i's are introduced for the data points such that

$$\xi_i + y_i(\beta' x_i + \beta_0) \geq 1, \qquad \xi_i \geq 0 \quad \text{for } i = 1, \ldots, n. \tag{3}$$

These ξ_i's are often called *slack variables* in the optimization literature as they loosen the rigid constraints. However, if they are too large, many data points could be incorrectly classified. If the ith data point is misclassified by the hyperplane $\beta' x + \beta_0 = 0$, that is, $y_i(\beta' x_i + \beta_0) \leq 0$, then $\xi_i \geq 1$. So, $\sum_{i=1}^{n} \xi_i$ provides an upper bound of the misclassification error of the classifier $\phi(x) = sgn(\beta' x + \beta_0)$. To minimize the error bound and at the same time to maximize the margin, the SVM formulation for the separable case is modified to seek β_0, β, and $\xi := (\xi_1, \ldots, \xi_n)'$ that minimize

$$\frac{1}{n} \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2} \|\beta\|^2 \tag{4}$$

subject to (Eq. 3). Here λ is a positive tuning parameter that controls the trade-off between the error bound and the margin.

By noting that given a constant a, $(\min \xi_i$ subject to $\xi_i \geq 0$ and $\xi_i \geq a) = \max\{a, 0\} := a_+$, it can be shown that the above modification is equivalent to finding β_0 and β that minimize

$$\frac{1}{n} \sum_{i=1}^{n} (1 - y_i(\beta' x_i + \beta_0))_+ + \frac{\lambda}{2} \|\beta\|^2. \tag{5}$$

This equivalent form brings a new loss function known as the *hinge loss* for measuring a 'goodness of fit' of a real-valued discriminant function. For a discriminant function $f(x) = \beta' x + \beta_0$, consider a loss criterion, $L(f(x_i), y_i) = (1 - y_i f(x_i))_+ = (1 - y_i(\beta' x_i + \beta_0))_+$. $y_i(\beta' x_i + \beta_0)$ is called the *functional margin* of the individual point (x_i, y_i) differently from the geometric class margin in the separable case. The functional margin of (x, y) is the product of a signed distance from x to the hyperplane $\beta' x + \beta_0 = 0$ and $\|\beta\|$. That is, if

$y(\beta' x + \beta_0) > 0$,

$$yf(x) = |\beta' x + \beta_0| = \|\beta\| \times \text{distance of } x \text{ from the hyperplane } \beta' x + \beta_0 = 0,$$

and otherwise, $yf(x) = -|\beta' x + \beta_0| = -\|\beta\| \times$ distance of x from the hyperplane.

Figure 2 shows the hinge loss together with the 0-1 loss (misclassification loss) in terms of the functional margin. For a discriminant function that induces a classifier through $sgn(f(x))$, the misclassification loss is given by

$$L_{0-1}(f(x), y) := I(y \neq sgn(f(x))) = I(yf(x) \leq 0).$$

Clearly, the hinge loss is a convex upper bound of the 0-1 loss and is monotonically decreasing in $yf(x) = y(\beta' x + \beta_0)$, the functional margin. The convexity of the hinge loss makes the SVM computationally more attractive than direct minimization of the empirical error rate.

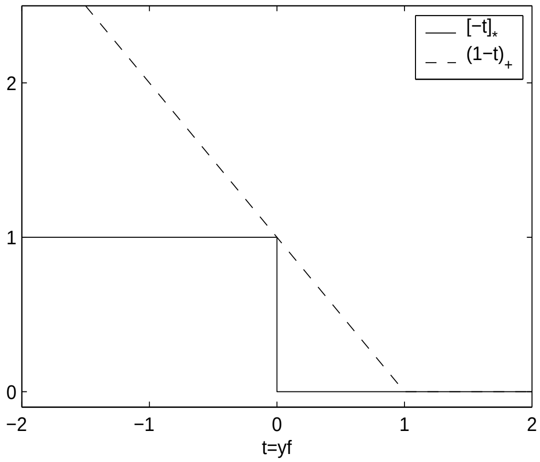

Figure 2: The solid line is the 0-1 loss, and the dashed line is the hinge loss in terms of the functional margin $yf(x)$.

In the case with overlapping classes, the geometric interpretation of $2/\|\beta\|$ as the separation margin between two classes no longer holds although $2/\|\beta\|$ may still be viewed as a 'soft' margin analogous to the 'hard' margin in the separable case. Rather, $\|\beta\|^2$ in (Eq. 5) can be immediately regarded as a penalty imposed on the linear discriminant function f.

From this perspective, the SVM procedure can be cast in the regularization framework where a function estimation method is formulated as an optimization problem of finding f in a class of candidate functions $\mathcal{F}$ that minimizes

$$\frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \lambda J(f).$$

Here $L(f(x), y)$ is a loss function, $J(f)$ is a regularizer or a penalty imposed on f, and $\lambda > 0$ is a tuning parameter which controls the trade-off between data fit and the complexity of f. There are numerous examples of regularization procedures in statistics. For instance, consider the multiple linear regression with $\mathcal{F} = \{f(x) = \beta' x + \beta_0 : \beta \in \mathbb{R}^p, \beta_0 \in \mathbb{R}\}$ and the squared error loss $(y - f(x))^2$ for L. $J(f) = \|\beta\|^2$ defines the ridge regression procedure in (13) while the least absolute shrinkage and selection operator (LASSO) in (14) takes $J(f) = \sum_{j=1}^{p} |\beta_j|$ as a penalty for a sparse linear model. In light of these, the SVM can be viewed as a procedure for penalized risk minimization with the hinge loss and ridge-like penalty.

2·3 Computation: Constrained Optimization

To describe the operational properties of the SVM solution, derivation of the dual optimization problem for the optimal hyperplane is sketched in this section. Thorough explanation of the theory behind is omitted for ease of discussion. Details and the relevant optimization theory can be found in (7, 10). To solve (Eq. 5) through the equivalent problem in (Eq. 4), we need to handle the inequality constraints in (Eq. 3). Using the standard machinery of primal-dual formulations in constrained optimization theory (15), two sets of Lagrange multipliers or dual variables are introduced for the constraints: α_i and γ_i $(i = 1, \ldots, n)$ for

$\xi_i \geq 1 - y_i(\beta' x_i + \beta_0)$ and $\xi_i \geq 0$, respectively. Define $h_i(\beta, \beta_0, \xi) = 1 - y_i(\beta' x_i + \beta_0) - \xi_i$ and $h_{n+i}(\beta, \beta_0, \xi) = -\xi_i$ so that the constraints are of the form $h_i(\beta, \beta_0, \xi) \leq 0$ for $i = 1, \ldots, 2n$. Then the Lagrangian primal function is given by

$$l_P(\beta, \beta_0, \xi, \alpha, \gamma) = \sum_{i=1}^{n} \xi_i + \frac{n\lambda}{2} \|\beta\|^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i(\beta' x_i + \beta_0) - \xi_i) - \sum_{i=1}^{n} \gamma_i \xi_i$$

with the following constraints

$$\begin{aligned}
\frac{\partial l_P}{\partial \beta} &= n\lambda\beta - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Leftrightarrow \beta = \frac{1}{n\lambda} \sum_{i=1}^{n} \alpha_i y_i x_i, \\
\frac{\partial l_P}{\partial \beta_0} &= -\sum_{i=1}^{n} \alpha_i y_i = 0 \Leftrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0, \\
\frac{\partial l_P}{\partial \xi_i} &= 1 - \alpha_i - \gamma_i = 0 \Leftrightarrow \gamma_i = 1 - \alpha_i, \\
& \quad \alpha_i \geq 0 \text{ and } \gamma_i \geq 0 \text{ for } i = 1, \ldots, n.
\end{aligned}$$

Simplifying l_P by using the constraints, we have the dual problem of maximizing

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2n\lambda} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i' x_j \text{ with respect to } \alpha := (\alpha_1, \ldots, \alpha_n)' \tag{6}$$

subject to $0 \leq \alpha_i \leq 1$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$ for $i = 1, \ldots, n$. Note that the dual problem is a quadratic programming (QP) problem with a non-negative definite matrix given by $\left[y_i y_j x_i' x_j \right]$.

The dual problem in (Eq. 6) itself reveals a few notable characteristics of the SVM. First, it involves n dual variables, so the sample size could be the main factor that determines the size of the problem not the number of predictors. This implies a great computational advantage when n is relatively small while p is very large. For example, typical microarray data have such a 'large p small n' structure. Second, the solution $\hat{\alpha}$ depends on the attributes in the training data only through their pairwise inner products $x_i'x_j$. This observation proves to be particularly useful for nonlinear extension of the linear SVM. Third, once $\hat{\alpha}$ satisfying the bound condition ($0 \le \hat{\alpha}_i \le 1$) and the equilibrium condition ($\sum_{i=1}^n \hat{\alpha}_i y_i = 0$) is found, the normal vector of the optimal hyperplane is determined by $\hat{\beta} = \frac{1}{n\lambda} \sum_{i=1}^n \hat{\alpha}_i y_i x_i$.

As part of necessary and sufficient conditions for the optimality of the solution, known as the *Karush-Kuhn-Tucker (KKT) conditions*, the following complementarity conditions have to be met: for $i = 1, \ldots, n$,

$$\begin{aligned} \alpha_i h_i(\beta, \beta_0, \xi) &= \alpha_i \{1 - y_i(\beta' x_i + \beta_0) - \xi_i\} = 0, \text{ and} \\ \gamma_i h_{n+i}(\beta, \beta_0, \xi) &= -\gamma_i \xi_i = -(1 - \alpha_i)\xi_i = 0. \end{aligned}$$

Since for any $0 < \alpha_{i^*} < 1$, $\xi_{i^*} = 0$ and $1 - y_{i^*}(\beta' x_{i^*} + \beta_0) - \xi_{i^*} = 0$ by the conditions, we have $1 - y_{i^*}(\beta' x_{i^*} + \beta_0) = 0$. This gives an equation for $\hat{\beta}_0$ once $\hat{\beta}$ is determined:

$$\hat{\beta}_0 = y_{i^*} - \hat{\beta}' x_{i^*} = y_{i^*} - \frac{1}{n\lambda} \sum_{i=1}^n \hat{\alpha}_i y_i x_i' x_{i^*}.$$

Also, by the complementarity conditions, the data points can be categorized into two kinds: those with a positive Lagrange multiplier ($\hat{\alpha}_i > 0$) and those with zero ($\hat{\alpha}_i = 0$). If a data point falls outside the margin, $y_i(\hat{\beta}' x_i + \hat{\beta}_0) > 1$, then the corresponding Lagrange multiplier must be $\hat{\alpha}_i = 0$, and thus it plays no role in determining $\hat{\beta}$. On the other hand, the attribute vectors of the data points with $\hat{\alpha}_i > 0$ expand $\hat{\beta}$, and such data points are called the *support vectors*. The proportion of the support vectors depends on λ, but typically for a range of values of λ, only a fraction of the data points are support vectors. In the sense, the SVM solution admits a *sparse* expression in terms of the data points. This sparsity is due to the singularity of the hinge loss at 1. A simple analogy of the sparsity can be made to median regression with the absolute deviation loss that has a singular point at 0.

For classification of a new point x, the following linear discriminant function is used:

$$\hat{f}_\lambda(x) = \hat{\beta}'x + \hat{\beta}_0 = \frac{1}{n\lambda} \sum_{i:\ \hat{\alpha}_i > 0} \hat{\alpha}_i y_i x_i' x + \hat{\beta}_0. \tag{7}$$

Note that the final form of $\hat{f}_\lambda$ does not depend on the dimensionality of x explicitly but depends on the inner products of x_i and x as in the dual problem. This fact enables construction of hyperplanes even in infinite-dimensional Hilbert spaces (p.406, (4)). In addition, (Eq. 7) shows that all the information necessary for discrimination is contained in the support vectors. As a consequence, it affords efficient data reduction and fast evaluation at the testing phase.

2·4 Nonlinear Generalization

In general, hyperplanes in the input space may not be sufficiently flexible to attain the smallest possible error rate for a given problem. As noted earlier, the linear SVM solution and prediction of a new case x depends on the x_i's only through the inner product $x_i'x_j$ and $x_i'x$. This fact leads to a straightforward generalization of the linear SVM to the nonlinear case by taking a basis expansion. The main idea of the nonlinear extension is to map the data in the original input space to a feature space and find the hyperplane with a large margin in the feature space. For an enlarged feature space, consider transformations of x, say, $\phi_m(x)$, $m = 1, \ldots, M$. Let $\Phi(x) := (\phi_1(x), \ldots, \phi_M(x))'$ be the so-called *feature mapping* from $\mathbb{R}^p$ to a higher dimensional feature space, which can be even infinite dimensional. Then by replacing the inner product $x_i'x_j$ with $\Phi(x_i)'\Phi(x_j)$, the formulation of the linear SVM can be easily extended. For instance, suppose the input space is $\mathbb{R}^2$ and $x = (x_1, x_2)'$. Define $\Phi : \mathbb{R}^2 \to \mathbb{R}^3$ as $\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)'$. Then the mapping gives a new dot product in the feature space,

$$\Phi(x)'\Phi(t) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)(t_1^2, t_2^2, \sqrt{2}t_1t_2)' = (x_1t_1 + x_2t_2)^2 = (x't)^2.$$

In fact, for this generalization to work, the feature mapping Φ does not need to be explicit. Specification of the bivariate function $K(x,t) := \Phi(x)'\Phi(t)$ would suffice. With K, the nonlinear discriminant function is then given as $\hat{f}_\lambda(x) = \frac{1}{n\lambda}\sum_{i=1}^{n}\hat{\alpha}_i y_i K(x_i, x) + \hat{\beta}_0$, which is in the span of $K(x_i, x)$, $i = 1, \ldots, n$. So, the shape of the classification boundary, $\{x \in \mathbb{R}^p : \hat{f}_\lambda(x) = 0\}$ is determined by K.

From the property of the dot product, it is clear that such a bivariate function is non-negative definite. Replacing the Euclidean inner product in a linear method with a non-negative definite bivariate function, $K(x,t)$, known as a *kernel* function to obtain its non-linear generalization is often referred to as the 'kernel trick' in machine learning. The only condition for a kernel to be valid is that it is a symmetric non-negative (semi-positive) definite function: for every $N \in \mathbb{N}$, $a_i \in \mathbb{R}$, and $z_i \in \mathbb{R}^p$ $(i = 1, \ldots, N)$, $\sum_{i,j}^{N} a_i a_j K(z_i, z_j) \geq 0$. In other words, $K_N := [K(z_i, z_j)]$ is a non-negative definite matrix. Some kernels in common use are polynomial kernels with dth degree, $K(x,t) = (1 + x't)^d$ or $(x't)^d$ for some positive integer d and the radial basis (or Gaussian) kernel, $K(x,t) = \exp(-\|x - t\|^2/2\sigma^2)$ for $\sigma > 0$.

It turns out that this generalization of the linear SVM is closely linked to the function estimation procedure known as the reproducing kernel Hilbert space (RKHS) method in statistics (16, 17). And the theory behind the RKHS methods or kernel methods in short provides a unified view of smoothing splines, a classical example of the RKHS methods for nonparametric regression, and the kernelized SVM. The connection allows more abstract treatment of the SVM, offering a different perspective on the methodology, in particular, the nonlinear extension.

2·5 Kernel Methods

Kernel methods can be viewed as a method of regularization in a function space characterized by a kernel. A brief description of general framework for the regularization method is given here for advanced readers in order to elucidate the connection, to show how seamlessly the SVM sits in the framework, and to broaden the scope of its applicability in a wide range of problems.

Consider a Hilbert space (complete inner product space) of real-valued functions defined on a domain $\mathcal{X}$ (not necessarily $\mathbb{R}^p$), $\mathcal{H}$ with an inner product $\langle f, g\rangle_{\mathcal{H}}$ for $f, g \in \mathcal{H}$. A Hilbert space is an RKHS if there is a kernel function (called reproducing kernel) $K(\cdot,\cdot) : \mathcal{X}^2 \to \mathbb{R}$ such that

i) $K(x,\cdot) \in \mathcal{H}$ for every $x \in \mathcal{X}$, and

ii) $\langle K(x,\cdot), f(\cdot)\rangle_{\mathcal{H}} = f(x)$ for every $f \in \mathcal{H}$ and $x \in \mathcal{X}$.

The second condition is called the *reproducing* property for the obvious reason that K reproduces every f in $\mathcal{H}$. Let $K_x(t) := K(x,t)$ for a fixed x. Then the reproducing property gives a useful identity that $K(x,t) = \langle K_x(\cdot), K_t(\cdot)\rangle_{\mathcal{H}}$. For a comprehensive treatment of the RKHS, see (18). Consequently, reproducing kernels are non-negative definite. Conversely, by the Moore-Aronszajn Theorem, for every non-negative definite function $K(x,t)$ on $\mathcal{X}$, there corresponds a unique RKHS $\mathcal{H}_K$ that has $K(x,t)$ as its reproducing kernel. So, non-negative definiteness is the defining property of kernels.

Now, consider a regularization method in the RKHS, $\mathcal{H}_K$ with reproducing kernel K:

$$\min_{f\in\{1\}\oplus\mathcal{H}_K} \frac{1}{n}\sum_{i=1}^{n} L(f(x_i), y_i) + \lambda \|h\|^2_{\mathcal{H}_K}, \tag{8}$$

where $f(x) = \beta_0 + h(x)$ with $h \in \mathcal{H}_K$ and the penalty $J(f)$ is given by $\|h\|^2_{\mathcal{H}_K}$. In general, the null space can be extended to a larger linear space than $\{1\}$. As an example, $\mathcal{X} = \mathbb{R}^p$

and $\mathcal{H}_K = \{h(x) = \beta' x \mid \beta \in \mathbb{R}^p\}$ with $K(x,t) = x't$. For $h_1(x) = \beta_1' x$ and $h_2(x) = \beta_2' x \in \mathcal{H}$, $\langle h_1, h_2 \rangle_{\mathcal{H}_K} = \beta_1' \beta_2$. Then for $h(x) = \beta' x$, $\|h\|^2_{\mathcal{H}_K} = \|\beta' x\|^2_{\mathcal{H}_K} = \|\beta\|^2$. Taking $f(x) = \beta_0 + \beta' x$ and the hinge loss $L(f(x), y) = (1 - yf(x))_+$ gives the linear SVM as a regularization method in $\mathcal{H}_K$. So, encompassing the linear SVM as a special case, the SVM can be cast as a regularization method in an RKHS $\mathcal{H}_K$, which finds $f(x) = \beta_0 + h(x) \in \{1\} \oplus \mathcal{H}_K$ minimizing

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i f(x_i))_+ + \lambda \|h\|^2_{\mathcal{H}_K}. \tag{9}$$

The *representer* theorem in (19) says that the minimizer of (Eq. 8) has a representation of the form

$$\hat{f}_\lambda(x) = b + \sum_{i=1}^{n} c_i K(x_i, x), \tag{10}$$

where b and $c_i \in \mathbb{R}$, $i = 1, \ldots, n$. As previously mentioned, the kernel trick leads to the expression of the SVM solution:

$$\hat{f}_\lambda(x) = \hat{\beta}_0 + \frac{1}{n\lambda}\sum_{i=1}^{n} \hat{\alpha}_i y_i K(x_i, x).$$

It agrees with what the representer theorem generally implies for the SVM formulation. Finally, the solution in (Eq. 10) can be determined by minimizing

$$\frac{1}{n}\sum_{i=1}^{n}\{1 - y_i(b + \sum_{j=1}^{n} c_j K(x_j, x_i))\}_+ + \lambda \sum_{i,j=1}^{n} c_i c_j K(x_i, x_j)$$

over b and c_i's. For further discussion of the perspective, see (17).

Notably the abstract formulation of kernel methods has no restriction on input domains and the form of kernel functions. Because of that, the SVM in combination with a variety of kernels is modular and flexible. For instance, kernels can be defined on non-numerical domains such as strings of DNA bases, text, and graph, expanding the realm of applications well beyond Euclidean vector spaces. Many applications of the SVM in computational biology capitalize on the versatility of the kernel method. See (20) for examples.

2·6 Statistical Properties

Contrasting the SVM with more traditional approaches to classification, we discuss statistical properties of the SVM and their implications. Theoretically, the 0-1 loss criterion defines the rule that minimizes the error rate over the population as optimal. With the symmetric labeling of ±1 and conditional probability $\eta(x) := P(Y = 1|X = x)$, the optimal rule, namely, the Bayes decision rule is given by $\phi_B(x) := sgn(\eta(x) - 1/2)$, predicting the label of the most likely class. In the absence of the knowledge of $\eta(x)$, there are two different approaches for building a classification rule that emulates the Bayes classifier. One is to construct a probability model for the data first and then use the estimate of $\eta(x)$ from the model for classification. This yields such model-based plug-in rules as logistic regression, LDA, QDA and other density based classification methods. The other is to aim at direct minimization of the error rate without estimating $\eta(x)$ explicitly. Large margin classifiers with a convex surrogate of the 0-1 loss fall into the second type, and the SVM with the hinge loss is a typical example.

The discrepancy of the 0-1 loss from the surrogate loss that is actually used for training a classifier in the latter approach generated an array of theoretical questions regarding necessary conditions for the surrogate loss to guarantee the Bayes risk consistency of the resulting rules. (21–23) delve into the issues and provide proper conditions for a convex surrogate loss. It turns out that only minimal conditions are necessary in the binary case to ensure the risk consistency while much care has to be taken in the multiclass case (24–26). In particular, it is shown that the hinge loss is class-calibrated, meaning that it satisfies a weak notion of consistency known as Fisher consistency. Furthermore, the Bayes risk consistency of the

SVM has been established under the assumption that the space generated by a kernel is sufficiently rich (21, 27).

A simple way to see the effect of each loss criterion on the resulting rule is to look at its population analog and identify the limiting discriminant function which is defined as the population risk minimizer among measurable functions of f. Interestingly, for the hinge loss, the population minimizer of $E(1 - Yf(X))_+$ is $f^*_{SVM}(x) := sgn(\eta(x) - 1/2)$, the Bayes classifier itself, while that of the negative log likelihood loss for logistic regression is $f^*_{LR}(x) := \log\{\eta(x)/(1 - \eta(x))\}$, the true logit, for comparison. This difference is illustrated in Figure 3. For 300 equally spaced x_i in $(-2, 2)$, y_i's were generated with the probability of class 1 equal to $\eta(x)$ in Figure 3 (the solid line is $2\eta(x) - 1$). The dotted line is the estimate of $2\eta(x) - 1$ by penalized logistic regression and the dashed line is the SVM. The radial basis kernel was used for both methods. Note that the logistic regression estimate is very close to the true probability $2\eta(x) - 1$ while the SVM is close to $sgn(\eta(x) - 1/2)$. Nonetheless, the resulting classifiers are almost identical.

If prediction is of primary concern, then the SVM can be an effective choice. However, there are many applications where accurate estimation of the conditional probability $\eta(x)$ is required for making better decisions than just prediction of a dichotomous outcome. In those cases, the SVM offers very limited information as there is no principled way to recover the probability from the SVM output in general.

However, the remark pertains only to the SVM with a flexible kernel since it is based on the property that the asymptotic discriminant function is $sgn(\eta(x) - 1/2)$. The SVM with simple kernels, the linear SVM for one, needs to be analyzed separately. A recent study (28) shows that under the normality and equal variance assumption on the distribution of attributes for two classes, the linear SVM coincides with the LDA in the limit. Technically, the analysis exploits a close link between the SVM and median regression yet with categorical responses. At least in this case, the probability information would not be masked and can be

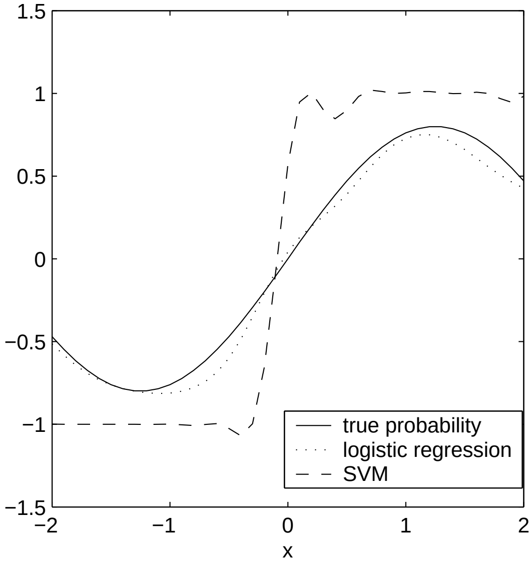

Figure 3: Comparison of the SVM and logistic regression. The solid line is the true function, $2\eta(x) - 1$, the dotted line is $2\hat{\eta}_{LR}(x) - 1$ from penalized logistic regression, and the dashed line is $\hat{f}_{SVM}(x)$ of the SVM.

recovered from the linear discriminant function with additional computation. However, it is generally advised that the SVM is a tool for prediction, not for modeling of the probabilistic mechanism underlying the data.

3 Data Example

Taking breast cancer data in (29) as an example, we illustrate the method and discuss various aspects of its application and some practical issues. The data consist of expression levels of 24,481 genes collected from patients with primary breast tumors who were lymph node negative at the time of diagnosis. The main goal of the study was to find a gene expression signature prognostic of distant metastases within five years, which can be used to select patients who would benefit from adjuvant therapy such as chemotherapy or hormone therapy. Out of 78 patients in the training data, 34 developed metastasis within five years (labeled a poor prognosis) and 44 remained metastasis free for at least five years (labeled a good prognosis).

Following a similar preprocessing step in the paper, we filtered those genes that exhibited at least a two-fold change in the expression from the pooled reference sample with a p-value < 0.01 in five or more tumors in the training data and discarded two additional genes with missing values, yielding 4,008 genes. Sample 54 with more than 20% of missing values was removed before filtering.

First, we applied the linear SVM to the training data (77 observations), varying the number of genes from large to relatively small ($d = 4008$, 70, and 20) to see the effect of the input dimension on error rates and the number of support vectors. Whenever a subset of genes were used, we included in classification those top ranked genes by the p-value of a t-test statistic for marginal association with the prognostic outcomes. 70 is the number of genes selected for the prediction algorithm in the original paper although the selection procedure was not based on the p-values.

λ affects classification accuracy and the number of support vectors as well. To elucidate the effect of λ, we obtained all the possible solutions indexed by the tuning parameter λ for each fixed set of genes (using R package `svmpath`).

Figure 4 shows the error rate curves as a function of λ. The dotted lines are the apparent error rates of the linear SVM over the training data set itself, and the solid lines are the test error rates evaluated over the 19 test patients, where 7 remained metastasis free for at least five years and 12 developed metastasis within five years. Clearly, when all of 4,008 genes are included in the classifier, the training error rates can be driven to zero as the λ decreases to zero, that is, classifiers get less regularized. On the other hand, the corresponding test error rates in the same panel for small values of λ are considerably higher than the training error rates, exemplifying the well-known phenomenon of overfitting. Hence, to attain the best test error rate, the training error rate and the complexity of a classifier need to be

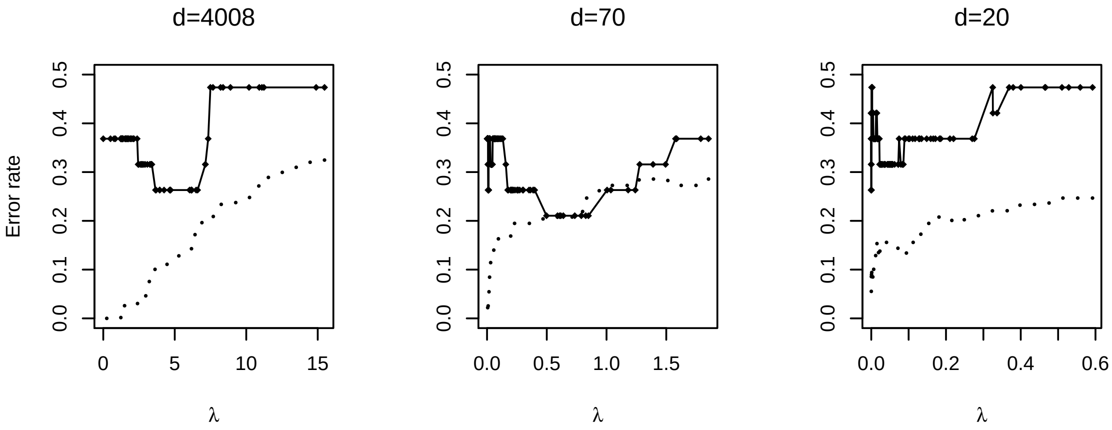

Figure 4: Error rate curves of the linear SVMs with three input dimensions (left: 4,008, center: 70, and right: 20). The dotted lines are the apparent error rates over 77 training patients, and the solid lines are the test error rates over 19 test patients.

properly balanced. However, for smaller input dimensions, the relationship between the apparent error rates and test error rates is quite different. In particular, when only 20 genes are used, in other words, the feature space is small, regularization provides little benefit in minimizing the test error rate and the two error rate curves are roughly parallel to each other. In contrast, for $d = 70$ and $d = 4,008$, penalization ('maximizing the margin') does help in reducing the test error rate. The overall minimum error rate of around 20% was achieved when $d = 70$.

Like the error rates, the number of support vectors also depends on the tuning parameter, the degree of overlap between two classes and the input dimensionality among other factors. Figure 5 depicts how it varies as a function of λ for the three cases of high to relatively low dimension. When $d = 4,008$, the number of support vectors is approximately constant and except a few observations almost all the observations are support vectors. A likely reason is that the dimension is so high compared to the sample size that nearly every observation is close to the classification boundary. However, for the lower dimensions, as the λ approaches

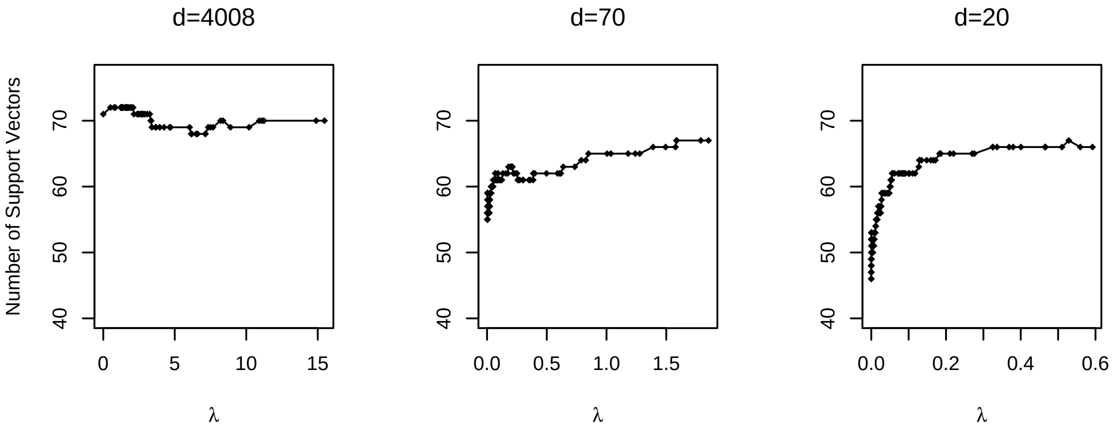

Figure 5: Relationship between the input dimension (left: 4,008, center: 70, and right: 20) and the number of support vectors for the linear SVM.

zero, a smaller fraction of observations come out to be support vectors.

Generally, changing the kernel from linear to nonlinear leads to reduction in the overall training error rate, and it often translates into a lower test error rate. As an example, we obtained the training and test error rate curves for the Gaussian kernel, $K(x,t) = \exp(-\|x-t\|^2/2\sigma^2)$, with the 70 genes as shown in Figure 6. The bandwidth σ, which is another tuning parameter, was set to be the median of pairwise distances between two classes in the left panel, its half in the center, and nearly a third of the median in the right panel, respectively. Figure 6 illustrates that with a smaller bandwidth, the training error rates can be made substantially small over a range of λ. Moreover, for $\sigma = 1.69$ and 1.20, if λ is properly chosen, then fewer mistakes are made in prediction for the test cases by the nonlinear SVM than the linear SVM.

As emphasized before, generally the SVM output values can not be mapped to class-conditional probabilities in a theoretically justifiable way perhaps with the only exception of the linear SVM in a limited situation. For comparison of logistic regression and SVM, we applied penalized logistic regression to the breast cancer data with the expression levels

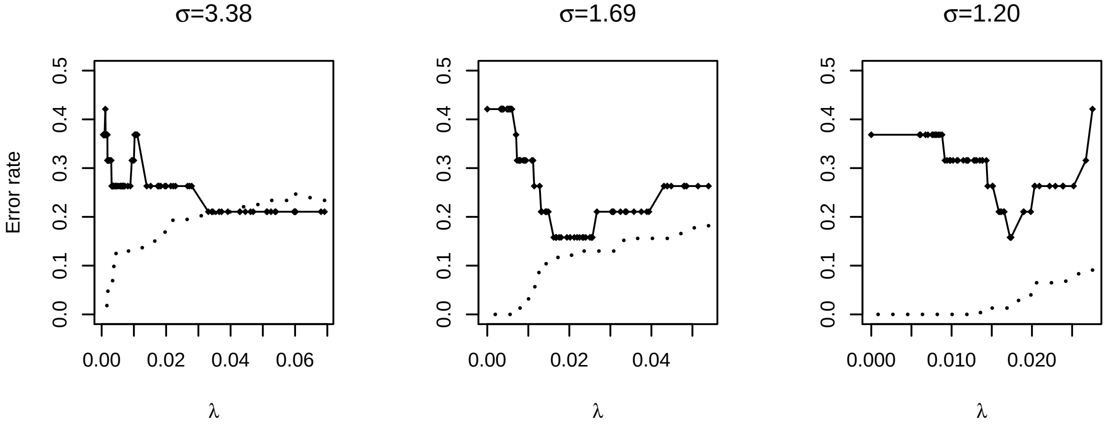

Figure 6: Error rate curves of the nonlinear SVM with 70 genes and the Gaussian kernel for three bandwidths. The dotted lines are the apparent error rates, and the solid lines are the test error rates.

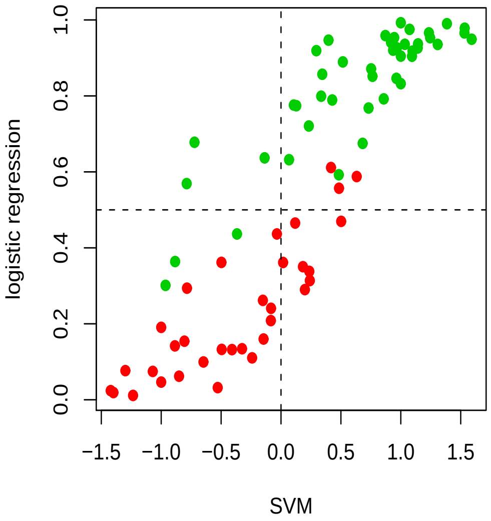

Figure 7: Scatter plot of the estimated probabilities of good prognosis from penalized logistic regression versus the values of the discriminant function from the linear SVM for training data. The green dots indicate the patients with good diagnosis and the red dots indicate those with poor diagnosis.

of the 70 genes as linear predictors. For simplicity, the optimal penalty size for logistic regression was again determined by the minimum test error rate. Figure 7 is a plot of the estimated probabilities of good prognosis from the logistic regression versus the values of the discriminant function from the linear SVM evaluated for the observations in the training data. It shows a monotonic relationship between the output values of the two methods, which could be used for calibration of the results from the SVM with class-conditional probabilities. When each method was best tuned in terms of the test error rate, logistic regression gave 10% of the training error rate and 30% of the test error rate while both error rates were around 20% for the SVM. For more comparison between the two approaches, see (30, 31).

The statistical issue of finding an optimal choice of the tuning parameter has not been discussed adequately in this data example. Instead, by treating the test set as if it were a validation set, the size of the penalty was chosen to minimize the test error rate directly for simple exposition. In practice, cross validation is commonly used for tuning in the absence of a separate validation set.

On a brief note, in the original paper, a correlation-based classifier was constructed on the basis of 70 genes that were selected sequentially and its threshold was adjusted for increased sensitivity to poor prognosis. With the adjusted threshold, only 2 out of 19 incorrect predictions were reported. This low test error rate could be explained as a result of the threshold adjustment. Recall that the good prognosis category is the majority for the training data set (good/poor=44/33) while the opposite is true for the test set (good/poor=7/12). As in this example, if two types of error (misclassifying good prognosis as poor or vice versa) are treated differentially, then the optimal decision boundary would be different from the region where two classes are equally likely, that is, $\eta(x) = 1/2$. For estimation of a different level of probability, say, $\eta_0 \neq 1/2$ with the SVM method, the hinge loss has to be modified with weights that are determined according to the class labels. This modification leads to a weighted SVM, and more details can be found in (32).

4 Further Extensions

So far, the standard SVM for the binary case has been mainly introduced. Since its inception, various methodological extensions have been considered, expanding its utility to many different settings and applications. Just to provide appropriate pointers to references for further reading, some of the extensions are briefly mentioned here.

First, consider situations that involve more than two classes. A proper extension of the binary SVM to the multiclass case is not as straightforward as a probability-model based approach to classification, as evident in the special nature of the discriminant function that minimizes the hinge loss in the binary case. (24, 25) discuss some extensions of the hinge loss that would carry the desired consistency of the binary SVM to the multicategory case.

Second, identification of the variables that discriminate given class labels is often crucial in many applications. There have been a variety of proposals to either combine or integrate variable or feature selection capability with the SVM for enhanced interpretability. For example, recursive feature elimination (33) combines the idea of backward elimination with the linear SVM. Similar to the ℓ_1 penalization approach to variable selection in regression such as the LASSO and the basis pursuit method (34), (35) and later (36) modified the linear SVM with the ℓ_1 penalty for feature selection, and (37) considered further the ℓ_0 penalty. For a nonlinear kernel function, (38, 39) introduced a scale factor for each variable and chose the scale factors by minimizing generalization error bounds. As an alternative, (40, 41) suggested functional analysis of variance approach to feature selection for the nonlinear SVM motivated by the nonparametric generalization of the LASSO in (42).

On the computational front, numerous algorithms to solve the SVM optimization problem have been developed for fast computation with enhanced algorithmic efficiency and for the capacity to cope with massive data. (43) provides a historical perspective of the development in terms of relevant computational issues to the SVM optimization. Some of the implementations are available at http://www.kernel-machines.org, including SVM light in (44) and LIBSVM in (45). The R package `e1071` is an R interface to LIBSVM, and `kernlab` is another R implementation of the SVM. Note that the aforementioned implementations are mostly for getting a solution at a given value of the tuning parameter λ. However, as seen in the data example, the classification error rate depends on λ, and thus, in practice, it is necessary to consider a range of λ values and get the corresponding solutions in pursuit of an optimal solution. It turns out that characterization of the entire solution path as a function of λ is possible as demonstrated in (46) for the binary case and (47) for the multicategory case. The solution path algorithms in the references provide a computational shortcut to obtain the entire spectrum of solutions, facilitating the choice of the tuning parameter.

The scope of extensions of kernel methods in current use is, in fact, far beyond classification. Details of other methodological developments with kernels for regression, novelty detection, clustering, and semi-supervised learning can be found in (7).

References

[1] Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning.* Springer Verlag, New York.

[2] Duda, R. O., Hart, P. E., and Stork, D. G. (2000) *Pattern Classification (2nd Edition).* Wiley-Interscience, New York.

[3] McLachlan, G. J. (2004) *Discriminant Analysis and Statistical Pattern Recognition.* Wiley-Interscience, New York.

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

[5] Boser, B., Guyon, I., and Vapnik, V. (1992) A training algorithm for optimal margin

classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* 5, 144–152.

[6] Cristianini, N., and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines.* Cambridge University Press.

[7] Schölkopf, B., and Smola, A. (2002) *Learning with Kernels - Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge, MA: MIT Press.

[8] Cortes, C., and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning* 20(3), 273–297.

[9] Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386–408.

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

[11] Bennett, K. P., and Campbell, C. (2000) Support vector machines: Hype or hallelujah? *SIGKDD Explorations* 2(2), 1–13.

[12] Moguerza, J. M., and Munoz, A. (2006) Support vector machines with applications. *Statistical Science* 21(3), 322–336.

[13] Hoerl, A., and Kennard, R. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12(3), 55–67.

[14] Tibshirani, R. (1996) Regression selection and shrinkage via the lasso. *Journal of the Royal Statistical Society* B 58(1), 267–88.

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

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

[17] Wahba, G. (1998) Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV. In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods: Support Vector Learning* pages 69–87. MIT Press.

[18] Aronszajn, N. (1950) Theory of reproducing kernel. *Transactions of the American Mathematical Society* 68, 3337–404.

[19] Kimeldorf, G., and Wahba, G. (1971) Some results on Tchebycheffian Spline functions. *Journal of Mathematics Analysis and Applications* 33(1), 82–95.

[20] Schölkopf, B., Tsuda, K., and Vert, J. P., editors (2004) *Kernel Methods in Computational Biology*. MIT Press Cambridge, MA.

[21] Zhang, T. (2004) Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics* 32(1), 56–85.

[22] Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006) Convexity, classification, and risk bounds. *Journal of the American Statistical Association* 101, 138–156.

[23] Lin, Y. (2002) A note on margin-based loss functions in classification. *Statistics and Probability Letters* 68, 73–82.

[24] 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. *Journal of the American Statistical Association* 99, 67–81.

[25] Tewari, A., and Bartlett, P. L. (2007) On the consistency of multiclass classification methods. *Journal of Machine Learning Research* 8, 1007–1025.

[26] Liu, Y., and Shen, X. (2006) Multicategory SVM and ψ-learning-methodology and theory. *Journal of the American Statistical Association* 101, 500–509.

[27] Steinwart, I. (2005) Consistency of support vector machines and other regularized kernel machines. *IEEE Transactions on Information Theory* 51, 128–142.

[28] Koo, J.-Y., Lee, Y., Kim, Y., and Park, C. (2008) A Bahadur representation of the linear Support Vector Machine. *Journal of Machine Learning Research* 9, 1343–1368.

[29] van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415(6871), 530–536.

[30] Zhu, J., and Hastie, T. (2004) Classification of gene microarrays by penalized logistic regression. *Biostatistics* 5(3), 427–443.

[31] Wahba, G. (2002) Soft and hard classification by reproducing kernel Hilbert space methods. *Proceedings of the National Academy of Sciences* 99, 16524–16530.

[32] Lin, Y., Lee, Y., and Wahba, G. (2002) Support vector machines for classification in nonstandard situations. *Machine Learning* 46, 191–202.

[33] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002) Gene selection for cancer classification using support vector machines. *Machine Learning* 46(1-3), 389–422.

[34] Chen, S. S., Donoho, D. L., and Saunders, M. A. (1999) Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing* 20(1), 33–61.

[35] Bradley, P. S., and Mangasarian, O. L. (1998) Feature selection via concave minimization and support vector machines. In Shavlik, J., editor, *Machine Learning Proceedings*

of the Fifteenth International Conference pages 82–90. San Francisco, California. Morgan Kaufmann.

[36] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2004) 1-norm support vector machines. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. Cambridge, MA. MIT Press.

[37] Weston, J., Elisseff, A., Schölkopf, B., and Tipping, M. (2003) Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research* 3, 1439–61.

[38] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2001) Feature selection for SVMs. In Solla, S. A., Leen, T. K., and Muller, K.-R., editors, *Advances in Neural Information Processing Systems* 13, 668–674. Cambridge, MA. MIT Press.

[39] Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002) Choosing multiple parameters for support vector machines. *Machine Learning* 46(1-3), 131–59.

[40] Zhang, H. H. (2006) Variable selection for support vector machines via smoothing spline ANOVA. *Statistica Sinica* 16(2), 659–674.

[41] Lee, Y., Kim, Y., Lee, S., and Koo, J.-Y. (2006) Structured Multicategory Support Vector Machine with ANOVA decomposition. *Biometrika* 93(3), 555–571.

[42] Lin, Y., and Zhang, H. H. (2006) Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics* 34, 2272–2297.

[43] Bottou, L., and Lin, C.-J. (2007) Support Vector Machine Solvers. In Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., editors, *Large Scale Kernel Machines* pages 301–320. MIT Press Cambridge, MA.

[44] Joachims, T. (1998) Making large-scale support vector machine learning practical. In Schölkopf, C. B., editor, *Advances in Kernel Methods: Support Vector Machines.* MIT Press, Cambridge, MA.

[45] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008) LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9, 1871–1874.

[46] Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004) The entire regularization path for the support vector machine. *Journal of Machine Learning Research* 5, 1391–1415.

[47] Lee, Y., and Cui, Z. (2006) Characterizing the solution path of Multicategory Support Vector Machines. *Statistica Sinica* 16(2), 391–409.