

# Support Vector Machines with Applications<sup>1</sup>

Javier M. Moguerza and Alberto Muñoz

*Abstract.* Support vector machines (SVMs) appeared in the early nineties as optimal margin classifiers in the context of Vapnik’s statistical learning theory. Since then SVMs have been successfully applied to real-world data analysis problems, often providing improved results compared with other techniques. The SVMs operate within the framework of regularization theory by minimizing an empirical risk in a well-posed and consistent way. A clear advantage of the support vector approach is that sparse solutions to classification and regression problems are usually obtained: only a few samples are involved in the determination of the classification or regression functions. This fact facilitates the application of SVMs to problems that involve a large amount of data, such as text processing and bioinformatics tasks. This paper is intended as an introduction to SVMs and their applications, emphasizing their key features. In addition, some algorithmic extensions and illustrative real-world applications of SVMs are shown.

*Key words and phrases:* Support vector machines, kernel methods, regularization theory, classification, inverse problems.

## 1. INTRODUCTION

In the last decade, support vector machines (SVMs) have increasingly turned into a standard methodology in the computer science and engineering communities. As Breiman [12] pointed out, these communities are often involved in the solution of consulting and industrial data analysis problems. The usual starting point is a sample data set  $\{(\mathbf{x}_i, \mathbf{y}_i) \in X \times Y\}_{i=1}^n$ , and the goal is to “learn” the relationship between the  $\mathbf{x}$  and  $\mathbf{y}$  variables. The variable  $X$  may be, for instance, the space of  $20 \times 20$  binary matrices that represent alphabetic uppercase characters and  $Y$  would be the label set  $\{1, \dots, 27\}$ . Similarly,  $X$  may be  $\mathbb{R}^{10,000}$ , the space

corresponding to a document data base with a vocabulary of 10,000 different words. In this case  $Y$  would be the set made up of a finite number of predefined semantic document classes, such as statistics, computer science, sociology and so forth. The main goal in this context usually is predictive accuracy, and in most cases it is not possible to assume a parametric form for the probability distribution  $p(\mathbf{x}, \mathbf{y})$ . Within this setting many practitioners concerned with providing practical solutions to industrial data analysis problems put more emphasis on algorithmic modeling than on data models. However, a solely algorithmic point of view can lead to procedures with a black box behavior, or even worse, with a poor response to the bias–variance dilemma. Neural networks constitute a paradigmatic example of this approach. The (semiparametric) model implemented by neural networks is powerful enough to approximate continuous functions with arbitrary precision. On the other hand, neural network parameters are very hard to tune and interpret, and statistical inference is usually not possible [51].

The SVMs provide a compromise between the parametric and the pure nonparametric approaches: As in linear classifiers, SVMs estimate a linear decision

---

Javier M. Moguerza is Associate Professor, School of Engineering, University Rey Juan Carlos, c/ Tulipan s/n, 28933 Mostoles, Spain (e-mail: javier.moguerza@urjc.es).  
Alberto Muñoz is Associate Professor, Department of Statistics, University Carlos III, c/Madrid 126, 28903 Getafe, Spain (e-mail: alberto.munoz@uc3m.es).

<sup>1</sup>Discussed in 10.1214/088342306000000457,  
10.1214/088342306000000466, 10.1214/088342306000000475  
and 10.1214/088342306000000484; rejoinder  
10.1214/088342306000000501.

function, with the particularity that a previous mapping of the data into a higher-dimensional feature space may be needed. This mapping is characterized by the choice of a class of functions known as kernels. The support vector method was introduced by Boser, Guyon and Vapnik [10] at the Computational Learning Theory (COLT92) ACM Conference. Their proposal subsumed into an elegant and theoretically well founded algorithm two seminal ideas, which had already individually appeared throughout previous years: the use of kernels and their geometrical interpretation, as introduced by Aizerman, Braverman and Rozonoer [1], and the idea of constructing an optimal separating hyperplane in a nonparametric context, developed by Vapnik and Chervonenkis [78] and by Cover [16]. The name “support vector” was explicitly used for the first time by Cortes and Vapnik [15]. In recent years, several books and tutorials on SVMs have appeared. A reference with many historical annotations is the book by Cristianini and Shawe-Taylor [20]. For a review of SVMs from a purely geometrical point of view, the paper by Bennett and Campbell [9] is advisable. An exposition of kernel methods with a Bayesian taste can be read in the book by Herbrich [30]. Concerning the statistical literature, the book by Hastie, Tibshirani and Friedman [28] includes a chapter dedicated to SVMs.

We illustrate the basic ideas of SVMs for the two-group classification problem. This is the typical version and the one that best summarizes the ideas that underlie SVMs. The issue of discriminating more than two groups can be consulted, for instance, in [37].

Consider a classification problem where the discriminant function is nonlinear, as illustrated in Figure 1(a). Suppose we have a mapping  $\Phi$  into a “feature space” such that the data under consideration have become linearly separable as illustrated in Figure 1(b).

From the infinite number of existing separating hyperplanes, the support vector machine looks for the plane that lies furthestmost from both classes, known as the optimal (maximal) margin hyperplane. To be more specific, denote the available mapped sample by  $\{(\Phi(\mathbf{x}_i), y_i)\}_{i=1}^n$ , where  $y_i \in \{-1, +1\}$  indicates the two possible classes. Denote by  $\mathbf{w}^T \Phi(\mathbf{x}) + b = 0$  any separating hyperplane in the space of the mapped data equidistant to the nearest point in each class. Under the assumption of separability, we can rescale  $\mathbf{w}$  and  $b$  so that  $|\mathbf{w}^T \Phi(\mathbf{x}) + b| = 1$  for those points in each class nearest to the hyperplane. Therefore, it holds that for every  $i \in \{1, \dots, n\}$ ,

$$(1.1) \quad \mathbf{w}^T \Phi(\mathbf{x}_i) + b \begin{cases} \geq 1, & \text{if } y_i = +1 \\ \leq -1, & \text{if } y_i = -1. \end{cases}$$

After the rescaling, the distance from the nearest point in each class to the hyperplane is  $1/\|\mathbf{w}\|$ . Hence, the distance between the two groups is  $2/\|\mathbf{w}\|$ , which is called the margin. To maximize the margin, the following optimization problem has to be solved:

$$(1.2) \quad \begin{aligned} \min_{\mathbf{w}, b} \quad & \|\mathbf{w}\|^2 \\ \text{subject to (s.t.)} \quad & y_i(\mathbf{w}^T \Phi(\mathbf{x}_i) + b) \geq 1, \\ & i = 1, \dots, n, \end{aligned}$$

where the square in the norm of  $\mathbf{w}$  has been introduced to make the problem quadratic. Notice that, given its convexity, this optimization problem has no local minima. Consider the solution of problem (1.2), and denote it by  $\mathbf{w}^*$  and  $b^*$ . This solution determines the hyperplane in the feature space  $D^*(\mathbf{x}) = (\mathbf{w}^*)^T \Phi(\mathbf{x}) + b^* = 0$ . Points  $\Phi(\mathbf{x}_i)$  that satisfy the

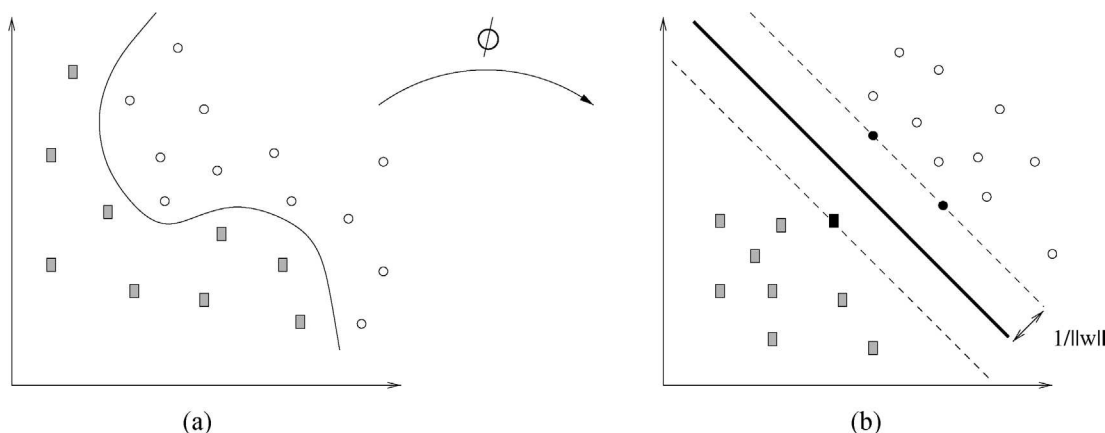


FIG. 1. (a) Original data in the input space. (b) Mapped data in the feature space.

equalities  $y_i((\mathbf{w}^*)^T \Phi(\mathbf{x}_i) + b^*) = 1$  are called support vectors [in Figure 1(b) the support vectors are the black points]. As we will make clear later, the support vectors can be automatically determined from the solution of the optimization problem. Usually the support vectors represent a small fraction of the sample, and the solution is said to be sparse. The hyperplane  $D^*(\mathbf{x}) = 0$  is completely determined by the subsample made up of the support vectors. This fact implies that, for many applications, the evaluation of the decision function  $D^*(\mathbf{x})$  is computationally efficient, allowing the use of SVMs on large data sets in real-time environments.

The SVMs are especially useful within ill-posed contexts. A discussion of ill-posed problems from a statistical point of view may be seen in [55]. A common ill-posed situation arises when dealing with data sets with a low ratio of sample size to dimension. This kind of difficulty often comes up in problems such as automatic classification of web pages or microarrays. Consider, for instance, the following classification problem, where the data set is a text data base that contains 690 documents. These documents have been retrieved from the LISA (Library Science Abstracts) and the INSPEC (bibliographic references for physics, computing and engineering research, from the IEE Institute) data bases, using, respectively, the search keywords “library science” (296 records) and “pattern recognition” (394 records). We have selected as data points the terms that occur in at least ten documents, obtaining 982 terms. Hence, the data set is given by a  $982 \times 690$  matrix, say  $T$ , where  $T_{ij} = 1$  if term  $i$  occurs in document  $j$  and  $T_{ij} = 0$  otherwise. For each term, we check the number of library science and pattern recognition documents that contain it. The highest value determines the class of the term. This procedure is standard in the field of automatic thesaurus generation (see [5]). The task is to check the performance of the SVM classifier in recovering the class labels obtained by the previous procedure. Notice that we are dealing with about 1000 points in nearly 700 dimensions. We have divided the data set into a training set (80% of the data points) and a test set (20% of the data points). Since the sample is relatively small with respect to the space dimension, it should be easy for any method to find a criterion that separates the training set into two classes, but this does not necessarily imply the ability to correctly classify the test data.

The results obtained using Fisher linear discriminant analysis (FLDA), the  $k$ -nearest neighbor classifier

TABLE 1  
*Classification percentage errors for a two-class text data base*

Method	Training error	Test error
FLDA	0.0%	31.4%
$k$ -NN ( $k = 1$ )	0.0%	14.0%
Linear SVM	0.0%	3.0%

( $k$ -NN) with  $k = 1$  and the linear SVM [i.e., taking  $\Phi$  as the identity map  $\Phi(x) = x$ ] are shown in Table 1.

It is apparent that the three methods have been able to find a criterion that perfectly separates the training data set into two classes, but only the linear SVM shows good performance when classifying new data points. The best result for the  $k$ -NN method (shown in the table) is obtained for  $k = 1$ , an unsurprising result, due to the “curse of dimensionality” phenomenon, given the high dimension of the data space. Regarding FLDA, the estimation of the mean vectors and covariance matrices of the groups is problematic given the high dimension and the small number of data points. The SVMs also calculate a linear hyperplane, but are looking for something different—margin maximization, which will only depend on the support vectors. In addition, there is no loss of information caused by projections of the data points. The successful behavior of the support vector method is not casual, since, as we will see below, SVMs are supported by regularization theory, which is particularly useful for the solution of ill-posed problems like the present one.

In summary, we have just described the basics of a classification algorithm which has the following features:

- Reduction of the classification problem to the computation of a linear decision function.
- Absence of local minima in the SVM optimization problem.
- A computationally efficient decision function (sparse solution).

In addition, in the next sections we will also discuss other important features such as the use of kernels as a primary source of information or the tuning of a very reduced set of parameters.

The rest of the paper is organized as follows. Section 2 shows the role of kernels within the SVM approach. In Section 3 SVMs are developed from the regularization theory perspective and some illustrative examples are given. Section 4 reviews a number of successful SVM applications to real-world problems.

In Section 5 algorithmic extensions of SVMs are presented. Finally, in Section 6 some open questions and final remarks are presented.

## 2. THE KERNEL MAPPING

In this section we face one of the key issues of SVMs: how to use  $\Phi(\mathbf{x})$  to map the data into a higher-dimensional space. This procedure is justified by Cover's theorem [16], which guarantees that any data set becomes arbitrarily separable as the data dimension grows. Of course, finding such nonlinear transformations is far from trivial. To achieve this task, a class of functions called kernels is used. Roughly speaking, a kernel  $K(\mathbf{x}, \mathbf{y})$  is a real-valued function  $K: X \times X \rightarrow \mathbb{R}$  for which there exists a function  $\Phi: X \rightarrow Z$ , where  $Z$  is a real vector space, with the property  $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})^T \Phi(\mathbf{y})$ . This function  $\Phi$  is precisely the mapping in Figure 1. The kernel  $K(\mathbf{x}, \mathbf{y})$  acts as a dot product in the space  $Z$ . In the SVM literature  $X$  and  $Z$  are called, respectively, input space and feature space (see Figure 1).

As an example of such a  $K$ , consider two data points  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , with  $\mathbf{x}_i = (x_{i1}, x_{i2})^T \in \mathbb{R}^2$ , and  $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^T \mathbf{x}_2)^2 = (1 + x_{11}x_{21} + x_{12}x_{22})^2 = \Phi(\mathbf{x}_1)^T \Phi(\mathbf{x}_2)$ , where  $\Phi(\mathbf{x}_i) = (1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2})$ . Thus, in this example  $\Phi: \mathbb{R}^2 \rightarrow \mathbb{R}^6$ . As we will show later, explicit knowledge of both the mapping  $\Phi$  and the vector  $\mathbf{w}$  will not be needed: we need only  $K$  in its closed form.

To be more specific, a kernel  $K$  is a positive definite function that admits an expansion of the form  $K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(\mathbf{x}) \Phi_i(\mathbf{y})$ , where  $\lambda_i \in \mathbb{R}^+$ . Sufficient conditions for the existence of such an expansion are given in Mercer's theorem [43]. The function  $K(\mathbf{x}, \mathbf{y})$ , known as a Mercer's kernel, implicitly defines the mapping  $\Phi$  by letting  $\Phi(\mathbf{x}) = (\sqrt{\lambda_1} \Phi_1(\mathbf{x}), \sqrt{\lambda_2} \Phi_2(\mathbf{x}), \dots)^T$ .

Examples of Mercer's kernels are the linear kernel  $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$ , polynomial kernels  $K(\mathbf{x}, \mathbf{y}) = (c + \mathbf{x}^T \mathbf{y})^d$  and the Gaussian kernel  $K_c(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/c}$ . In the first case, the mapping is the identity. Polynomial kernels map the data into finite-dimensional vector spaces. With the Gaussian kernel, the data are mapped onto an infinite dimensional space  $Z = \mathbb{R}^{\infty}$  (all the  $\lambda_i \neq 0$  in the kernel expansion; see [63] for the details).

Given a kernel  $K$ , we can consider the set of functions spanned by finite linear combinations of the form  $f(\mathbf{x}) = \sum_j \alpha_j K(\mathbf{x}_j, \mathbf{x})$ , where the  $\mathbf{x}_j \in X$ . The completion of this vector space is a Hilbert space known as

a reproducing kernel Hilbert space (RKHS) [3]. Since  $K(\mathbf{x}_j, \mathbf{x}) = \Phi(\mathbf{x}_j)^T \Phi(\mathbf{x})$ , the functions  $f(\mathbf{x})$  that belong to a RKHS can be expressed as  $f(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x})$ , with  $\mathbf{w} = \sum_j \alpha_j \Phi(\mathbf{x}_j)$ , that is,  $f(\mathbf{x}) = 0$  describes a hyperplane in the feature space determined by  $\Phi$  [as the one illustrated in Figure 1(b)]. Thus, reproducing kernel Hilbert spaces provide a natural context for the study of hyperplanes in feature spaces through the use of kernels like those introduced in Section 1. Without loss of generality, a constant  $b$  can be added to  $f$  (see [64] for a complete discussion), taking the form

$$(2.1) \quad f(\mathbf{x}) = \sum_j \alpha_j K(\mathbf{x}_j, \mathbf{x}) + b.$$

Equation (2.1) answers the question of how to use  $\Phi(\mathbf{x})$  to map the data onto a higher-dimensional space: Since  $f(\mathbf{x})$  can be evaluated using expression (2.1) [in which only the kernel values  $K(\mathbf{x}_j, \mathbf{x})$  are involved],  $\Phi$  acts implicitly through the closed form of  $K$ . In this way, the kernel function  $K$  is employed to avoid an explicit evaluation of  $\Phi$  (often a high-dimensional mapping). This is the reason why knowledge of the explicit mapping  $\Phi$  is not needed.

As we will show in the next section, SVMs work by minimizing a regularization functional that involves an empirical risk plus some type of penalization term. The solution to this problem is a function that has the form (2.1). This optimization process necessarily takes place within the RKHS associated with the kernel  $K$ . The key point in this computation is the way in which SVMs select the weights  $\alpha_j$  in (2.1) (the points  $\mathbf{x}_j$  are trivially chosen as the sample data points  $\mathbf{x}_i$ ). A nice fact is that the estimation of these weights, which determine the decision function in the RKHS, is reduced to the solution of a smooth and convex optimization problem.

## 3. SUPPORT VECTOR MACHINES: A REGULARIZATION METHOD

In Section 1 we introduced the formulation of SVMs for the situation illustrated in Figure 1(b), where the mapped data have become linearly separable. We consider now the more general case where the mapped data remain nonseparable. This situation is illustrated in Figure 2(a). The SVMs address this problem by finding a function  $f$  that minimizes an empirical error of the form  $\sum_{i=1}^n L(y_i, f(\mathbf{x}_i))$ , where  $L$  is a particular loss function and  $(\mathbf{x}_i, y_i)_{i=1}^n$  is the available data sample. There may be an infinite number of solutions, in which case the problem is ill-posed. Our aim is to show

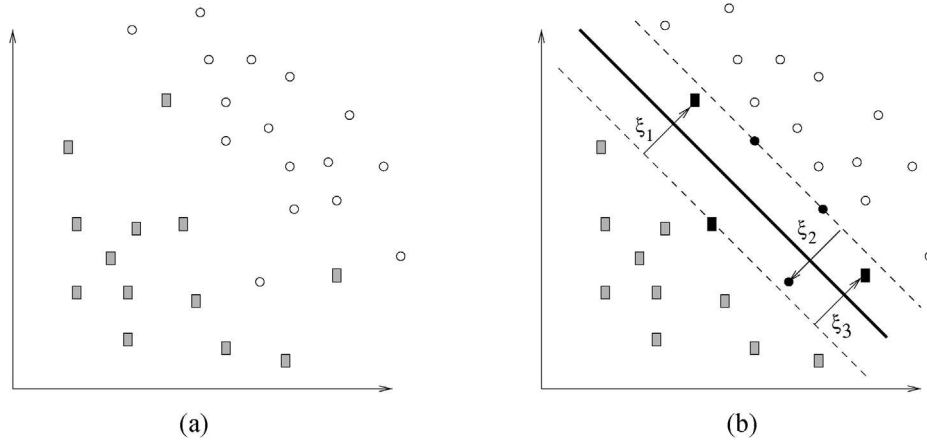


FIG. 2. (a) Nonseparable mapped data in the feature space. (b) Normalized hyperplane for the data in (a).

how SVMs make the problem well-posed. As a consequence, the decision function calculated by the SVM will be unique, and the solution will depend continuously on the data.

The specific loss function  $L$  used within the SVM approach is  $L(y_i, f(\mathbf{x}_i)) = (1 - y_i f(\mathbf{x}_i))_+$ , with  $(x)_+ = \max(x, 0)$ . This loss function is called hinge loss and is represented in Figure 3. It is zero for well classified points with  $|f(\mathbf{x}_i)| \geq 1$  and is linear otherwise. Hence, the hinge loss function does not penalize large values of  $f(\mathbf{x}_i)$  with the same sign as  $y_i$  (understanding large to mean  $|f(\mathbf{x}_i)| \geq 1$ ).

This behavior agrees with the fact that in classification problems only an estimate of the classification boundary is needed. As a consequence, we only take into account points such that  $L(y_i, f(\mathbf{x}_i)) > 0$  to determine the decision function.

To reach well-posedness, SVMs make use of regularization theory, for which several similar approaches

have been proposed [33, 60, 73]. The widest used setting minimizes Tikhonov’s regularization functional [73], which consists of solving the optimization problem

$$(3.1) \quad \min_{f \in H_K} \frac{1}{n} \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+ + \mu \|f\|_K^2,$$

where  $\mu > 0$ ,  $H_K$  is the RKHS associated with the kernel  $K$ ,  $\|f\|_K$  denotes the norm of  $f$  in the RKHS and  $\mathbf{x}_i$  are the sample data points. Given that  $f$  belongs to  $H_K$ , it takes the form  $f(\cdot) = \sum_j \alpha_j K(\mathbf{x}_j, \cdot)$ . As in Section 2,  $f(\mathbf{x}) = 0$  is a hyperplane in the feature space. Using the reproducing property  $\langle K(\mathbf{x}_j, \cdot), K(\mathbf{x}_l, \cdot) \rangle_K = K(\mathbf{x}_j, \mathbf{x}_l)$  (see [3]), it holds that  $\|f\|_K^2 = \langle f, f \rangle_K = \sum_j \sum_l \alpha_j \alpha_l K(\mathbf{x}_j, \mathbf{x}_l)$ .

In (3.1) the scalar  $\mu$  controls the trade-off between the fit of the solution  $f$  to the data (measured by  $L$ ) and the approximation capacity of the function space that  $f$  belongs to (measured by  $\|f\|_K$ ). It can be shown [11,

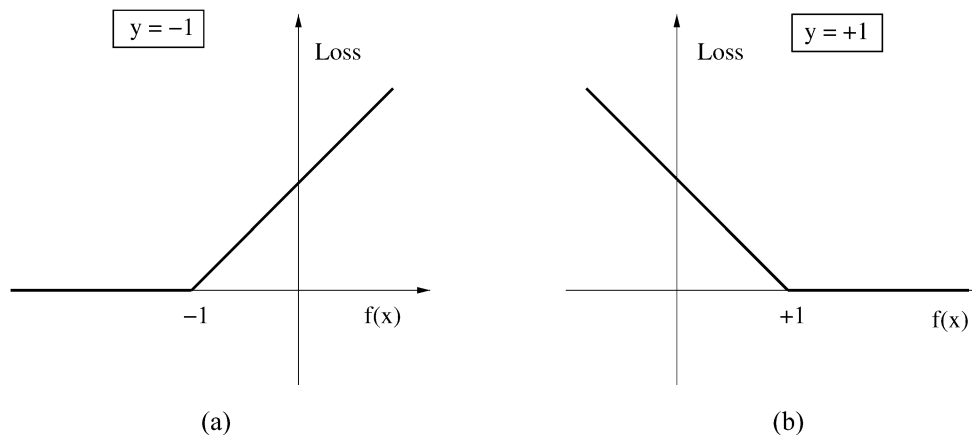


FIG. 3. Hinge loss function  $L(y_i, f(\mathbf{x}_i)) = (1 - y_i f(\mathbf{x}_i))_+$ : (a)  $L(-1, f(\mathbf{x}_i))$ ; (b)  $L(+1, f(\mathbf{x}_i))$ .

48] that the space where the solution is sought takes the form  $\{f \in H_K : \|f\|_K^2 \leq (\sup_{y \in Y} L(y, 0))/\mu\}$ , a compact ball in the RKHS. Note that the larger  $\mu$  is, the smaller is the ball and the more restricted is the search space. This is the way in which regularization theory imposes compactness in the RKHS. Cucker and Smale [21] showed that imposing compactness on the space assures well-posedness of the problem and, thus, uniqueness of the solution (refer to the Appendix for details).

The solution to problem (3.1) has the form  $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$ , where  $\mathbf{x}_i$  are the sample data points, a particular case of (2.1). This result is known as the representer theorem. For details, proofs and generalizations, refer to [36, 67] or [18]. It is immediate to show that  $\|f\|_K^2 = \|\mathbf{w}\|^2$ , where  $\mathbf{w} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}_i)$ . Given this last result, problem (3.1) can be restated as

$$(3.2) \quad \min_{\mathbf{w}, b} \frac{1}{n} \sum_{i=1}^n (1 - y_i(\mathbf{w}^T \Phi(\mathbf{x}_i) + b))_+ + \mu \|\mathbf{w}\|^2.$$

It is worth mentioning that the second term in (3.2) coincides with the term in the objective function of (1.2). Problems (3.1) and (3.2) review some of the key issues of SVMs enumerated at the end of Section 1: Through the use of kernels, the a priori problem of estimating a nonlinear decision function in the input space is transformed into the a posteriori problem of estimating the weights of a hyperplane in the feature space.

Because of the hinge loss function, problem (3.2) is nondifferentiable. This lack of differentiability implies a difficulty for efficient optimization techniques; see [7] or [47]. Problem (3.2) can be turned smooth by straightforwardly formulating it as (see [41])

$$(3.3) \quad \begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & i = 1, \dots, n, \\ & \xi_i \geq 0, \quad i = 1, \dots, n, \end{aligned}$$

where  $\xi_i$  are slack variables introduced to avoid the nondifferentiability of the hinge loss function and  $C = 1/(2\mu n)$ . This is the most widely used SVM formulation.

The slack variables  $\xi_i$  allow violations of constraints (1.1), extending problem (1.2) to the nonseparable case [problem (1.2) would not be solvable for nonseparable data]. The slack variables guarantee the

existence of a solution. The situation is shown in Figure 2(b), which constitutes a generalization of Figure 1(b). Notice that problem (1.2) is a particular case of problem (3.3). To be more specific, if the mapped data become separable, problem (1.2) is equivalent to problem (3.3) when, at the solution,  $\xi_i = 0$ . Intuitively, we want to solve problem (1.2) and, at the same time, minimize the number of nonseparable samples, that is,  $\sum_i \#(\xi_i > 0)$ . Since the inclusion of this term would provide a nondifferentiable combinatorial problem, the smooth term  $\sum_{i=1}^n \xi_i$  appears instead.

We have deduced the standard SVM formulation (3.3) via the use of regularization theory. This framework guarantees that the empirical error for SVMs converges to the expected error as  $n \rightarrow \infty$  [21], that is, the decision functions obtained by SVMs are statistically consistent. Therefore, the separating hyperplanes obtained by SVMs are neither arbitrary nor unstable. This remark is pertinent since Cover's theorem (which guarantees that any data set becomes arbitrarily separable as the data dimension grows) could induce some people to think that SVM classifiers are arbitrary.

By standard optimization theory, it can be shown that problem (3.3) is equivalent to solving

$$(3.4) \quad \begin{aligned} \min_{\lambda} \quad & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^n \lambda_i \\ \text{s.t.} \quad & \sum_{i=1}^n y_i \lambda_i = 0, \\ & 0 \leq \lambda_i \leq C, \quad i = 1, \dots, n. \end{aligned}$$

The  $\lambda_i$  variables are the Lagrange multipliers associated with the constraints in (3.3). This problem is known in optimization theory as the dual problem of (3.3) [7]. It is convex and quadratic and, therefore, every local minimum is a global minimum. In practice, this is the problem to solve, and efficient methods specific for SVMs have been developed (see [34, 58, 61]).

Let the vector  $\lambda^*$  denote the solution to problem (3.4). Points that satisfy  $\lambda_i^* > 0$  are the support vectors (shown in black in Figure 2(b) for the nonseparable case). It can be shown that the solution to problem (3.3) is  $\mathbf{w}^* = \sum_{i=1}^n \lambda_i^* y_i \Phi(\mathbf{x}_i)$  and

$$(3.5) \quad \begin{aligned} b^* = & -\frac{\sum_{i=1}^n \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}^+)}{2} \\ & + \frac{\sum_{i=1}^n \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}^-)}{2}, \end{aligned}$$

where  $\mathbf{x}^+$  and  $\mathbf{x}^-$  are, respectively, two support vectors in classes +1 and -1 such that their associ-

ated Lagrange multipliers  $\lambda^+$  and  $\lambda^-$  hold so that  $0 < \lambda^+ < C$  and  $0 < \lambda^- < C$ .

The desired decision function, which determines the hyperplane  $(\mathbf{w}^*)^T \Phi(\mathbf{x}) + b^* = 0$ , takes the form

$$(3.6) \quad \begin{aligned} D^*(\mathbf{x}) &= (\mathbf{w}^*)^T \Phi(\mathbf{x}) + b^* \\ &= \sum_{i=1}^n \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*. \end{aligned}$$

Equations (3.5) and (3.6) show that  $D^*(\mathbf{x})$  is completely determined by the subsample made up by the support vectors, the only points in the sample for which  $\lambda_i^* \neq 0$ . This definition of support vector is coherent with the geometrical one given in Section 1. The reason is that Lagrange multipliers  $\lambda_i^*$  must fulfill the strict complementarity conditions (see [7]), that is,  $\lambda_i^*(D^*(\mathbf{x}_i) - 1 + \xi_i) = 0$ , where either  $\lambda_i^* = 0$  or  $D^*(\mathbf{x}_i) = 1 - \xi_i$ . Therefore, if  $\lambda_i^* \neq 0$ , then  $D^*(\mathbf{x}_i) = 1 - \xi_i$  and  $\mathbf{x}_i$  is one of the points that defines the decision hyperplane [one of the black points in Figure 2(b)]. Often the support vectors are a small fraction of the data sample and, as already mentioned, the solution is said to be sparse. This property is due to the use of the hinge loss function.

Note that problem (3.4) and equation (3.6) depend only on kernel evaluations of the form  $K(\mathbf{x}, \mathbf{y})$ . Therefore, the explicit mapping  $\Phi$  is not needed to solve the SVM problem (3.4) or to evaluate the decision hyperplane (3.6). In particular, even when the kernel corresponds to an infinite-dimensional space (for instance, the Gaussian kernel), there is no problem with the evaluation of  $\mathbf{w}^* = \sum_{i=1}^n \lambda_i^* y_i \Phi(\mathbf{x}_i)$ , which is not explicitly needed. In practice,  $D^*(\mathbf{x})$  is evaluated using the right-hand side of equation (3.6).

### 3.1 SVMs and the Optimal Bayes Rule

The results in the previous section are coherent with the ones obtained by Lin [40], which state that the support vector machine classifier approaches the optimal Bayes rule and its generalization error converges to the optimal Bayes risk.

Consider a two-group classification problem with classes  $+1$  and  $-1$  and, to simplify, assume equal costs of misclassification. Under this assumption, the expected misclassification rate and the expected cost coincide. Let  $p_1(\mathbf{x}) = P(Y = +1 | X = \mathbf{x})$ , where  $X$  and  $Y$  are two random variables whose joint distribution is  $p(\mathbf{x}, y)$ . The optimal Bayes rule for the minimization of the expected misclassification rate is

$$(3.7) \quad BR(\mathbf{x}) = \begin{cases} +1, & \text{if } p_1(\mathbf{x}) > \frac{1}{2}, \\ -1, & \text{if } p_1(\mathbf{x}) < \frac{1}{2}. \end{cases}$$

On one hand, from the previous section we know that the minimization of problem (3.1) guarantees (via regularization theory) that the empirical risk  $\frac{1}{n} \sum_{i=1}^n (1 - y_i f(\mathbf{x}_i))_+$  converges to the expected error  $E[(1 - Yf(x))_+]$ . On the other hand, in [40] it is shown that the solution to the problem  $\min_f E[(1 - Yf(x))_+]$  is  $f^*(\mathbf{x}) = \text{sign}(p_1(\mathbf{x}) - 1/2)$ , an equivalent formulation of (3.7). Therefore, the minimizer sought by SVMs is exactly the Bayes rule.

In [41] it is pointed out that if the smoothing parameter  $\mu$  in (3.1) is chosen appropriately and the approximation capacity of the RKHS is large enough, then the solution to the SVM problem (3.2) approaches the Bayes rule as  $n \rightarrow \infty$ . For instance, in the two examples shown in the next subsection, where the linear kernel  $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$  is used, the associated RKHS (made up of linear functions) is rich enough to solve the classification problems. A richer RKHS should be used for more complex decision surfaces (see [41]), for instance, the one induced by the Gaussian kernel or those induced by high degree polynomial kernels. Regarding the choice of  $\mu$ , methods to determine it in an appropriate manner have been proposed by Wahba [79, 80, 82].

### 3.2 Illustrating the Performance with Simple Examples

In this first example we consider a two-class separable classification problem, where each class is made up of 1000 data points generated from a bivariate normal distribution  $N(\mu_i, I)$ , with  $\mu_1 = (0, 0)$  and  $\mu_2 = (10, 10)$ . Our aim is to illustrate the performance of the SVM in a simple example and, in particular, the behavior of the algorithm for different values of the regularization parameter  $C$  in problem (3.3). The identity mapping  $\Phi(\mathbf{x}) = \mathbf{x}$  is used. Figure 4(a) illustrates the result for  $C = 1$  (for  $C > 1$ , the same result is obtained). There are exactly three support vectors and the optimal margin separating hyperplane obtained by the SVM is  $1.05x + 1.00y - 10.4 = 0$ . For  $C = 0.01$ , seven support vectors are obtained [see Figure 4(b)], and the discriminant line is  $1.02x + 1.00y - 10.4 = 0$ . For  $C = 0.00001$ , 1776 support vectors are obtained [88.8% of the sample; see Figure 4(c)] and the separating hyperplane is  $1.00x + 1.00y - 13.0 = 0$ . The three hyperplanes are very similar to the (normal theory) linear discriminant function  $1.00x + 1.00y - 10.0 = 0$ . Notice that the smaller  $C$  is, the larger the number of support vectors. This is due to the fact that, in problem (3.3),  $C$  penalizes the value of the  $\xi_i$  variables,

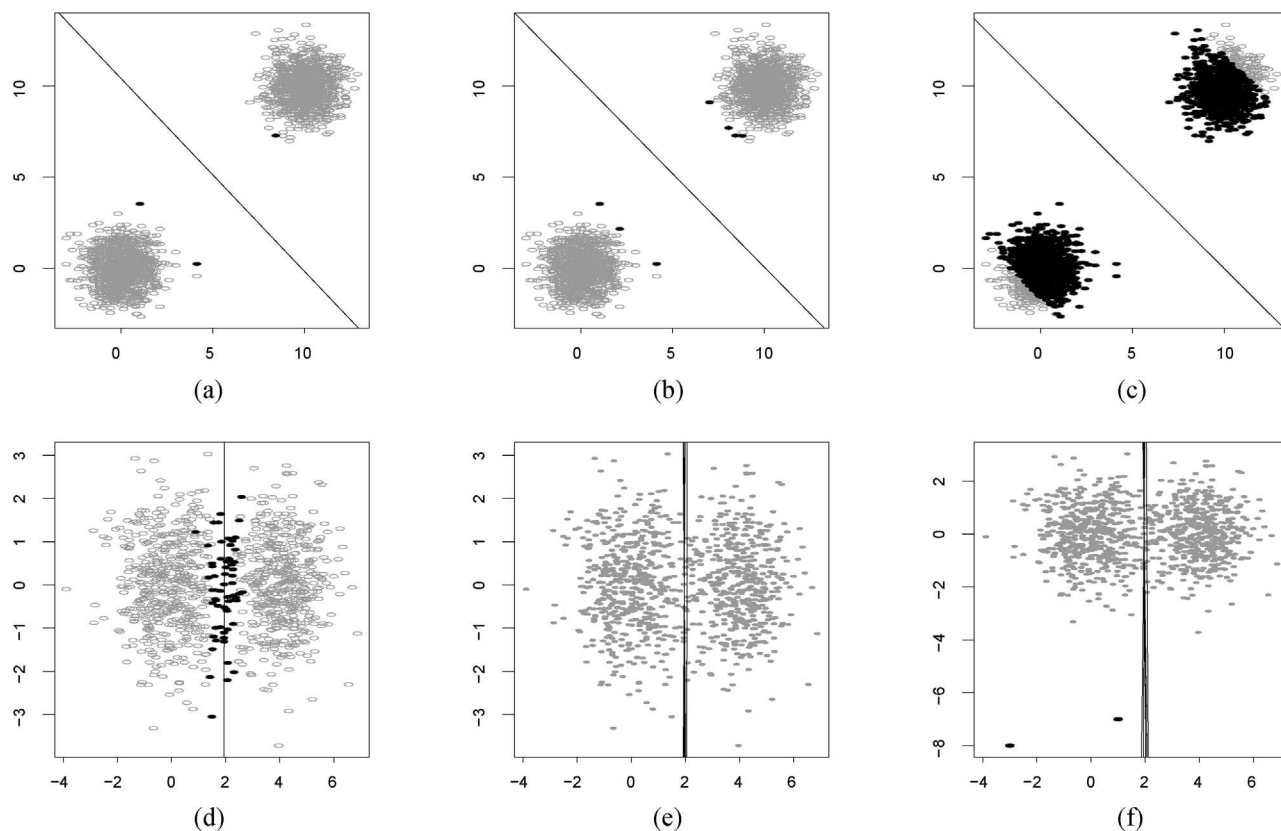


FIG. 4. (a)–(c) SVM hyperplanes for a separable data set. The support vectors are the black points. (d)–(f) SVM hyperplanes for a nonseparable data set.

which determine the width of the band that contains the support vectors.

This second example is quite similar to the previous one, but the samples that correspond to each class are not separable. In this case the mean vectors of the two normal clouds (500 data points in each group) are  $\mu_1 = (0, 0)$  and  $\mu_2 = (4, 0)$ , respectively. The theoretical Bayes error is 2.27%. The normal theory (and optimal) separating hyperplane is  $x = 2$ , that is,  $0.5x + 0y - 1 = 0$ . The SVM estimated hyperplane (taking  $C = 2$ ) is  $0.497x - 0.001y - 1 = 0$ . The error on a test data set with 20,000 data points is 2.3%. Figure 4(d) shows the estimated hyperplane and the support vectors (the black points), which represent 6.3% of the sample. To show the behavior of the method when the parameter  $C$  varies, Figure 4(e) shows the separating hyperplanes for 30 SVMs that vary  $C$  from 0.01 up to 10. All of them look very similar. Finally, Figure 4(f) shows the same 30 hyperplanes when two outlying points (enhanced in black) are added to the left cloud. Since the estimated SVM discriminant functions depend only on the support vectors, the hyperplanes remain unchanged.

### 3.3 The Waveform Data Set

We next illustrate the performance of SVMs on a well-known three-class classification example considered to be a difficult pattern recognition problem [28], the waveform data set introduced in [13]. For the sake of clarity, we reproduce the data description. Each class is generated from a random convex combination of two of three triangular waveforms, namely,  $h_1(i) = \max(6 - |i - 11|, 0)$ ,  $h_2(i) = h_1(i - 4)$  and  $h_3(i) = h_1(i + 4)$ , sampled at the integers  $i \in \{1, \dots, 21\}$ , plus a standard Gaussian noise term. Thus, each data point is represented by  $\mathbf{x} = (x_1, \dots, x_{21})$ , where each component is defined by

$$x_i = uh_1(i) + (1 - u)h_2(i) + \varepsilon_i, \quad \text{for Class 1,}$$

$$x_i = uh_1(i) + (1 - u)h_3(i) + \varepsilon_i, \quad \text{for Class 2,}$$

$$x_i = uh_2(i) + (1 - u)h_3(i) + \varepsilon_i, \quad \text{for Class 3,}$$

with  $u \sim U(0, 1)$  and  $\varepsilon_i \sim N(0, 1)$ . A nice picture of sampled waveforms can be found on page 404 of [28]. The waveform data base [available from the UCI repository (data sets available from the University of California, Irvine, at <http://kdd.ics.uci.edu/>)] contains 5000

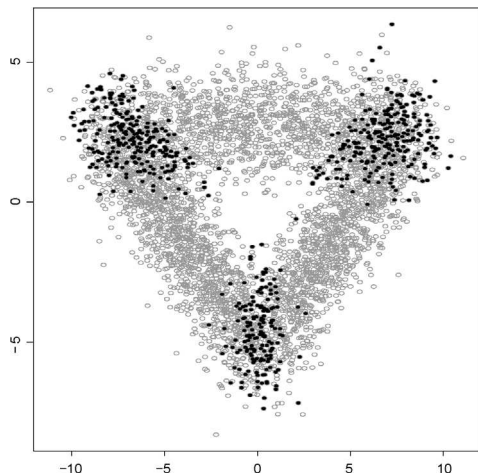


FIG. 5. A PCA projection of the waveform data. The black points represent the misclassified data points using an SVM with the Gaussian kernel.

instances generated using equal prior probabilities. In this experiment we have used 400 data values for training and 4600 for test. Breiman, Friedman, Olshen and Stone [13] reported a Bayes error rate of 14% for this data set. Since we are handling three groups, we use the “one-against-one” approach, in which  $\binom{3}{2}$  binary SVM classifiers are trained and the predicted class is found by a voting scheme: each classifier assigns to each datum a class, being the data point assigned to its most voted class [37]. A first run over ten simulations of the experiment using  $C = 1$  in problem (3.3) and the Gaussian kernel  $K(x, y) = e^{-\|x-y\|^2/200}$  gave an error rate of 14.6%. To confirm the validity of the result, we have run 1000 replications of the experiment. The average error rate over the 1000 simulations on the training data was 10.87% and the average error rate on the test data was 14.67%. The standard errors of the averages were 0.004 and 0.005, respectively. This result improves any other described in the literature to our knowledge. For instance, the best results described in [28] are provided by FLDA and Fisher FDA (flexible discriminant analysis) with MARS (multivariate adaptive regression splines) as the regression procedure (degree = 1), both achieving a test error rate of 19.1%. Figure 5 shows a principal component analysis (PCA) projection of the waveform data into two dimensions with the misclassified test data points (marked in black) for one of the SVM simulations.

#### 4. FURTHER EXAMPLES

In this section we will review some well-known applications of SVMs to real-world problems. In particu-

lar, we will focus on text categorization, bioinformatics and image recognition.

Text categorization consists of the classification of documents into a predefined number of given categories. As an example, consider the document collection made up of Usenet News messages. They are organized in predefined classes such as computation, religion, statistics and so forth. Given a new document, the task is to conduct the category assignment in an automatic way. Text categorization is used by many Internet search engines to select Web pages related to user queries. Documents are represented in a vector space of dimension equal to the number of different words in the vocabulary. Therefore, text categorization problems involve high-dimensional inputs and the data set consists of a sparse document by term matrix. A detailed treatment of SVMs for text categorization can be found in [34]. The performance of SVMs in this task will be illustrated on the Reuters data base. This is a text collection composed of 21,578 documents and 118 categories. The data space in this example has dimension 9947, the number of different words that describe the documents. The results obtained using a SVM with a linear kernel are consistently better along the categories than those obtained with four widely used classification methods: naive Bayes [24], Bayesian networks [29], classification trees [13] and  $k$ -nearest neighbors [17]. The average rate of success for SVMs is 87% while for the mentioned methods the rates are 72%, 80%, 79% and 82%, respectively (see [34] and [25] for further details). However, the most impressive feature of SVM text classifiers is their training time: SVMs are four times faster than the naive Bayes classifier (the fastest of the other methods) and 35 times faster than classification trees. This performance is due to the fact that SVM algorithms take advantage of sparsity in the document by term matrix. Note that methods that involve the diagonalization of large and dense matrices (like the criterion matrix in FLDA) are out of consideration for text classification because of their expensive computational requirements.

We next outline some SVM applications in bioinformatics. There is an increasing interest in analyzing microarray data, that is, analyzing biological samples using their genetic expression profiles. The SVMs have been applied recently to tissue classification [26], gene function prediction [59], protein subcellular location prediction [31], protein secondary structure prediction [32] and protein fold prediction [23], among other tasks. In almost all cases, SVMs outperformed

other classification methods and in their worst case, SVM performance is at least similar to the best non-SVM method. For instance, in protein subcellular location prediction [31], we have to predict protein subcellular positions from prokaryotic sequences. There are three possible location categories: cytoplasmic, periplasmic and extracellular. From a pure classification point of view, the problem reduces to classifying 20-dimensional vectors into three (highly unbalanced) classes. Prediction accuracy for SVMs (with a Gaussian kernel) amounts to 91.4%, while neural networks and a first-order Markov chain [75] have accuracy of 81% and 89.1%, respectively. The results obtained are similar for the other problems. It is important to note that there is still room for improvement.

Regarding image processing, we will overview two well-known problems: handwritten digit identification and face recognition. With respect to the first problem, the U.S. Postal Service data base contains 9298 samples of digits obtained from real-life zip codes (divided into 7291 training samples and 2007 samples for testing). Each digit is represented by a  $16 \times 16$  gray level matrix; therefore each data point is represented by a vector in  $\mathbb{R}^{256}$ . The human classification error for this problem is known to be 2.5% [22]. The error rate for a standard SVM with a third degree polynomial kernel is 4% (see [22] and references therein), while the best known alternative method, the specialized neural network LeNet1 [39], achieves an error rate of 5%. For this problem, using a specialized SVM with a third degree polynomial kernel [22] lowers the error rate to 3.2%—close to the human performance. The key to this specialization lies in the construction of the decision function in three phases: in the first phase, a SVM is trained and the support vectors are obtained; in the second phase, new data points are generated by transforming these support vectors under some groups of transformations, rotations and translations. In the third phase, the final decision hyperplane is built by training a SVM with the new points.

Concerning face recognition, gender detection has been analyzed by Moghaddam and Yang [45]. The data contain 1755 face images (1044 males and 711 females), and the overall error rate for a SVM with a Gaussian kernel is 3.2% (2.1% for males and 4.8% for females). The results for a radial basis neural network [63], a quadratic classifier and FLDA are, respectively, 7.6%, 10.4% and 12.9%.

Another outstanding application of SVMs is the detection of human faces in gray-level images [56]. The

problem is to determine in an image the location of human faces and, if there are any, return an encoding of their position. The detection rate for a SVM using a second degree polynomial kernel is 97.1%, while for the best competing system the rate is 94.6%. A number of impressive photographs that show the effectiveness of this application for face location can be consulted in [57].

## 5. EXTENSIONS OF SVMs: SUPPORT VECTOR REGRESSION

It is natural to contemplate how to extend the kernel mapping explained in Section 2 to well-known techniques for data analysis such as principal component analysis, Fisher linear discriminant analysis and cluster analysis. In this section we will describe support vector regression, one of the most popular extensions of support vector methods, and give some references regarding other extensions.

The ideas underlying support vector regression are similar to those within the classification scheme. From an intuitive viewpoint, the data are mapped into a feature space and then a hyperplane is fitted to the mapped data. From a mathematical perspective, the support vector regression function is also derived within the RKHS context. In this case, the loss function involved is known as the  $\varepsilon$ -insensitive loss function (see [76]), which is defined as  $L(y_i, f(\mathbf{x}_i)) = (|f(\mathbf{x}_i) - y_i| - \varepsilon)_+$ ,  $\varepsilon \geq 0$ . This loss function ignores errors of size less than  $\varepsilon$  (see Figure 6). A discussion of the relationship of the  $\varepsilon$ -insensitive loss function and the ones used in robust statistics can be found in [28]. Using this loss function, the following optimization problem, similar to (3.1) (also consisting of the minimization of

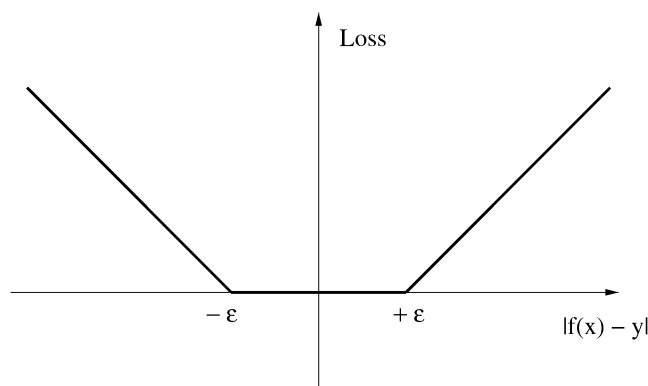


FIG. 6. The  $\varepsilon$ -insensitive loss function  $L(y_i, f(\mathbf{x}_i)) = (|f(\mathbf{x}_i) - y_i| - \varepsilon)_+$ ,  $\varepsilon > 0$ .

a Tikhonov regularization functional), arises:

$$(5.1) \quad \min_{f \in H_K} \frac{1}{n} \sum_{i=1}^n (|f(\mathbf{x}_i) - y_i| - \varepsilon)_+ + \mu \|f\|_K^2,$$

where  $\mu > 0$ ,  $H_K$  is the RKHS associated with the kernel  $K$ ,  $\|f\|_K$  denotes the norm of  $f$  in the RKHS and  $(\mathbf{x}_i, y_i)$  are the sample data points.

Once more, by the representer theorem, the solution to problem (5.1) has the form  $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$ , where  $\mathbf{x}_i$  are the sample data points. It is immediate to show that  $\|f\|_K^2 = \|\mathbf{w}\|^2$ , where  $\mathbf{w} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}_i)$  and  $\Phi$  is the mapping that defines the kernel function. Thus, problem (5.1) can be restated as

$$(5.2) \quad \min_{\mathbf{w}, b} \frac{1}{n} \sum_{i=1}^n (|\mathbf{w}^T \Phi(\mathbf{x}_i) + b - y_i| - \varepsilon)_+ + \mu \|\mathbf{w}\|^2.$$

Since the  $\varepsilon$ -insensitive loss function is nondifferentiable, this problem has to be formulated so that it can be solved by appropriate optimization methods. Straightforwardly, the equivalent (convex) problem to solve is

$$(5.3) \quad \begin{aligned} \min_{\mathbf{w}, b, \xi, \xi'} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi'_i) \\ \text{s.t.} \quad & (\mathbf{w}^T \Phi(\mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i, \\ & \quad \quad \quad i = 1, \dots, n, \\ & y_i - (\mathbf{w}^T \Phi(\mathbf{x}_i) + b) \leq \varepsilon + \xi'_i, \\ & \quad \quad \quad i = 1, \dots, n, \\ & \xi_i, \xi'_i \geq 0, \quad i = 1, \dots, n, \end{aligned}$$

where  $C = 1/(2\mu n)$ . Notice that  $\varepsilon$  appears only in the constraints, forcing the solution to be calculated by taking into account a confidence band around the regression equation. The  $\xi_i$  and  $\xi'_i$  are slack variables that allow for some data points to stay outside the confidence band determined by  $\varepsilon$ . This is the standard support vector regression formulation. Again, the dual of problem (5.3) is a convex quadratic optimization problem, and the regression function takes the same form as equation (2.1). For a detailed exposition of support vector regression, refer to [71] or [69].

One of the most popular applications of support vector regression concerns load forecasting, an important issue in the power industry. In 2001 a proposal based on SVMs for regression was the winner of the European Network of Excellence on Intelligent Technologies competition. The task was to supply the prediction of maximum daily values of electrical loads

for January 1999 (31 data values altogether). To this aim each challenger was given half an hour loads, average daily temperatures and the holidays for the period 1997–1998. The mean absolute percentage error for daily data using the SVM regression model was about 2%, significantly improving the results of most competition proposals. It is important to point out that the SVM procedure used in the contest was standard, in the sense that no special modifications were made for the particular problem at hand. See [14] for further details.

Many other kernel methods have been proposed in the literature. To name a few, there are extensions to PCA [70], Fisher discriminant analysis [6, 44], cluster analysis [8, 46], partial least squares [66], time series analysis [50], multivariate density estimation [49, 68, 54], classification with asymmetric proximities [52], combination with neural network models [53] and Bayesian kernel methods [74].

## 6. OPEN ISSUES AND FINAL REMARKS

The underlying model implemented in SVMs is determined by the choice of the kernel. Deciding which kernel is the most suitable for a given application is obviously an important (and open) issue. A possible approach is to impose some restrictions directly on the structure of the classification (or regression) function  $f$  implemented by the SVM. A way to proceed is to consider a linear differential operator  $D$ , and choose  $K$  as the Green's function for the operator  $D^*D$ , where  $D^*$  is the adjoint operator of  $D$  [4]. It is easy to show that the penalty term  $\|f\|_K^2$  equals  $\|Df\|_{L_2}^2$ . Thus, the choice of the differential operator  $D$  imposes smoothing conditions on the solution  $f$ . This is also the approach used in functional data analysis [65]. For instance, if  $D^*D$  is the Laplacian operator, the kernels obtained are harmonic functions. The simplest case corresponds to (see, e.g., [35])  $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} + c$ , where  $c$  is a constant. Another interesting example is the Gaussian kernel. This kernel arises from a differential operator which penalizes an infinite sum of derivatives. The details for its derivation can be found in [63].

A different approach is to build a specific kernel directly for the data at hand. For instance, Wu and Amari [83] proposed the use of differential geometry methods [2] to derive kernels that improve class separation in classification problems.

An alternative research line arises when a battery of different kernels is available. For instance, when dealing with handwriting recognition, there are a number of

different (nonequivalent) metrics that provide complementary information. The task here is to derive a single kernel which combines the most relevant features of each metric to improve the classification performance (see, e.g., [38] or [42]).

Regarding more theoretical questions, Cucker and Smale [21], as already mentioned, provided sufficient conditions for the statistical consistency of SVMs from a functional analysis point of view (refer to the Appendix for the details). On the other hand, the statistical learning theory developed by Vapnik and Chervonenkis (summarized in [77]) provides necessary and sufficient conditions in terms of the Vapnik–Chervonenkis (VC) dimension (a capacity measure for functions). However, the estimation of the VC dimension for SVMs is often not possible and the relationship between both approaches is still an open issue.

From a statistical point of view an important subject remains open: the interpretability of the SVM outputs. Some (practical) proposals can be consulted in [62, 76] and [72] about the transformation of the SVM classification outputs into a posteriori class probabilities.

Regarding the finite sample performance of SVMs, a good starting point can be found in [55], where bias and variability computations for linear inversion algorithms (a particular case of regularization methods) are studied. The way to extend these ideas to the SVM nonlinear case is an interesting open problem.

Concerning software for SVMs, a variety of implementations are freely available from the Web, most reachable at <http://www.kernel-machines.org/>. In particular, Matlab toolboxes and R/Splus libraries can be downloaded from this site. Additional information on implementation details concerning SVMs can be found in [20] and [69].

As a final proposal, a novice reader could find it interesting to review a number of other regularization methods, such as penalized likelihood methods [27], classification and regression with Gaussian processes [72, 82], smoothing splines [81], functional data analysis [65] and kriging [19].

## APPENDIX: STATISTICAL CONSISTENCY OF THE EMPIRICAL RISK

When it is not possible to assume a parametric model for the data, ill-posed problems arise. The number of data points which can be recorded is finite, while the unknown variables are functions which require an infinite number of observations for their exact description. Therefore, finding a solution implies a choice from an

infinite collection of alternative models. A problem is well-posed in the sense of Hadamard if (1) a solution exists; (2) the solution is unique; (3) the solution depends continuously on the observed data. A problem is ill-posed if it is not well-posed.

Inverse problems constitute a broad class of ill-posed problems [73]. Classification, regression and density estimation can be regarded as inverse problems. In the general setting, we consider a mapping  $H_1 \xrightarrow{A} H_2$ , where  $H_1$  represents a metric function space and  $H_2$  represents a metric space in which the observed data (which could be functions) live. For instance, in a linear regression problem,  $H_1$  corresponds to the finite-dimensional vector space  $\mathbb{R}^{k+1}$ , where  $k$  is the number of regressors;  $H_2$  is  $\mathbb{R}^n$ , where  $n$  is the number of data points; and  $A$  is the linear operator induced by the data matrix of dimension  $n \times (k + 1)$ . Let  $\mathbf{y} = (y_1, \dots, y_n)$  be the vector of response variables and denote by  $f$  the regression equation we are looking for. Then the regression problem consists of solving the inverse problem  $Af = \mathbf{y}$ . A similar argument applies to the classification setting. In this case, the  $\mathbf{y}$  values live in a compact subset of the  $H_2$  space [77].

An example of an inverse problem in which  $H_2$  is a function space is the density estimation one. In this problem  $H_1$  and  $H_2$  are both function spaces and  $A$  is a linear integral operator given by  $(Af)(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}$ , where  $K$  is a predetermined kernel function and  $f$  is the density function we are seeking. The problem to solve is  $Af = F$ , where  $F$  is the distribution function. If  $F$  is unknown, the empirical distribution function  $F_n$  is used instead, and the inverse problem to solve is  $Af = \mathbf{y}$ , with  $\mathbf{y} = F_n$ .

We will focus on classification and regression tasks. Therefore, we assume there exist a function  $f: X \rightarrow Y$  and a probability measure  $p$  defined in  $X \times Y$  so that  $E[y|\mathbf{x}] = f(\mathbf{x})$ . For an observed sample  $\{(\mathbf{x}_i, y_i) \in X \times Y\}_{i=1}^n$ , the goal is to obtain the “best” possible solution to  $Af = \mathbf{y}$ , where, as mentioned above,  $\mathbf{y}$  is the  $n$ -dimensional vector of  $y_i$ ’s and  $A$  is an operator that depends on the  $\mathbf{x}_i$  values. To evaluate the quality of a particular solution, a “loss function”  $L(f; \mathbf{x}, \mathbf{y})$  has to be introduced, which we will denote  $L(y, f(\mathbf{x}))$  in what follows. A common example of a loss function for regression is the quadratic loss  $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ .

Consider the Banach space  $C(X)$  of continuous functions on  $X$  with the norm  $\|f\|_\infty = \sup_{\mathbf{x} \in X} |f(\mathbf{x})|$ . The solution to the inverse problem in each case is the minimizer  $f^*$  of the risk functional  $R(f): C(X) \rightarrow$

$\mathbb{R}$  defined by (see [21])

$$(A.1) \quad R(f) = \int_{X \times Y} L(y, f(\mathbf{x})) p(\mathbf{x}, y) d\mathbf{x} dy.$$

Of course, the solution depends on the function space in which  $f$  lives. Following [21], the hypothesis space, denoted by  $\mathcal{H}$  in the sequel, is chosen to be a compact subset of  $C(X)$ . In particular, only bounded functions  $f: X \rightarrow Y$  are considered.

In these conditions, and assuming a continuous loss function  $L$ , Cucker and Smale [21] proved that the functional  $R(f)$  is continuous. The existence of  $f^* = \arg \min_{f \in \mathcal{H}} R(f)$  follows from the compactness of  $\mathcal{H}$  and the continuity of  $R(f)$ . In addition, if  $\mathcal{H}$  is convex,  $f^*$  will be unique and the problem becomes well-posed.

In practice, it is not possible to calculate  $R(f)$  and the empirical risk  $R_n(f) = \frac{1}{n} \sum_{i=1}^n L(y_i, f(\mathbf{x}_i))$  must be used. This is not a serious complication since asymptotic uniform convergence of  $R_n(f)$  to the risk functional  $R(f)$  is a proven fact (see [21]).

In summary, imposing compactness on the hypothesis space assures well-posedness of the problem to be solved and uniform convergence of the empirical error to the risk functional for a broad class of loss functions, including the square loss and loss functions used in the SVM setting.

The question of how to impose compactness on the hypothesis space is fixed by regularization theory. A possibility (followed by SVMs) is to minimize Tikhonov's regularization functional

$$(A.2) \quad \min_{f \in H} \frac{1}{n} \sum_{i=1}^n L(y_i, f(\mathbf{x}_i)) + \lambda \Omega(f),$$

where  $\lambda > 0$ ,  $H$  is an appropriate function space, and  $\Omega(f)$  is a convex positive functional. By standard optimization theory arguments, it can be shown that, for fixed  $\lambda$ , the inequality  $\Omega(f) \leq C$  holds for a constant  $C > 0$ . Therefore, the space where the solution is searched takes the form  $\mathcal{H} = \{f \in H : \Omega(f) \leq C\}$ , that is, a convex compact subset of  $H$ .

#### ACKNOWLEDGMENTS

Thanks are extended to Executive Editors George Casella and Edward George, and an anonymous editor for their very helpful comments. The first author was supported in part by Spanish Grants TIC2003-05982-C05-05 (MCyT) and MTM2006-14961-C05-05 (MEC). The second author was supported in part by Spanish Grants SEJ2004-03303 and 06/HSE/0181/2004.

#### REFERENCES

- [1] AIZERMAN, M. A., BRAVERMAN, E. M. and ROZONOER, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automat. Remote Control* **25** 821–837.
- [2] AMARI, S.-I. (1985). *Differential-Geometrical Methods in Statistics. Lecture Notes in Statist.* **28**. Springer, New York. [MR0788689](#)
- [3] ARONSZAJN, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc.* **68** 337–404. [MR0051437](#)
- [4] ARONSZAJN, N. (1951). Green's functions and reproducing kernels. In *Proc. Symposium on Spectral Theory and Differential Problems* 355–411.
- [5] BAEZA-YATES, R. and RIBEIRO-NETO, B. (1999). *Modern Information Retrieval*. Addison-Wesley, Harlow.
- [6] BAUDAT, G. and ANOUAR, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation* **12** 2385–2404.
- [7] BAZARAA, M. S., SHERALI, H. D. and SHETTY, C. M. (1993). *Nonlinear Programming: Theory and Algorithms*, 2nd ed. Wiley, New York.
- [8] BEN-HUR, A., HORN, D., SIEGELMANN, H. and VAPNIK, V. (2001). Support vector clustering. *J. Mach. Learn. Res.* **2** 125–137.
- [9] BENNETT, K. P. and CAMPBELL, C. (2000). Support vector machines: Hype or hallelujah? *SIGKDD Explorations* **2** (2) 1–13.
- [10] BOSER, B. E., GUYON, I. and VAPNIK, V. (1992). A training algorithm for optimal margin classifiers. In *Proc. Fifth ACM Workshop on Computational Learning Theory (COLT)* 144–152. ACM Press, New York.
- [11] BOUSQUET, O. and ELISSEEFF, A. (2002). Stability and generalization. *J. Mach. Learn. Res.* **2** 499–526. [MR1929416](#)
- [12] BREIMAN, L. (2001). Statistical modeling: The two cultures (with discussion). *Statist. Sci.* **16** 199–231. [MR1874152](#)
- [13] BREIMAN, L., FRIEDMAN, J., OLSEN, R. and STONE, C. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA. [MR0726392](#)
- [14] CHEN, B.-J., CHANG, M.-W. and LIN, C.-J. (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. *IEEE Transactions on Power Systems* **19** 1821–1830.
- [15] CORTES, C. and VAPNIK, V. (1995). Support-vector networks. *Machine Learning* **20** 273–297.
- [16] COVER, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers* **14** 326–334.
- [17] COVER, T. M. and HART, P. E. (1967). Nearest neighbour pattern classification. *IEEE Trans. Inform. Theory* **13** 21–27.
- [18] COX, D. and O'SULLIVAN, F. (1990). Asymptotic analysis of penalized likelihood and related estimators. *Ann. Statist.* **18** 1676–1695. [MR1074429](#)
- [19] CRESSIE, N. (1993). *Statistics for Spatial Data*. Wiley, New York. [MR1239641](#)
- [20] CRISTIANINI, N. and SHAWE-TAYLOR, J. (2000). *An Introduction to Support Vector Machines*. Cambridge Univ. Press.
- [21] CUCKER, F. and SMALE, S. (2002). On the mathematical foundations of learning. *Bull. Amer. Math. Soc. (N.S.)* **39** 1–49. [MR1864085](#)

- [22] DE COSTE, D. and SCHÖLKOPF, B. (2002). Training invariant support vector machines. *Machine Learning* **46** 161–190.
- [23] DING, C. and DUBCHAK, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* **17** 349–358.
- [24] DOMINGOS, P. and PAZZANI, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* **29** 103–130.
- [25] DUMAIS, S., PLATT, J., HECKERMAN, D. and SAHAMI, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proc. 7th International Conference on Information and Knowledge Management* 148–155. ACM Press, New York.
- [26] FUREY, T. S., CRISTIANINI, N., DUFFY, N., BEDNARSKI, D., SCHUMMER, M. and HAUSSLER, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16** 906–914.
- [27] GREEN, P. J. (1999). Penalized likelihood. *Encyclopedia of Statistical Sciences Update* **3** 578–586. Wiley, New York.
- [28] HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*. Springer, New York. [MR1851606](#)
- [29] HECKERMAN, D., GEIGER, D. and CHICKERING, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* **20** 197–243.
- [30] HERBRICH, R. (2002). *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, Cambridge, MA.
- [31] HUA, S. and SUN, Z. (2001). Support vector machine approach for protein subcellular localization prediction. *Bioinformatics* **17** 721–728.
- [32] HUA, S. and SUN, Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *J. Molecular Biology* **308** 397–407.
- [33] IVANOV, V. V. (1976). *The Theory of Approximate Methods and their Application to the Numerical Solution of Singular Integral Equations*. Noordhoff International, Leyden. [MR0405045](#)
- [34] JOACHIMS, T. (2002). *Learning to Classify Text Using Support Vector Machines*. Kluwer, Boston.
- [35] KANWAL, R. P. (1983). *Generalized Functions*. Academic Press, Orlando, FL. [MR0732788](#)
- [36] KIMELDORF, G. S. and WAHBA, G. (1970). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.* **41** 495–502. [MR0254999](#)
- [37] KRESSEL, U. (1999). Pairwise classification and support vector machines. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 255–268. MIT Press, Cambridge, MA.
- [38] LANCKRIET, G. R. G., CRISTIANINI, N., BARLETT, P., EL GHAOU, L. and JORDAN, M. I. (2002). Learning the kernel matrix with semi-definite programming. In *Proc. 19th International Conference on Machine Learning* 323–330. Morgan Kaufmann, San Francisco.
- [39] LECUN, Y., BOSER, B., DENKER, J. S., HENDERSON, D., HOWARD, R. E., HUBBARD, W. and JACKEL, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation* **1** 541–551.
- [40] LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **6** 259–275. [MR1917926](#)
- [41] LIN, Y., WAHBA, G., ZHANG, H. and LEE, Y. (2002). Statistical properties and adaptive tuning of support vector machines. *Machine Learning* **48** 115–136.
- [42] MARTIN, I., MOGUERZA, J. M. and MUÑOZ, A. (2004). Combining kernel information for support vector classification. *Multiple Classifier Systems. Lecture Notes in Comput. Sci.* **3077** 102–111. Springer, Berlin.
- [43] MERCER, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London A* **209** 415–446.
- [44] MIKA, S., RÄTSCH, G., WESTON, J., SCHÖLKOPF, B. and MÜLLER, K.-R. (1999). Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing* (Y.-H. Hu, J. Larsen, E. Wilson and S. Douglas, eds.) 41–48. IEEE Press, Piscataway, NJ.
- [45] MOGHADDAM, B. and YANG, M.-H. (2002). Learning gender with support faces. *IEEE Trans. Pattern Analysis and Machine Intelligence* **24** 707–711.
- [46] MOGUERZA, J. M., MUÑOZ, A. and MARTIN-MERINO, M. (2002). Detecting the number of clusters using a support vector machine approach. *Proc. International Conference on Artificial Neural Networks. Lecture Notes in Comput. Sci.* **2415** 763–768. Springer, Berlin.
- [47] MOGUERZA, J. M. and PRIETO, F. J. (2003). An augmented Lagrangian interior-point method using directions of negative curvature. *Math. Program. Ser. A* **95** 573–616. [MR1969766](#)
- [48] MUKHERJEE, S., RIFKIN, P. and POGGIO, T. (2003). Regression and classification with regularization. *Nonlinear Estimation and Classification. Lecture Notes in Statist.* **171** 111–128. Springer, New York. [MR2005786](#)
- [49] MUKHERJEE, S. and VAPNIK, V. (1999). Multivariate density estimation: A support vector machine approach. Technical Report, AI Memo 1653, MIT AI Lab.
- [50] MÜLLER, K.-R., SMOLA, A. J., RÄTSCH, G., SCHÖLKOPF, B., KOHLMORGEN, J. and VAPNIK, V. (1999). Using support vector machines for time series prediction. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 243–253. MIT Press, Cambridge, MA.
- [51] MÜLLER, P. and RIOS INSUA, D. (1998). Issues in Bayesian analysis of neural network models. *Neural Computation* **10** 749–770.
- [52] MUÑOZ, A., MARTIN, I. and MOGUERZA, J. M. (2003). Support vector machine classifiers for asymmetric proximities. *Artificial Neural Networks and Neural Information. Lecture Notes in Comput. Sci.* **2714** 217–224. Springer, Berlin.
- [53] MUÑOZ, A. and MOGUERZA, J. M. (2003). Combining support vector machines and ARTMAP architectures for natural classification. *Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Artificial Intelligence* **2774** 16–21. Springer, Berlin.
- [54] MUÑOZ, A. and MOGUERZA, J. M. (2006). Estimation of high-density regions using one-class neighbor machines. *IEEE Trans. Pattern Analysis and Machine Intelligence* **28** 476–480.
- [55] O’SULLIVAN, F. (1986). A statistical perspective on ill-posed inverse problems (with discussion). *Statist. Sci.* **1** 502–527. [MR0874480](#)

- [56] OSUNA, E., FREUND, R. and GIROSI, F. (1997). Training support vector machines: An application to face detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition* 130–136. IEEE Press, New York.
- [57] OSUNA, E., FREUND, R. and GIROSI, F. (1997). Support vector machines: Training and applications. CBCL Paper 144/AI Memo 1602, MIT AI Lab.
- [58] OSUNA, E., FREUND, R. and GIROSI, F. (1997). An improved training algorithm for support vector machines. In *Proc. IEEE Workshop on Neural Networks for Signal Processing* 276–285. IEEE Press, New York.
- [59] PAVLIDIS, P., WESTON, J., CAI, J. and GRUNDY, W. N. (2001). Gene functional classification from heterogeneous data. In *Proc. Fifth Annual International Conference on Computational Biology* 249–255. ACM Press, New York.
- [60] PHILLIPS, D. L. (1962). A technique for the numerical solution of certain integral equations of the first kind. *J. Assoc. Comput. Mach.* **9** 84–97. [MR0134481](#)
- [61] PLATT, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 185–208. MIT Press, Cambridge, MA.
- [62] PLATT, J. C. (2000). Probabilities for SV machines. In *Advances in Large-Margin Classifiers* (P. J. Bartlett, B. Schölkopf, D. Schuurmans and A. J. Smola, eds.) 61–74. MIT Press, Cambridge, MA.
- [63] POGGIO, T. and GIROSI, F. (1990). Networks for approximation and learning. *Proc. IEEE* **78** 1481–1497.
- [64] POGGIO, T., MUKHERJEE, S., RIFKIN, R., RAKHLIN, A. and VERRI, A. (2001). *b*. CBCL Paper 198/AI Memo 2001-011, MIT AI Lab.
- [65] RAMSAY, J. O. and SILVERMAN, B. W. (1997). *Functional Data Analysis*. Springer, New York.
- [66] ROSIPAL, R. and TREJO, L. J. (2001). Kernel partial least squares regression in reproducing kernel Hilbert space. *J. Mach. Learn. Res.* **2** 97–123.
- [67] SCHÖLKOPF, B., HERBRICH, R., SMOLA, A. J. and WILLIAMSON, R. C. (2001). A generalized representer theorem. *Lecture Notes in Artificial Intelligence* **2111** 416–426. Springer, Berlin.
- [68] SCHÖLKOPF, B., PLATT, J. C., SHAWE-TAYLOR, J., SMOLA, A. J. and WILLIAMSON, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation* **13** 1443–1471.
- [69] SCHÖLKOPF, B. and SMOLA, A. J. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.
- [70] SCHÖLKOPF, B., SMOLA, A. J. and MÜLLER, K.-R. (1999). Kernel principal component analysis. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 327–352. MIT Press, Cambridge, MA.
- [71] SMOLA, A. J. and SCHÖLKOPF, B. (1998). A tutorial on support vector regression. NeuroColt2 Technical Report Series, NC2-TR-1998-030.
- [72] SOLLICH, P. (2002). Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning* **46** 21–52.
- [73] TIKHONOV, A. N. and ARSENIN, V. Y. (1977). *Solutions of Ill-Posed Problems*. Wiley, New York.
- [74] TIPPING, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **1** 211–244. [MR1875838](#)
- [75] VAN KAMPEN, N. G. (1981). *Stochastic Processes in Physics and Chemistry*. North-Holland, Amsterdam. [MR0648937](#)
- [76] VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York. [MR1367965](#)
- [77] VAPNIK, V. (1998). *Statistical Learning Theory*. Wiley, New York. [MR1641250](#)
- [78] VAPNIK, V. and CHERVONENKIS, A. (1964). A note on a class of perceptrons. *Automat. Remote Control* **25** 103–109.
- [79] WAHBA, G. (1980). Spline bases, regularization, and generalized cross validation for solving approximation problems with large quantities of noisy data. In *Approximation Theory III* (W. Cheney, ed.) 905–912. Academic Press, New York. [MR0602818](#)
- [80] WAHBA, G. (1985). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Ann. Statist.* **13** 1378–1402. [MR0811498](#)
- [81] WAHBA, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia. [MR1045442](#)
- [82] WAHBA, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 69–88. MIT Press, Cambridge, MA.
- [83] WU, S. and AMARI, S.-I. (2002). Conformal transformation of kernel functions: A data-dependent way to improve support vector machine classifiers. *Neural Processing Letters* **15** 59–67.

# Comment

Olivier Bousquet and Bernhard Schölkopf

Our contribution will be short, but we will try to compensate by being particularly opinionated. The field of support vector machines (SVMs) and related kernel methods has produced an impressive range of theoretical results, algorithms and success stories in real-world applications. While it originated in machine learning, it is also concerned with core problems of statistics and it is thus timely to publish a comprehensive article that discusses these methods from a statistician's point of view. We shall use this opportunity to make a few general comments, largely about the field rather than about the present paper.

Many papers about SVMs start off saying something like “SVMs are great because they are based on statistical learning theory” (this probably includes some of our own writings). Moguerza and Muñoz are more careful and only say that SVMs appeared in the context of statistical learning theory. What actually is the connection between SVMs and statistical learning theory?

Historically, SVMs and their precursors were (co-) developed by Vladimir Vapnik, one of the fathers of statistical learning theory. Statistical learning theory includes an analysis of machine learning which is independent of the distribution underlying the data. However, this analysis cannot provide any a priori guarantee that SVMs (or any other algorithm) will work well on a real-world problem. So what is special about SVMs, if anything?

In our view, what is special about SVMs is the combination of the following ingredients: first and foremost, the use of positive definite kernels; then regularization via the norm in the associated reproducing kernel Hilbert space; finally, the use of a convex loss function which is minimized by a classifier and not a regressor.

*The magic of kernels.* Positive definite kernels and their feature space interpretation do provide a very nice

---

Olivier Bousquet is Director of Research, Pertinence, F-75002 Paris, France (e-mail: [o.bousquet@pertinence.com](mailto:o.bousquet@pertinence.com)). Bernhard Schölkopf is Professor and Director, Max Planck Institute for Biological Cybernetics, D-72076 Tübingen, Germany (e-mail: [bs@tuebingen.mpg.de](mailto:bs@tuebingen.mpg.de)).

way to look at a whole class of algorithms; however, it is important to stress that they do not bring any *statistical* guarantee by themselves. The statistical guarantees available stem from the regularization (or learning theory) point of view. We shall return to this point below.

The main advantages of positive definite kernels are the following:

1. They allow easy construction of a nonlinear algorithm from a linear one, often without incurring additional computational cost.
2. They provide generality via the fact that they can be defined on nonvectorial data and do not, in general, require an explicit mapping to a reproducing kernel Hilbert space.

Historically, the first point was initially considered one of the major advantages of kernels and it triggered a significant number of kernel algorithms other than SVMs, starting with kernel principal component analysis (PCA). More recently, the second point has arguably taken over the role of the key selling point for kernel methods. The application of learning algorithms to nonvectorial data has become the field where nowadays a lot of the action is happening in the machine learning world, in particular concerning applications on structured data (e.g., in biology or natural language processing). We are curious to see whether the field of statistics will also embrace these possibilities.

*A sober look at the geometric interpretation.* The geometric point of view is an original way to look at SVMs and quite possibly the right way to come up with an algorithm like the SVM in the first place. However, it does not yield comprehensive statistical understanding. More precisely, there is no way to prove that large margin separating hyperplanes perform better than other types of hyperplanes independently of the distribution of the data.

Sure enough, the geometric point of view does provide intuition and motivates a large number of related algorithms, but one should not be fooled by geometric intuition or two-dimensional illustrations. The fact that data that are not linearly separable in input space suddenly becomes linearly separable in the so-called feature space (as depicted on Figure 1 of the main paper)

has led to misconceptions. Indeed, the picture seems to suggest that the kernel has magically placed the two clouds of points in two separate regions of the space, and hence uncovered the right decision boundary.

The feature space often has a nonintuitive geometry. Let us take the example of the Gaussian Radial Basis Function (RBF) kernel. The corresponding feature space is of infinite dimension and the points are all mapped to the positive orthant of the unit sphere. Any two disjoint point sets in input space can be separated by a hyperplane in this feature space.

There is thus something mysterious happening in this space, but this space is but one way to look at things. We might instead directly look at the SVM algorithm and see that it loosely speaking tries to combine functions of the form  $k(x_i, \cdot)$  using coefficients chosen to maximize the real-valued predictions  $y_i f(x_i)$  on the training set. This brings us to the concept of *margin*. People usually say that maximizing the margin is good for generalization. There are two concepts of margin to be distinguished:

- *Geometric margin* (distance to the hyperplane). This is related to the norm of the weight vector, so that maximizing the margin corresponds to minimizing the norm (i.e., to regularization). Regularization can indeed lead to good generalization, provided the kind of smoothness enforced by the regularizer reflects the specifics of the problem.
- *Numerical margin* [i.e., the quantity  $y_i f(x_i)$  which appears in the hinge loss used by the standard SVM]. The main reason why it makes sense to maximize this margin is because the hinge loss is a convex non-increasing upper bound of the classification loss, so that making  $y_i f(x_i)$  large will ensure that the hinge loss is small and thus that we minimize the number of misclassification errors. However, this only means that minimizing the empirical hinge loss might lead to minimizing the empirical misclassification error, but does not guarantee that the expected misclassification error will be minimized as well.

These two notions are quite distinct, yet they are sometimes confused because they are entangled in the algorithms. For instance, if one minimizes the hinge loss over linear combinations of kernels and if there exists a combination such that the total hinge loss on the training set is zero, then this combination is not unique: we can multiply it by an arbitrary positive scale factor. Introducing a constraint on the norm of the weight vector is a natural way to remove this gauge freedom. This

constraint is not innocent. It introduces a coupling between the numerical and the geometric margins: maximizing the geometric margin (in the context of an appropriate nonlinear kernel) leads to regularization which prevents overfitting by penalizing complex functions, while maximizing the numerical margin leads to minimization of the empirical error. Searching for a function with small empirical error while penalizing the complexity is the key to most reasonable learning algorithms.

*Convexity and loss functions.* Another attractive feature of positive definite kernels is that they allow non-linearization of learning algorithm while preserving the *convexity* of the associated optimization problem. This is also one reason for the success of SVMs: the optimization problem is easier to handle than that of other algorithms such as artificial neural networks. The introduction of SVMs with kernels in the machine learning community suddenly moved the focus from optimization algorithms (e.g., multiple variants of gradient descent) to optimization criteria. This has created significant interest in convex functionals (for all kinds of problems such as model selection, semisupervised or unsupervised learning) and methods of convexifying existing functionals.

In the context of supervised learning, this search for convexity has led to the introduction of many different convex loss functions. However, something that has often been overlooked is the set of properties the loss function has to satisfy so that it leads to a consistent algorithm. For example, in the classification setting, a minimum requirement is that with sufficient data, minimizing the loss should lead to minimization of the misclassification error. For standard SVMs, the fact that the hinge loss satisfies this property was noticed relatively late (see reference [40] of the main paper) and, more surprisingly, in the context of multi-class classification this has been addressed only very recently. It has been proved in [1] that several variants of multiclass SVM do not have the required property. Of course, this is not to say that they perform poorly on a finite sample, but it is important to understand what an algorithm is aiming at and how it should behave as the sample size increases.

Moguerza and Muñoz are indeed aware of the fact that the minimizer of the hinge loss is the Bayes classifier (or rather is a function which has the same sign as that of the Bayes classifier), but they later say that there is still work to be done to provide a probabilistic interpretation of the output values produced by SVM

classifiers. This is somewhat problematic because, at least asymptotically, there is no possible relationship between probabilities and output values. [This follows from the consistency property: With an appropriate kernel, the values of the function produced by the SVM algorithm will converge to exactly  $+1$  or  $-1$  on all the points where  $P(Y|X) \in ]0, 1/2[ \cup ]1/2, 1[$  so that the value of  $f(X)$  will have no relationship to  $P(Y|X)$ .] Hence, on a finite sample, if a relationship occurs, it will likely be by pure chance (or because the kernel happens to regularize exactly in the way needed for the preferred functions to look like the conditional probability density function).

To conclude this section with a more philosophical viewpoint, let us mention that the SVM algorithm also reinforces the belief that one should be concerned about the objective rather than about the model: what is important is not whether one can identify the “true” target function; rather, one should try to find *some* function, from a large class, which will perform well. This belief is shared by many researchers in the machine learning community, and it probably distinguishes them from “classical” statisticians, as argued, for example, in [3].

*Theoretical considerations.* Regarding the statistical analysis of the SVM algorithm, besides the works cited in the paper, there are a few additional references that are worth mentioning; for example, [6] first proved universal consistency of  $L_1$ -SVM with a Gaussian kernel, while Steinwart and Scovel [8] and Steinwart [7] obtained rates of convergence under various conditions. Also, more recently, the consistency of SVM has been proved by Vert and Vert [9] in the case where the regularization parameter is held fixed, but the kernel width goes to zero. This suggests that there is a coupling between both types of regularization (provided by a small norm of the function and a large kernel width).

It is now clear that the VC dimension is not the right parameter to capture the rates of convergence, especially when studying real-valued functions classes. Alternative possibilities (based on Rademacher averages) along with finite-sample performance bounds can be found, for example, in [2].

Progress has also been made in understanding the role of sparsity in SVMs. First of all, the number of support vectors is asymptotically linear in the sample size if the Bayes error is nonzero. Second, on large data bases the number of support vectors is usually too large for fast testing (hence the development of reduced set methods which can be applied to nonsparse models [4, 5]).

*Why do SVM work so well in practice?* There is probably no theoretical answer to this question. The fact that they are universally consistent is surely interesting, but does not explain anything about finite sample performance on real-world data sets (e.g., the  $k$ -nearest neighbor algorithm is also universally consistent). The sparsity also does not explain it. Regularization (by the kernel width and by the function norm) surely plays a role (by preventing overfitting) but this cannot be quantified. Indeed, in statistical terms, one can only tell the effect of regularization on the *variance* but not on the *bias*, at least if one does not make specific assumptions on the smoothness of the target function. The only possible answer to this question might thus be that on those problems where SVMs excel, the kernel that is used induces a regularizer that incorporates appropriate prior knowledge about the problems or, equivalently, it captures the right notion of similarity. In a large majority of applications, the Gaussian RBF kernel is used and its success simply means that the Euclidean distance in input space is locally meaningful for those problems. [Indeed, the Gaussian kernel incorporates a notion of similarity which is a monotonic function of the Euclidean distance. In this case, the SVM produces a “local rule”: The prediction at a given point is a weighted combination of the labels of nearby points (where the weight mainly depends on the distance and is adapted by the coefficients  $\lambda_i$  which appear in equation (3.4) of the main paper).]

*Future directions for research.* Although, as explained by Moguerza and Muñoz, the SVM algorithm in itself has several interesting merits, we think that what is most important about it is its impact on the field of machine learning and statistics. It has introduced new concepts and ideas that have considerably influenced their progress, and we expect that the acquired momentum will lead to further advances, in domains such as structured learning, joint kernels (mixing inputs and outputs), links to graphical models, and semi-supervised learning, to name but a few. In a different direction, one could try to extend the notion of kernel so as to handle higher level similarities, such as analogies (which can be considered as similarities between pairs of examples).

There are also several important questions that need to be addressed so as to bridge the gap between basic research and applications. For instance, there is no satisfactory method for choosing the parameters other than using cross-validation, which can be an obstacle in applications. Moreover, there are still significant computational issues arising from the implementation of

SVM-like algorithms using nonlinear kernels for large-scale problems.

#### REFERENCES

- [1] BARTLETT, P. (2006). Asymptotic properties of convex optimization methods for multiclass classification. Mathematical Foundations of Learning Theory II. Slides available at [www.dma.ens.fr/~stoltz/MFLT/Talks%20MFLT2/June%203/PeterBartlett.pdf](http://www.dma.ens.fr/~stoltz/MFLT/Talks%20MFLT2/June%203/PeterBartlett.pdf).
- [2] BLANCHARD, G., BOUSQUET, O. and MASSART, P. (2006). Statistical performance of support vector machines. Preprint. Available at [ida.first.fraunhofer.de/~blanchard/publi/BlaBouMas06\\_rev01.pdf](http://ida.first.fraunhofer.de/~blanchard/publi/BlaBouMas06_rev01.pdf).
- [3] BREIMAN, L. (2001). Statistical modeling: The two cultures (with discussion). *Statist. Sci.* **16** 199–231. MR1874152
- [4] BURGESS, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **2** 121–167.
- [5] SCHÖLKOPF, B. and SMOLA, A. J. (2002). *Learning with Kernels*. MIT Press.
- [6] STEINWART, I. (2002). Support vector machines are universally consistent. *J. Complexity* **18** 768–791. MR1928806
- [7] STEINWART, I. (2005). Consistency of support vector machines and other regularized kernel machines. *IEEE Trans. Inform. Theory* **51** 128–142. MR2234577
- [8] STEINWART, I. and SCOVEL, C. (2005). Fast rates for support vector machines. *Learning Theory. Lecture Notes in Comput. Sci.* **3359** 279–294. Springer, Berlin. MR2203268
- [9] VERT, R. and VERT, J.-P. (2006). Consistency and convergence rates of one-class SVMs and related algorithms. *J. Mach. Learn. Res.* **7** 817–854.

# Comment

Peter L. Bartlett, Michael I. Jordan and Jon D. McAuliffe

## INTRODUCTION

The support vector machine (SVM) has played an important role in bringing certain themes to the fore in computationally oriented statistics. However, it is important to place the SVM in context as but one member of a class of closely related algorithms for nonlinear classification. As we discuss, several of the “open problems” identified by the authors have in fact been the subject of a significant literature, a literature that may have been missed because it has been aimed not only at the SVM but at a broader family of algorithms. Keeping the broader class of algorithms in mind also helps to make clear that the SVM involves certain specific algorithmic choices, some of which have favorable consequences and others of which have unfavorable consequences—both in theory and in practice. The broader context helps to clarify the ties of the SVM to the surrounding statistical literature.

We have at least two broader contexts in mind for the SVM. The first is the family of “large-margin” classification algorithms—a class that includes boosting and logistic regression. All of these algorithms involve the minimization of a convex contrast or loss function that upper bounds the 0–1 loss function. The SVM makes a specific choice of convex loss function—the so-called hinge loss. Hinge loss has some potentially desirable properties (e.g., sparseness) and some potentially undesirable properties (e.g., lack of calibration to posterior probabilities). As we discuss, much of the theoretical analysis of the SVM is best carried out by focusing on convexity and abstracting away from the details of specific loss functions.

Second, as the authors note, the SVM is an instance of the broader family of statistical procedures based on

reproducing kernel Hilbert spaces (RKHSs). The authors’ emphasis is on the use of RKHS methods to provide basis expansions for discriminant functions and regression functions. RKHS ideas have, however, been carried significantly further in recent years, enlivening areas of computationally oriented statistics beyond classification and regression. We wish to convey some of the reasons for this broader interest in RKHS-based approaches.

There are both computational and statistical motivations for focusing on methods based on convexity and reproducing kernel Hilbert spaces. In the remainder of this discussion we attempt to disentangle some of these motivations, but we wish to emphasize at the outset that it is precisely because these methods bring computational and statistical considerations together that they are so interesting.

## CONVEXITY

The SVM is one example of a general strategy for solving the binary classification problem via a “convex surrogate loss function.” To develop this perspective, let us define binary classification as the problem of choosing a discriminant function  $f : \mathcal{X} \rightarrow \mathbb{R}$  that minimizes misclassification risk,

$$R(f) = P(Y \neq \text{sgn}(f(X))) = \mathbf{E}\ell(Yf(X)),$$

where  $X \in \mathcal{X}$  is the covariate,  $Y \in \{\pm 1\}$  is the binary response, and  $\ell(\alpha) = 1$  for  $\alpha \leq 0$  and  $= 0$  otherwise. The family of large-margin classification algorithms attacks this problem indirectly by minimizing a quantity known as the  $\phi$ -risk,

$$R_\phi(f) = \mathbf{E}\phi(Yf(X)),$$

where  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  is a surrogate for the loss function  $\ell$ , and  $Yf(X)$  is called the *margin* of  $f$  on the observation  $(X, Y)$ . The margin indicates not only whether the observation is correctly classified by  $f$ , but how close  $f$  comes to choosing the opposite label. The surrogate loss function  $\phi$  is chosen so that large margins correspond to small losses.

Given a data set  $(X_1, Y_1), \dots, (X_n, Y_n)$ , we can form the *empirical*  $\phi$ -risk

$$\hat{R}_\phi(f) = \frac{1}{n} \sum_{i=1}^n \phi(Y_i f(X_i))$$

---

*Peter L. Bartlett and Michael I. Jordan are Professors, Computer Science Division and Department of Statistics, University of California, Berkeley, California 94720, USA (e-mail: bartlett@stat.berkeley.edu; jordan@stat.berkeley.edu). Jon D. McAuliffe is Assistant Professor, Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104-6340, USA (e-mail: mcjon@wharton.upenn.edu).*

and attempt to minimize this quantity with respect to the discriminant function  $f$ . When  $\phi$  is chosen as a convex function and  $f$  is constrained to lie in a convex family of prediction rules, the minimization becomes a convex optimization problem. Contrast this with minimization of empirical 0–1 (misclassification) risk,

$$\min_{f \in \mathcal{F}} \hat{R}(f) = \frac{1}{n} \sum_{i=1}^n \ell(Y_i f(X_i)),$$

a problem whose exact or even approximate solution is known to be intractable for most nontrivial function classes  $\mathcal{F}$  (e.g., Arora, Babai, Stern and Sweedyk, 1997).

In the case of SVMs, the convex surrogate is the hinge loss

$$\phi(\alpha) = \begin{cases} 1 - \alpha, & \text{if } \alpha \leq 1, \\ 0, & \text{otherwise,} \end{cases}$$

and the function  $f$  is chosen from the RKHS  $\mathcal{H}$  defined by the kernel. However, the hinge loss is not the only convex surrogate worth considering. Using the binomial deviance function as a convex surrogate and optimizing over linear functions on  $R^p$  yields logistic regression. Just as with the SVM, a nonlinear version of logistic regression can be defined by making use of reproducing kernels (Zhu and Hastie, 2005). AdaBoost (Schapire and Singer, 1999) can also be interpreted as a large-margin method, with  $\phi(\alpha) = e^{-\alpha}$ ; similar greedy ensemble methods correspond to other choices of  $\phi$ .

The benefits of empirical convex risk minimization are not just computational. Searching for a prediction rule which achieves a large margin on many training examples, rather than just correctly classifying them, is an implicit form of regularization. For example, certain function classes of infinite Vapnik–Chervonenkis (VC) dimension, where empirical 0–1 risk minimization does not yield good classifiers, can be used effectively in the large-margin framework (Bartlett, 1998; Schapire, Freund, Bartlett and Lee, 1998).

Taking the margin-based viewpoint highlights the important role convexity plays in the success of SVMs. On the other hand, the authors' emphasis on the need to find a differentiable or smooth formulation seems misplaced. In the differentiable case, the key property of a convex objective function  $f$  is that, for any two points  $x, y$  in the domain,

$$(1) \quad f(y) \geq f(x) + \left( \frac{\partial f(x)}{\partial x} \right)^\top (y - x).$$

Thus, local behavior of  $f$  (its gradient at  $x$ ) determines a global lower bound on  $f$ . The existence of this bound makes possible efficient algorithms for convex optimization. However, property (1) holds in a slightly generalized form even for nondifferentiable convex functions. A *subgradient* of a convex  $f$  at  $x$  is a vector  $g$  such that

$$f(y) \geq f(x) + g^\top (y - x) \quad \forall y.$$

The *subdifferential*  $\partial f(x)$  of  $f$  at  $x$  is the set of  $f$ 's subgradients at  $x$ . The subdifferential is the natural analog of the gradient for nonsmooth objectives: any point in  $\partial f(x)$  provides the equivalent of property (1);  $0 \in \partial f(x)$  if and only if  $x$  is a global minimizer of  $f$ ; and the subdifferential contains only the gradient at points of differentiability. Moreover, every convex function has nonempty subdifferentials throughout the interior of its domain. There has been a great deal of successful research on efficient algorithms for nonsmooth convex optimization using “bundle methods” based on subdifferentials. See, for example, Hiriart-Urruty and Lemaréchal (1993), Borwein and Lewis (2000) and Boyd and Vandenberghe (2004).

## Statistical Analysis

To address issues such as statistical consistency and finite sample behavior in the large-margin framework, we need to decompose the risk  $R(f_n)$  into three components. Two of the components are the approximation error and the estimation error that are familiar from other areas of nonparametric statistics. The third component arises from the use of the convex surrogate  $\phi$  in place of the 0–1 loss.

We can quantify the effect of this third component through an inequality of the form

$$\psi(R(f) - R^*) \leq R_\phi(f) - R_\phi^*,$$

where  $\psi$  is a convex, nonnegative function and  $f$  is an arbitrary measurable function (Bartlett, Jordan and McAuliffe, 2006). Notice that such an inequality relates the *excess risk*,  $R(f) - R^*$ , to the *excess  $\phi$ -risk*,  $R_\phi(f) - R_\phi^*$ . Here, the Bayes risk,  $R^*$ , is defined by  $R^* = \inf_g R(g)$ , where the infimum is over all measurable functions, and  $R_\phi^*$  is the minimal  $\phi$ -risk,  $R_\phi^* = \inf_g R_\phi(g)$ . An optimal inequality of this form has been obtained for any nonnegative surrogate loss function  $\phi$ , where  $\psi$  can be written explicitly in terms of  $\phi$  (Bartlett, Jordan and McAuliffe, 2006). In the case of SVMs, where  $\phi$  is the hinge loss,  $\psi$  turns out to be the identity (Zhang 2004; Blanchard, Bousquet and Massart, 2006).

Thus, for SVMs we can write an optimal upper bound on the excess risk as

$$\begin{aligned} R(f_n) - R^* &\leq R_\phi(f_n) - R_\phi^* \\ &= \left( R_\phi(f_n) - \inf_{f \in \mathcal{H}} R_\phi(f) \right) + \left( \inf_{f \in \mathcal{H}} R_\phi(f) - R_\phi^* \right). \end{aligned}$$

This decomposition into the estimation error and the approximation error reflects the bias–variance trade-off, ubiquitous in nonparametric estimation. On the one hand, the function  $f_n$  must come from a suitably simple class to ensure that the performance of  $f_n$  on the finite training sample is representative of its true performance and so the estimation error  $R_\phi(f_n) - \inf_{f \in \mathcal{H}} R_\phi(f)$  is not too large. On the other hand,  $f_n$  must be suitably complex, so that its  $\phi$ -risk is not too much larger than the optimal  $R_\phi^*$ . One common approach to this model selection problem is the method of sieves, where the function  $f_n$  is chosen as the minimizer of the empirical  $\phi$ -risk over classes  $\mathcal{F}_n$  that grow progressively richer as the sample size  $n$  increases. The other common approach is to add a regularization term, so that  $f_n$  is chosen as the minimizer of

$$\hat{R}_\phi(f) + \lambda_n \Omega(f)$$

for some regularization functional  $\Omega$  that penalizes complex functions and for some regularization coefficients  $\lambda_n$ . In the case of SVMs, the regularization functional is the squared norm in the reproducing kernel Hilbert space  $\mathcal{H}$ . The regularization approach and the method of sieves are closely related. In particular, since  $0 \in \mathcal{H}$  and  $\hat{R}_\phi(0) = 1$  for the hinge loss, we know that

$$\begin{aligned} \|f_n\|_{\mathcal{H}}^2 &\leq \frac{1}{\lambda_n} (\hat{R}_\phi(f_n) + \lambda_n \|f_n\|_{\mathcal{H}}^2) \\ &\leq \frac{1}{\lambda_n} (\hat{R}_\phi(0) + \lambda_n \|0\|_{\mathcal{H}}^2) = \frac{1}{\lambda_n}. \end{aligned}$$

Thus, the SVM chooses a function from the sieve  $\mathcal{F}_n = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}}^2 \leq 1/\lambda_n\}$ .

It is important to note that the regularization coefficient  $\lambda_n$  must decrease with the sample size, so as to ensure universal consistency. Indeed, the approximation error,  $\inf_{f \in \mathcal{F}_n} R_\phi(f) - R_\phi^*$ , must go to zero. On the other hand, it should decrease sufficiently slowly that the estimation error,  $R_\phi(f_n) - \inf_{f \in \mathcal{F}_n} R_\phi(f)$ , also decreases to zero. (Consistency does not follow from uniform convergence of the empirical to expected error in a ball of fixed radius, as the appendix of the paper suggests.)

One of the most important consequences of the choice of a kernel is the way it affects this trade-off between the estimation and approximation errors. We can view the kernel, and hence the norm in the RKHS, as defining a complexity hierarchy. A good kernel is one for which a good approximation to  $R_\phi^*$  can be obtained with a function that is low in the complexity hierarchy, in the sense that it has a small norm.

The paper lists several statistical issues as important open problems, notably the finite-sample performance of SVMs and the estimation of a posteriori class probabilities. In fact, bounds on the estimation error,  $R_\phi(f_n) - \inf_{f \in \mathcal{F}_n} R_\phi(f)$ , for finite sample size have been known for years. These bounds are expressed in terms of properties of the eigenvalues of the *Gram matrix*, the matrix whose entries are  $k(x_i, x_j)$  (see, e.g., Shawe-Taylor, Bartlett, Williamson and Anthony, 1998; Bartlett and Shawe-Taylor, 1999; Williamson, Smola and Schölkopf, 1999; Mendelson, 2002; Bartlett, Bousquet and Mendelson, 2005; Blanchard, Bousquet and Massart, 2006). Moreover, these results provide an essential foundation for proofs of consistency (Steinwart, 2002, 2005; Zhang, 2004). In contrast, while the VC dimension is central to the analysis of methods that minimize the empirical 0–1 risk, it is not relevant to SVMs. Indeed, for any kernel that is sufficiently rich to allow a universal consistency result, the RKHS necessarily has infinite VC dimension. See Bartlett and Shawe-Taylor (1999) for an explanation of the role of the VC dimension and other complexity measures that are more appropriate to the analysis of the finite-sample behavior of SVMs.

Moreover, the problem of estimation of a posteriori class probabilities using SVM classification outputs is not open, in the following sense. It is an easy calculation to see that, for any  $x$ , the minimizer of the conditional expectation  $\mathbf{E}[\phi(Yf(x))|X = x]$  is  $f^*(x) = \text{sign}(2\eta(x) - 1)$ , where  $\eta(x) = \Pr[Y = 1|X = x]$ . Furthermore, results of Steinwart (2003) establish that, under reasonable conditions on the kernel function and the rate at which the regularization coefficients go to zero, the function  $f_n$  chosen by the SVM converges to  $f^*$ . Thus, asymptotically there is no information about the a posteriori class probabilities in the SVM classification outputs. Thus the SVM framework does not appear to provide an appropriate starting point for the estimation of a posteriori probabilities. This is a distinguishing feature of the hinge loss  $\phi$ : its minimization cannot correspond to fitting a probability model, since it is indifferent to distinct values of the class probability.

To elaborate on this point, note that one of the attractive features of SVM classifiers is their sparseness: The proportion of nonzero dual variables (“support vectors”) is typically small. Steinwart (2004) has presented a beautiful relationship between the number of support vectors in an SVM and the Bayes risk. Assuming that the kernel is appropriately chosen and the regularization is reduced sufficiently slowly as the sample size increases, the asymptotic proportion of support vectors is equal to twice the Bayes risk. On the other hand, it is known that if we replace the hinge loss  $\phi$  with the differentiable quadratic loss, sparseness disappears, but then the a posteriori class probabilities can be estimated asymptotically. Indeed, sparseness and the ability to estimate conditional probabilities seem to be incompatible. If the hinge loss is replaced by any of a large family of loss functions, it can be shown (Bartlett and Tewari, 2004) that the proportion of support vectors approaches the expectation of a certain function of the conditional probability  $\eta(X)$ , and this function is maximal for those values of  $\eta(X)$  for which estimation of the conditional probability is possible asymptotically.

### KERNEL METHODS

While the SVM has helped to bring RKHS ideas to new prominence, focusing on the SVM runs the risk of limiting the appreciation of the scope and potential impact of RKHS methods. In this section we augment the presentation by Moguerza and Muñoz to provide a broader context for the understanding of the RKHS aspect of the SVM approach.

The main point that we wish to make in this section is this: A RKHS provides computationally efficient machinery for evaluating and optimizing a variety of statistical functionals of interest. The empirical loss functions of nonparametric classification and regression are a special case—one in which the focus is on the basis function expansions provided by a RKHS—but there are other roles for a RKHS.

A key idea in RKHS methodology is that an inner product can be computed by evaluating a *reproducing kernel*  $k(x, x')$ —a function of two arguments that obeys symmetry and positive definiteness conditions. A reproducing kernel may take the form of an analytic function (e.g., the Gaussian kernel) or may take the form of a computational procedure (e.g., the string kernel, which involves a dynamic program).

Given a set of  $n$  data points  $\{x_1, x_2, \dots, x_n\}$  and given a reproducing kernel  $k(x, x')$ , one can form the

Gram matrix. The SVM and the other kernel methods mentioned in Section 5 of Moguerza and Muñoz are all based on various operations (computation of eigenvectors, determinants, inverses, etc.) on Gram matrices. This reduction of the data to a Gram matrix is significant computationally; indeed, while the naive computational complexity of many kernel methods is  $O(n^3)$ , the exploitation of more sophisticated numerical linear back ends can drive this cost down to  $O(nk^2)$ , where  $k$  is a measure of the effective rank of a Gram matrix. The effective rank is typically small.

Let us now consider a problem that at first glance seems to have little to do with kernel methods—the problem of assessing whether random variables  $X_1$  and  $X_2$  are independent. Independence can be reduced to correlation by considering transformations of the random variables. In particular,  $X_1$  and  $X_2$  are independent if and only if

$$\rho = \max_{h_1, h_2 \in \mathcal{H}} \text{Corr}(h_1(X_1), h_2(X_2)) = 0$$

for a suitably rich function space  $\mathcal{H}$ . Indeed, if  $\mathcal{H}$  is  $L_2$  and thus contains the Fourier basis, this statement reduces to a classical fact about characteristic functions. More interestingly, the result also holds for certain RKHSs. Moreover, the reproducing property of the kernel implies that function evaluation in a RKHS reduces to an inner product,  $h_1(X_1) = \langle k(\cdot, X_1), h_1 \rangle$ , where  $k(\cdot, \cdot)$  is the reproducing kernel for  $\mathcal{H}$  and  $\langle \cdot, \cdot \rangle$  is the corresponding inner product. Thus correlations can be computed as

$$\begin{aligned} & \text{Corr}(h_1(X_1), h_2(X_2)) \\ &= \text{Corr}(\langle k(\cdot, X_1), h_1 \rangle, \langle k(\cdot, X_2), h_2 \rangle). \end{aligned}$$

Maximizing over  $h_1$  and  $h_2$  thus amounts to maximizing the correlation between projections of vectors in a pair of Hilbert spaces; this is nothing but canonical correlation analysis (CCA) in  $\mathcal{H}$  (cf. Leurgans, Moyeed and Silverman, 1993). Moreover, using the reproducing property it is easy to show that this functional CCA computation can be reduced to a generalized eigenvector problem on a pair of Gram matrices (one Gram matrix for the observations of  $X_1$  and another Gram matrix for the observations of  $X_2$ ). Thus we can assess independence by carrying out a kernelized version of CCA.

This line of argument is due to Bach and Jordan (2002), who showed how it could be used to fit a semi-parametric model known as independent component analysis. The general approach has been carried further by Gretton et al. (2005), who established relationships

between RKHS-based measures of independence and mutual information.

Further work in this vein has shown how RKHS ideas can be used to develop computationally efficient methods for fitting a wide range of other nonparametric and semiparametric models. Consider, for example, the problem of *sufficient dimension reduction* (SDR) for regression (Cook, 1998). This problem can be formulated as the problem of finding a matrix  $B_0$  whose column space spans a subspace  $\mathcal{S}$  of the covariate space and which satisfies

$$p_{Y|X}(y|x) = p_{Y|B_0^T X}(y|B_0^T x).$$

That is, the regression of  $Y$  on  $X$  depends only on the projection of  $X$  on  $\mathcal{S}$ . Note that no additional assumptions are made about the regression function. We see that SDR can be viewed in terms of an assertion of *conditional independence*. Fukumizu, Bach and Jordan (2006) have shown how such assertions can be evaluated in terms of covariance operators on RKHSs. In particular, they showed that  $B_0$  can be alternatively characterized as

$$B_0 = \arg \min_B \Sigma_{YY|X}^B,$$

where  $\Sigma_{YY|X}^B$  is a conditional covariance operator on a RKHS, based on the kernel function  $k^B(x, x') = k(B^T x, B^T x')$ . This operator can be estimated and minimized (in the sense of the partial order of self-adjoint operators) using Gram matrices. Under weak conditions, this yields a consistent procedure for estimating  $B_0$ .

## CONCLUSIONS

A statistician who encounters SVMs for the first time might have difficulty understanding the source of the excitement. After all, the SVM is a modest variation on some standard statistical methodology—it involves RKHS expansions of discriminant or regression functions combined with a simple piecewise-linear loss function. Nonetheless, this combination has noteworthy practical consequences. In particular, by paying careful attention to the optimization problem that arises in the SVM and by paying careful attention to the resulting numerical linear algebra, the SVM can be applied to very large classification and regression problems. Moreover, these lessons extend beyond the specific setting of the SVM. As we have emphasized, the key ideas of convex relaxation and reproducing kernels have applications well beyond the SVM. They permit an approach to nonparametric statistics that blends

tools from nearby areas of applied mathematics such as optimization theory, functional analysis, numerical linear algebra and combinatorics, undeniably expanding the scope of activities in nonparametric statistics and expanding the scale of problems that can be addressed.

## REFERENCES

- ARORA, S., BABAI, L., STERN, J. and SWEEDYK, Z. (1997). The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. System Sci.* **54** 317–331. [MR1462727](#)
- BACH, F. R. and JORDAN, M. I. (2002). Kernel independent component analysis. *J. Mach. Learn. Res.* **3** 1–48. [MR1966051](#)
- BARTLETT, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inform. Theory* **44** 525–536. [MR1607706](#)
- BARTLETT, P. L., BOUSQUET, O. and MENDELSON, S. (2005). Local Rademacher complexities. *Ann. Statist.* **33** 1497–1537. [MR2166554](#)
- BARTLETT, P. L., JORDAN, M. I. and MCAULIFFE, J. D. (2006). Convexity, classification and risk bounds. *J. Amer. Statist. Assoc.* **101** 138–156.
- BARTLETT, P. L. and SHAWE-TAYLOR, J. (1999). Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 43–54. MIT Press, Cambridge, MA.
- BARTLETT, P. L. and TEWARI, A. (2004). Sparseness versus estimating conditional probabilities: Some asymptotic results. *Learning Theory. Lecture Notes in Comput. Sci.* **3120** 564–578. Springer, Berlin. [MR2177935](#)
- BLANCHARD, G., BOUSQUET, O. and MASSART, P. (2006). Statistical performance of support vector machines. Preprint. Available at [ida.first.fraunhofer.de/~blanchard/publi/BlaBouMas06\\_rev01.pdf](http://ida.first.fraunhofer.de/~blanchard/publi/BlaBouMas06_rev01.pdf).
- BORWEIN, J. M. and LEWIS, A. S. (2000). *Convex Analysis and Nonlinear Optimization*. Springer, New York. [MR1757448](#)
- BOYD, S. and VANDENBERGHE, L. (2004). *Convex Optimization*. Cambridge Univ. Press. [MR2061575](#)
- COOK, R. D. (1998). *Regression Graphics*. Wiley, New York. [MR1645673](#)
- FUKUMIZU, K., BACH, F. R. and JORDAN, M. I. (2006). Kernel dimension reduction for regression. Technical report, Dept. Statistics, Univ. California, Berkeley.
- GRETTON, A., HERBRICH, R., SMOLA, A., BOUSQUET, O. and SCHÖLKOPF, B. (2005). Kernel methods for measuring independence. *J. Mach. Learn. Res.* **6** 2075–2129.
- HIRIART-URRUTY, J.-B. and LEMARÉCHAL, C. (1993). *Convex Analysis and Minimization Algorithms* **1, 2**. Springer, Berlin. [MR1261420](#), [MR1295240](#)
- LEURGANS, S. E., MOYEED, R. A. and SILVERMAN, B. (1993). Canonical correlation analysis when the data are curves. *J. Roy. Statist. Soc. Ser. B* **55** 725–740. [MR1223939](#)
- MENDELSON, S. (2002). Geometric parameters of kernel machines. *Computational Learning Theory. Lecture Notes in Comput. Sci.* **2375** 29–43. Springer, Berlin. [MR2040403](#)

- SCHAPIRE, R. E., FREUND, Y., BARTLETT, P. L. and LEE, W. S. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Statist.* **26** 1651–1686. [MR1673273](#)
- SCHAPIRE, R. E. and SINGER, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning* **37** 297–336.
- SHAWE-TAYLOR, J., BARTLETT, P. L., WILLIAMSON, R. C. and ANTHONY, M. (1998). Structural risk minimization over data-dependent hierarchies. *IEEE Trans. Inform. Theory* **44** 1926–1940. [MR1664055](#)
- STEINWART, I. (2002). Support vector machines are universally consistent. *J. Complexity* **18** 768–791. [MR1928806](#)
- STEINWART, I. (2003). Sparseness of support vector machines. *J. Mach. Learn. Res.* **4** 1071–1105. [MR2125346](#)
- STEINWART, I. (2004). Sparseness of support vector machines—Some asymptotically sharp bounds. In *Advances in Neural Information Processing Systems* (B. Schölkopf, L. K. Saul and S. Thrun, eds.) **16** 1069–1076. MIT Press, Cambridge, MA.
- STEINWART, I. (2005). Consistency of support vector machines and other regularized kernel machines. *IEEE Trans. Inform. Theory* **51** 128–142. [MR2234577](#)
- WILLIAMSON, R. C., SMOLA, A. J. and SCHÖLKOPF, B. (1999). Entropy numbers, operators and support vector kernels. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 127–144. MIT Press, Cambridge, MA.
- ZHANG, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.* **32** 56–85. [MR2051001](#)
- ZHU, J. and HASTIE, T. (2005). Kernel logistic regression and the import vector machine. *J. Comput. Graph. Statist.* **14** 185–205. [MR2137897](#)

# Comment

Grace Wahba

## 1. INTRODUCTION

The authors are to be commended for jumping in to describe support vector machines (SVMs), not an easy thing to do since the literature for SVMs has grown at least exponentially in the last few years. A Google search for “support vector machines” gave “about 1,180,000” hits as of this writing. The authors have nevertheless made a nice selection of important points to emphasize. As noted, SVMs were proposed for classification in the early 1990s by arguments like those behind Figure 1 in their paper. The use of SVMs grew rapidly among computer scientists, as it was found that they worked very well in all kinds of practical applications. The theoretical underpinnings that went with the original proposals were different than those in the classical statistical literature, for example, those related to Bayes risk, and so had less impact in the statistical literature. The convergence of SVMs and regularization methods (or, rather the convergence of the “SVM community” and the “regularization community”) was a major impetus in the study of the (classical) statistical properties of the SVM. One point at which this convergence took place was at an American Mathematical Society meeting at Mt. Holyoke in 1996. The speaker was describing the SVM with the so-called kernel trick when an anonymous person at the back of the room remarked that the SVM with the kernel trick was the solution to an optimization problem in a reproducing kernel Hilbert space (RKHS). Once it was clear to statisticians that the SVM can be obtained as the result of an optimization/regularization problem in a RKHS, tools known to statisticians in this context were rapidly employed to show how the SVM could be modified to take into account nonrepresentative sample sizes, unequal misclassification costs and more than two classes, and to show in each case that it directly targets the Bayes risk under very general circumstances (see also [5, 8]). Thus, a “classical” explanation of why they work so well was provided.

---

*Grace Wahba is the IJ Schoenberg-Hilldale Professor of Statistics, Department of Statistics, University of Wisconsin, 1300 University Avenue, Madison, Wisconsin 53706, USA (e-mail: wahba@stat.wisc.edu), and is also a member of the Computer Sciences Department and the Biostatistics and Medical Informatics Department.*

## 2. MERCER'S KERNELS AND POSITIVE DEFINITE FUNCTIONS

Let  $\mathcal{T}$  be a.d.o. (any dirty old) domain and let  $K(s, t), s, t \in \mathcal{T}$ , be a symmetric, positive definite function of two variables;  $K$  is said to be positive definite if for any  $n$ , and any  $t_1, \dots, t_n \in \mathcal{T}$ , the  $n \times n$  matrix with  $ij$ th element  $K(t_i, t_j)$  is nonnegative definite. In the early SVM literature, as well as in the present paper, the kernel is described as having a representation  $K(s, t) = \sum_{\nu=1}^{\infty} \lambda_{\nu} \Phi_{\nu}(s) \Phi_{\nu}(t)$ . Here the (nonnegative)  $\lambda_{\nu}$  and the  $\Phi_{\nu}$  are the eigenvalues and eigenvectors of  $K$ . A representation as in this sum is sufficient for  $K$  to be positive definite (see [13] on the Mercer Hilbert–Schmidt theorem), but the so-called radial basis functions (RBF) popular in machine learning, of the form  $K(s, t) = k(\|s - t\|)$ ,  $s, t$  in Euclidean  $d$ -space  $E^d$ , do not have a countable sequence of eigenvalues and eigenvectors—complex exponentials play the role of eigenvectors (see [3]). The Gaussian kernel  $K_c(x, y) = e^{-\|x-y\|^2/c}$  is such an example. Although the notion of a countable expansion was used in uncoupling the linear SVM from its linearity restriction (and seems to be repeated over and over), the lack of a countable set of eigenvectors and eigenvalues does not affect the use of the Gaussian kernel or any other positive definite function in an SVM; as the authors note, only values of  $K$  are needed. The RBF probably just do not want to be called “Mercer’s kernels” (!). Positive definite functions are sometimes called reproducing kernels, relating to their association with RKHS [1].

Given a collection of objects (which could be vectors, images, sounds, graphs, texts, trees, ...) in a.d.o. domain  $\mathcal{T}$ , a positive definite matrix with  $ij$  entry  $K(i, j)$  defines a (squared) distance  $d_{ij}$  between the  $i$ th and  $j$ th object as

$$d_{ij} = K(i, i) + K(j, j) - 2K(i, j)$$

(and, in addition, this distance comes with an inner product). It can be argued that using distance between objects, defined in some way, is truly fundamental to classification and, therefore, positive definite kernels, since they provide a distance, play a fundamental role.

### 3. LARGE MARGIN CLASSIFIERS AND REGULARIZATION

Referring to equation (3.1) in the main paper, note that the elementary cost function  $(1 - y_i f(\mathbf{x}_i))_+$  depends only on  $\tau_i = y_i f(\mathbf{x}_i)$ . If  $y_i$  and  $f(\mathbf{x}_i)$  have the same sign, then  $f$  will classify  $y_i$  correctly, and if they have different signs, then  $f$  will classify  $y_i$  incorrectly. The term  $\tau_i$  is frequently called the margin, and classifiers that depend on the data only through  $\tau$  are called large margin classifiers. The cost function  $c(\tau) = (\tau)_+$  is called the misclassification counter, and it would be considered the ideal cost function if it were not for the fact that it leads to a nonconvex, nontractable optimization problem. Considering Bernoulli data coded as  $y_i = 1$  or  $y_i = 0$ , the penalized likelihood estimate, where the cost function is the negative log likelihood, goes back at least to [12]. In that paper, members of the exponential family were considered as cost functions and it was natural to put the log likelihood in the canonical form for distributions in the exponential family. Thus the log likelihood for Bernoulli data is parameterized by the logit  $f(x) = \log p(x)/(1 - p(x))$ . However, if Bernoulli data are recoded as  $y_i = \pm 1$ , then the log likelihood (cost function) becomes  $\mathcal{L}(y, f) = (1 + e^{-yf})$ . Since thresholding  $p(x)$  at  $p = 1/2$  is equivalent to thresholding at  $f = 0$ , the penalized log likelihood estimate is also a large margin classifier.

It turns out that there are lots of large margin classifiers with the property that the sign of the estimate that

minimizes

$$\frac{1}{n} \sum_{i=1}^n c(y_i f(\mathbf{x}_i)) + \mu \|f\|_K^2$$

tends to the sign of the log odds ratio, assuming that the problem is tuned adequately and that the RKHS associated with  $K$  is rich enough for the problem at hand. The following rather amazing result is from [6]: Let  $c(z) < c(-z)$ , every  $z > 0$ , and let  $c'(0) \neq 0$  exist. If  $Ec(Yf((X)|\mathbf{X} = \mathbf{x}))$  has a global minimizer  $\bar{f}(\mathbf{x})$  and  $f(\mathbf{x}) \neq 0$ , then  $\text{sign}(\bar{f}(\mathbf{x})) = (\text{sign } f(\mathbf{x}))$ . A bunch of examples are given in [6]. Note the result that the lowly squared difference  $\mathcal{L}(y, f) = (y - f)^2$  leads to a large margin classifier since if  $|y| = 1$ , then  $(y - f)^2 \equiv (1 - yf)^2$ . This large margin classifier (!) is sometimes called the least squares support vector machine, but it is nothing more than ordinary ridge regression on data that have been coded as  $\pm 1$ . Many large margin classifiers have been proposed, both convex and nonconvex, that claim various properties; four of the many are described in [11, 14, 17, 19]. These classifiers are said to have some special advantages, either theoretical, computational or practical, and it is interesting to understand more generally the circumstances under which one cost function can be better than another. Considering accuracy as well as computational tractability, it is unlikely that there will be just one best cost function for all classification problems (see the comparison in Figure 1). The hinge function occupies a niche as a

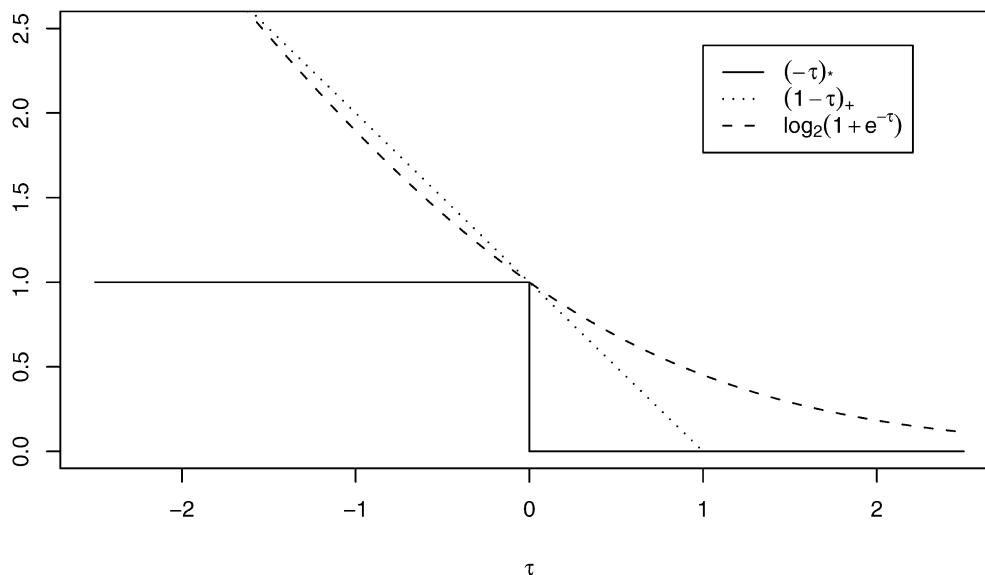


FIG. 1. Comparison of the cost functions  $c(\tau) = (-\tau)_+$ ,  $c(\tau) = (1 - \tau)_+$  and  $c(\tau) = \log_2(1 + e^{-\tau})$ , which are the misclassification function, the hinge function and the negative log-likelihood function, respectively. Any strictly convex function that goes through 1 at  $\tau = 0$  will be an upper bound on the misclassification function  $(-\tau)_+$  and will be a looser bound than some hinge function  $(1 - \theta\tau)_+$ .

general purpose large margin classifier that is the closest convex upper bound, in some sense, to the misclassification function.

#### 4. PROBABILITY ESTIMATES AND THE SVM

I respectfully disagree with the authors' remark that "from a statistical point of view, an important subject remains open: the interpretability of the SVM outputs." I think the appropriate interpretation is that the SVM targets the sign of the log odds ratio *directly*; see [7]. Since the target function  $\text{sign } f(\mathbf{x})$  is discontinuous at  $f(\mathbf{x})$ , and the SVM is found as an optimization problem in a RKHS which is typically a space of continuous functions, it cannot jump at the boundary, but there may be a Gibbs effect there. Since the SVM is generally a smooth approximation which tends not to stray too far outside of the interval  $[-1, 1]$ , there is a tendency to believe that  $2p - 1$  can be inferred from the SVM. This is not, however the case. A toy problem which is easy to drive toward asymptopia illustrates this point.

In Figure 2, the solid line gives  $2p(x) - 1$ , where  $p(x)$  is the true conditional probability of the + class. Data  $y_i$  have been generated as  $y_i = 1$  with probability  $p(x_i)$  and  $-1$  with probability  $1 - p(x_i)$  for 300

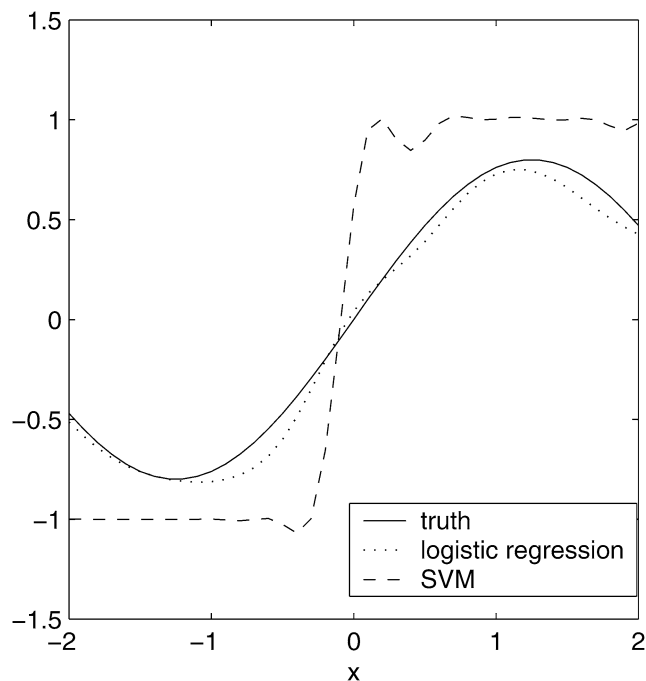


FIG. 2. Solid line: true conditional probability  $2p(x) - 1 = \text{prob} Y = 1$ ; dashed line: fitted SVM; dotted line: fitted penalized likelihood estimate. Data  $y_i$  have been generated according to  $p(x)$  for 300 equally spaced values of  $x$ .

equally spaced points  $x(i)$  in the interval  $[-2, 2]$ . The logit  $f(x)$  has been estimated as  $\hat{f}(x)$  via penalized likelihood, and  $2\hat{p}(x) - 1$  is plotted as the dotted line, where  $\hat{p} = e^{\hat{f}} / (1 + e^{\hat{f}})$ . The dashed line gives the support vector machine estimate from the same data. It can be seen that the SVM is trying to estimate  $-1$  for  $x < 0$  and  $+1$  for  $x > 0$ , which is the Bayes optimal classifier here. A small Gibbs effect near the class boundary  $x = 0$  is evident, although the penalized likelihood and SVM will essentially pick out the same classification boundary. Further examples of this phenomenon in the context of the multicategory SVM of Lee, Lin and Wahba can be found in [5]. A comparative discussion of the multicategory SVM and a multicategory penalized likelihood estimate can be found in [16].

#### 5. SUPPORT VECTOR REGRESSION

A precursor of the  $\varepsilon$  insensitive loss function can be found in [15], where the loss function is  $L(y, f(\mathbf{x}_i)) = 0$  if  $|y - f| \leq \varepsilon$  and  $\infty$  otherwise. In 1969 only highly quantized data were available from satellites, but computation of such estimates was iffy.

#### 6. SPARSITY, VARIABLE SELECTION

In many classification problems, it is desirable to learn which components of the proposed attribute vector are actually contributing substantially to the actual classification. Two recent contributions are [4] and [18]. The trick is to add  $\ell_1$  (absolute value) penalties on coefficients of variables or terms in the penalty functional, which induces sparsity, as is well known. An early proponent of adding  $\ell_1$  penalties in classification algorithms to promote sparsity is [2]; there are many other recent related contributions. In practice the major challenge in many problems involves which attributes, or clusters of attributes, to put into the model to begin with. This challenge appears in images, sounds, handwriting, text, genomic data, meteorological data, astronomical data and elsewhere. Many open questions remain in particular contexts.

#### 7. REGULARIZED KERNELS FROM DISSIMILARITY DATA

Some recent work [9] focused on fitting kernels from noisy, scattered, incomplete dissimilarity data, which can then be used as a dimension reduction tool or in a SVM or multicategory SVM. Given a set of objects (protein sequences in [9]) and dissimilarity information  $d_{ij}$  between the  $i$ th and  $j$ th object, for a sufficiently

rich subset  $\Omega$  of the  $\binom{n}{2}$  pairs, one finds an  $n \times n$  kernel (nonnegative definite matrix)  $K_\mu$  over “object space” to yield

$$(1) \quad \min_{K \in \mathcal{S}} \sum_{ij \in \Omega} |d_{ij} - \hat{d}_{ij}| + \mu \text{trace } K,$$

where  $\mathcal{S}$  is the class of nonnegative definite  $n \times n$  matrices and  $\hat{d}_{ij} = K_\mu(i, i) + K_\mu(j, j) - 2K_\mu(i, j)$ . It is necessary to choose  $\mu$  and it is useful to truncate the eigenvalues of  $K_\mu$  after the first  $p$ , where  $p$  can be chosen so as to retain some specified percentage of the trace. Suppressing  $\mu$  and the truncation level  $p$ , a support vector machine  $f(i)$ ,  $i = 1, \dots, n$ , can be defined in object space as

$$f(i) = \sum_{\ell=1}^n c_\ell K(i, \ell)$$

by minimizing

$$\sum_{i=1}^n (1 - y_i f(i))_+ + \mu c' K^\dagger c$$

or its multicategory analog from [5].

To classify a new object ( $i = n + 1$ ), the “newbie” algorithm is used. It goes as follows: Given  $d_{i,n+1}$  for sufficiently many  $i$ , find  $b \in E^n$  and constant  $c$  to minimize

$$\sum_i |d_{i,n+1} - \hat{d}_{i,n+1}|$$

over  $b \in \text{range}(K)$  and  $c - b' K^\dagger b \geq 0$ . The  $b$  and  $c$  are used to give a new  $(n + 1) \times (n + 1)$  nonnegative definite matrix with  $K$  in the upper left block, and  $K(n + 1, n + 1) = c$ ,  $K(i, n + 1) = b_i$ ,  $i = 1, \dots, n$ , and  $\hat{d}_{ij} = K(n + 1, n + 1) + K(i, i) - 2K(i, n + 1)$ . Then the classifier evaluated at the  $(n + 1)$ st object is

$$f(n + 1) = \sum_{\ell=1}^n c_\ell K(n + 1, \ell).$$

Pseudo-attribute vectors may be defined as  $\mathbf{x}(i) = (\sqrt{\lambda_1} \phi_1(i), \dots, \sqrt{\lambda_p} \phi_p(i))$ , where the  $\{\lambda_\nu, \phi_\nu\}$  are the eigenvalues and eigenvectors of  $K$ . The newbie can be placed in this pseudo-attribute coordinate system by using its fitted distance from a sufficiently large subset of the fitted training set distances. Since  $K(i, j) = (\mathbf{x}(i), \mathbf{x}(j))$ , the resulting SVM is linear in the pseudo-attribute vectors. However, other SVMs can be built on the labeled pseudo-attribute vectors.

The so-called semisupervised version of this problem occurs when only some of the original training

objects are labeled. Thus, there are three kinds of objects: (1) those that are in the training set and labeled; (2) those that are in the training set and not labeled, but are used to determine the geometry of the object space; and (3) unlabeled newbies. Both kinds of unlabeled data can then be classified by the SVM.

The tuning parameter  $\mu$  in equation (1) can be tuned by leaving out pairs of objects (CV2) and comparing their observed distances with their fitted (pseudo-attribute) distances for a range of  $\mu$ .

## 8. ROBUST MANIFOLD UNROLLING

A related problem occurs when the objects of interest are believed to lie in a low-dimensional (nonlinear) manifold in some higher-dimensional space. Here then it is desired to “flatten out” the manifold and reduce the dimension before carrying out a classification or regression operation. Recent references can be found in [10], where we proposed an approach related to that in equation (1) with two modifications: (a) only distances between  $k$  nearest neighbors will be used and (b)  $\mu \text{trace } K$  is replaced by  $-\mu \text{trace } K$ . The effect on the resulting pseudo-attribute vectors is that they tend to “flatten out” or “unroll” due to the fact that only nearest neighbor distances are used, as well as the fact that the minus sign propels distant objects to become more distant. A longer discussion of the rationale behind this algorithm and demonstrations of its behavior are found in [10]. The semisupervised version of this problem can be defined similarly, with many potential applications. Both of these optimization problems can be solved numerically using convex cone optimization code.

## 9. WHERE ARE WE GOING?

The theory, computation and application of classification problems that relate to support vector machines and other regularization based classifiers is by no means finished work, although the extent of work so far is breathtaking. Many problems remain. Using subject matter knowledge to build kernels that embody subject matter information efficiently in various fields remains an interesting challenge. For example, text and language processing have interesting problems that involve complex relationships between components of text. Huge attribute vectors and small training sets as occur in genetic data of various kinds present their own challenges, as does the merging of heterogenous kinds of information. Multiple correlated inputs and outputs provide challenges. Improved systematic ways to

choose important attributes or groups of attributes remain to be found. As the authors note, the relationships between statistical learning theory based on Vapnik–Chervonenkis dimension and SVM theory based on regularization remain to be understood better, as do regularization based approaches and other approaches to classification. Collaboration between statisticians, computer scientists, mathematicians and subject matter experts will no doubt be needed for many of the practical challenges.

### ACKNOWLEDGMENTS

This research was supported by NSF Grant DMS-00-7292, by ONR Grant NN00140610 095 and by NIH Grant EY09946.

### REFERENCES

- [1] ARONSZAJN, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc.* **68** 337–404. [MR0051437](#)
- [2] GUNN, S. and KANDOLA, J. (2002). Structural modelling with sparse kernels. *Machine Learning* **48** 137–163.
- [3] HALMOS, P. (1957). *Introduction to Hilbert Space and the Theory of Spectral Multiplicity*. Chelsea, New York. [MR1653399](#)
- [4] LEE, Y., KIM, Y., LEE, S. and KOO, J.-Y. (2005). Structured multicategory support vector machine with ANOVA decomposition. Technical Report 743rr, Dept. Statistics, Ohio State Univ.
- [5] 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. [MR2054287](#)
- [6] LIN, Y. (2001). A note on margin-based loss functions in classification. *Statist. Probab. Lett.* **68** 73–82. [MR2064687](#)
- [7] LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **6** 259–275. [MR1917926](#)
- [8] LIN, Y., LEE, Y. and WAHBA, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning* **46** 191–202.
- [9] LU, F., KELEŞ, S., WRIGHT, S. and WAHBA, G. (2005). A framework for kernel regularization with application to protein clustering. *Proc. Natl. Acad. Sci. USA* **102** 12,332–12,337. [MR2168854](#)
- [10] LU, F., LIN, Y. and WAHBA, G. (2005). Robust manifold unfolding with kernel regularization. Technical Report 1108, Dept. Statistics, Univ. Wisconsin.
- [11] MARRON, J. and TODD, M. (2002). Distance weighted discrimination. Technical Report, School Operations Res. and Industrial Engineering, Cornell Univ.
- [12] O’SULLIVAN, F., YANDELL, B. and RAYNOR, W. (1986). Automatic smoothing of regression functions in generalized linear models. *J. Amer. Statist. Assoc.* **81** 96–103. [MR0830570](#)
- [13] RIESZ, F. and NAGY-SZ., B. (1955). *Functional Analysis*. Ungar, New York. [MR0071727](#)
- [14] SHEN, X., TSENG, G., ZHANG, X. and WONG, W. H. (2003). On  $\psi$ -learning. *J. Amer. Statist. Assoc.* **98** 724–734. [MR2011686](#)
- [15] WAHBA, G. (1969). Estimating derivatives from outer space. Technical Report 989, Mathematics Research Center, Univ. Wisconsin. Available at [www.stat.wisc.edu/~wahba/ftp1/oldie/989.pdf](http://www.stat.wisc.edu/~wahba/ftp1/oldie/989.pdf).
- [16] WAHBA, G. (2002). Soft and hard classification by reproducing kernel Hilbert space methods. *Proc. Natl. Acad. Sci. USA* **99** 16,524–16,530. [MR1947755](#)
- [17] XIE, X. (2005). Smoothing in magnetic resonance image analysis and a hybrid loss for support vector machine. Ph.D. dissertation, Technical Report 1110, Dept. Statistics, Univ. Wisconsin.
- [18] ZHANG, H. H. (2006). Variable selection for support vector machines via smoothing spline ANOVA. *Statist. Sinica* **16** 659–674.
- [19] ZHANG, H. H., AHN, J., LIN, X. and PARK, C. (2006). Gene selection using support vector machines with nonconvex penalty. *Bioinformatics* **22** 88–95.

# Comment

Trevor Hastie and Ji Zhu

We congratulate the authors for a well written and thoughtful survey of some of the literature in this area. They are mainly concerned with the geometry and the computational learning aspects of the support vector machine (SVM). We will therefore complement their review by discussing from the statistical function estimation perspective. In particular, we will elaborate on the following points:

- Kernel regularization is essentially a generalized ridge penalty in a certain feature space.
- In practice, the effective dimension of the data kernel matrix is not always equal to  $n$ , even when the implicit dimension of the feature space is infinite; hence, the training data are not always perfectly separable.
- Appropriate regularization plays an important role in the success of the SVM.
- The SVM is not fundamentally different from many statistical tools that our statisticians are familiar with, for example, penalized logistic regression.

We acknowledge that many of the comments are based on our earlier paper Hastie, Rosset, Tibshirani and Zhu (2004).

## KERNEL REGULARIZATION AND THE GENERALIZED RIDGE PENALTY

Given a positive definite kernel  $K(\mathbf{x}, \mathbf{x}')$ , where  $\mathbf{x}, \mathbf{x}'$  belong to a certain domain  $\mathcal{X}$ , we consider the general function estimation problem

$$(1) \quad \min_{\beta_0, f} \sum_{i=1}^n \ell(y_i, \beta_0 + f(\mathbf{x}_i)) + \frac{\lambda}{2} \|f(\mathbf{x})\|_{\mathcal{H}_K}^2.$$

Here  $\ell(\cdot, \cdot)$  is a convex loss function that describes the “closeness” between the observed data and the fitted model, and  $f$  is an element in the span of  $\{K(\cdot, \mathbf{x}'), \mathbf{x}' \in \mathcal{X}\}$ . More precisely,  $f \in \mathcal{H}_K$  is a function in the

---

*Trevor Hastie is Professor, Department of Statistics, Stanford University, Stanford, California 94305, USA (e-mail: hastie@stanford.edu). Ji Zhu is Assistant Professor, Department of Statistics, University of Michigan, Ann Arbor, Michigan 48109, USA (e-mail: jizhu@umich.edu).*

reproducing kernel Hilbert space  $\mathcal{H}_K$  (RKHS) generated by  $K(\cdot, \cdot)$  (see Burges, 1998; Evgeniou, Pontil and Poggio, 2000; and Wahba, 1999, for details).

Suppose the positive definite kernel  $K(\cdot, \cdot)$  has a (possibly finite) eigenexpansion,

$$K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} \delta_j \phi_j(\mathbf{x}) \phi_j(\mathbf{x}'),$$

where  $\delta_1 \geq \delta_2 \geq \dots \geq 0$  are the eigenvalues and  $\phi_j(\mathbf{x})$ 's are the corresponding eigenfunctions. Elements of  $\mathcal{H}_K$  have an expansion in terms of these eigenfunctions

$$(2) \quad f(\mathbf{x}) = \sum_{j=1}^{\infty} \beta_j \phi_j(\mathbf{x}),$$

with the constraint that

$$\|f\|_{\mathcal{H}_K}^2 \stackrel{\text{def}}{=} \sum_{j=1}^{\infty} \beta_j^2 / \delta_j < \infty,$$

where  $\|f\|_{\mathcal{H}_K}$  is the norm induced by  $K(\cdot, \cdot)$ .

Then we can rewrite (1) as

$$(3) \quad \min_{\beta_0, \beta} \sum_{i=1}^n \ell\left(y_i, \beta_0 + \sum_{j=1}^{\infty} \beta_j \phi_j(\mathbf{x}_i)\right) + \lambda \sum_{j=1}^{\infty} \frac{\beta_j^2}{\delta_j},$$

and we can see that the regularization term  $\|f\|_{\mathcal{H}_K}^2$  in (1) can be interpreted as a generalized ridge penalty, where eigenfunctions with small eigenvalues in the expansion (2) get penalized more and vice versa.

Formulation (3) seems to be an infinite dimensional optimization problem, but according to the representer theorem (Kimeldorf and Wahba, 1971; Wahba 1990), the solution is finite dimensional and has the form

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}, \mathbf{x}_i).$$

Using the reproducing property of  $\mathcal{H}_K$ , that is,  $\langle K(\cdot, \mathbf{x}_i), K(\cdot, \mathbf{x}_{i'}) \rangle = K(\mathbf{x}_i, \mathbf{x}_{i'})$ , (3) also reduces to a finite-dimensional criterion,

$$(4) \quad \min_{\beta_0, \alpha} L(\mathbf{y}, \beta_0 + \mathbf{K}\alpha) + \lambda \alpha^T \mathbf{K}\alpha.$$

Here we use vector notation,  $\mathbf{K}$  is the  $n \times n$  data kernel matrix with elements equal to  $K(\mathbf{x}_i, \mathbf{x}_{i'})$ ,  $i, i' =$

$1, \dots, n$ , and  $L(\mathbf{y}, \beta_0 + \mathbf{K}\alpha) = \sum_{i=1}^n \ell(y_i, \beta_0 + f(\mathbf{x}_i))$ . We reparametrize (4) using the eigendecomposition of  $\mathbf{K} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ , where  $\mathbf{D}$  is diagonal and  $\mathbf{U}$  is orthogonal. Let  $\mathbf{K}\alpha = \mathbf{U}\beta$ , where  $\beta = \mathbf{D}\mathbf{U}^T\alpha$ . Then (4) becomes

$$(5) \quad \min_{\beta_0, \beta} L(\mathbf{y}, \beta_0 + \mathbf{U}\beta) + \lambda\beta^T\mathbf{D}^{-1}\beta.$$

Now the columns of  $\mathbf{U}$  are unit-norm basis functions that span the column space of  $\mathbf{K}$ ; and again, we see that those members that correspond to small eigenvalues (the elements of the diagonal matrix  $\mathbf{D}$ ) get heavily penalized and vice versa.

In the machine learning community, people tend to view the kernel as providing an implicit map of  $\mathbf{x}$  from  $\mathcal{X}$  to a certain high-dimensional feature space, and  $K(\cdot, \cdot)$  computes inner products in this (possibly infinite-dimensional) feature space. Specifically, the features are

$$h_j(\mathbf{x}) = \sqrt{\delta_j}\phi_j(\mathbf{x}) \quad \text{or} \quad \mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \dots)^T,$$

and we have

$$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{h}(\mathbf{x}), \mathbf{h}(\mathbf{x}') \rangle.$$

Furthermore, let  $\theta_j = \beta_j/\sqrt{\delta_j}$  and  $\mathbf{H} = \mathbf{U}\mathbf{D}^{1/2}$ . Then (3) and (5) become

$$(6) \quad \min_{\beta_0, \theta} \sum_{i=1}^n \ell\left(y_i, \sum_{j=1}^{\infty} \theta_j h_j(\mathbf{x})\right) + \lambda \sum_{j=1}^{\infty} \theta_j^2$$

and

$$(7) \quad \min_{\beta_0, \theta} L(\mathbf{y}, \beta_0 + \mathbf{H}\theta) + \lambda\theta^T\theta,$$

respectively. This shows kernel regularization as an exact ridge penalty in the feature space, but unlike (3) and (5), it hides the fact that eigenfunctions are differentially penalized according to their corresponding eigenvalues.

To illustrate the point, we consider a simple example. The data  $x_i$ 's are one-dimensional and were generated from the standard Gaussian distribution ( $n = 50$ ). The radial kernel function  $K(x, x') = \exp(-\gamma\|x - x'\|^2)$  was used, with  $\gamma = 1$ . Figure 1 shows the eigenvalues of the kernel matrix  $\mathbf{K}$ . The left panel of Figure 2 shows the first 16 eigenvectors of  $\mathbf{K}$  (columns of  $\mathbf{U}$ ) and the right panel shows the corresponding features (columns of  $\mathbf{H}$ ). As we can see from the left panel of Figure 2, eigenvectors with large eigenvalues (hence get penalized less) tend to be smooth, while eigenvectors with small eigenvalues (hence get penalized more) tend to be wiggly; therefore,  $\|f\|_{\mathcal{H}_K}^2$  is also always interpreted as a roughness measure of the function  $f$ . We can also see from the right panel of Figure 2 that many of the features are “norm challenged,” that is, they are squashed down dramatically by their eigenvalues.

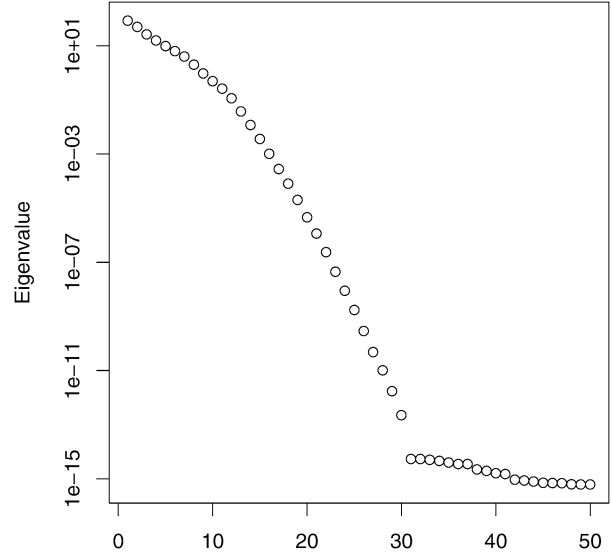


FIG. 1. Eigenvalues (on the log scale) of the data kernel matrix  $\mathbf{K}$ .

### EFFECTIVE DIMENSION OF THE DATA KERNEL MATRIX

As we have seen in the previous section, the kernel  $K(\cdot, \cdot)$  maps  $\mathbf{x}$  from its original input space to some high-dimensional feature space  $\mathbf{h}(\mathbf{x})$ . In the case of classification, it is sometimes argued that the implicit feature space can be infinite-dimensional (e.g., via the radial basis kernel), which suggests that perfect separation of the training data is always possible. However, this is not always true in practice.

To illustrate the point, we consider a two-class classification example. The data were generated from a pair of mixture Gaussian densities, described in detail by Hastie, Tibshirani and Friedman (2001). The radial kernel function  $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$  was used. Four different values of  $\gamma$  (0.1, 0.5, 1 and 5) were tried. For each of the values of  $\gamma$ , the SVM was fitted for a sequence of values of  $\lambda$ , ranging from the most regularized model to the least regularized model.

We were at first surprised to discover that not all these sequences achieved zero training errors on the 200 training data points, at their least regularized fit. The minimal training errors and the corresponding values for  $\gamma$  are summarized in Table 1. The second row of the table shows the effective rank of the data kernel matrix  $\mathbf{K}$  (which we defined to be the number of eigenvalues greater than  $10^{-12}$ ). This  $200 \times 200$  matrix has elements  $K(\mathbf{x}_i, \mathbf{x}_{i'})$ ,  $i, i' = 1, \dots, 200$ . In this example, a full rank  $\mathbf{K}$  is required to achieve perfect separation. Similar observations have also appeared in Williams and Seeger (2000) and Bach and Jordan (2002).

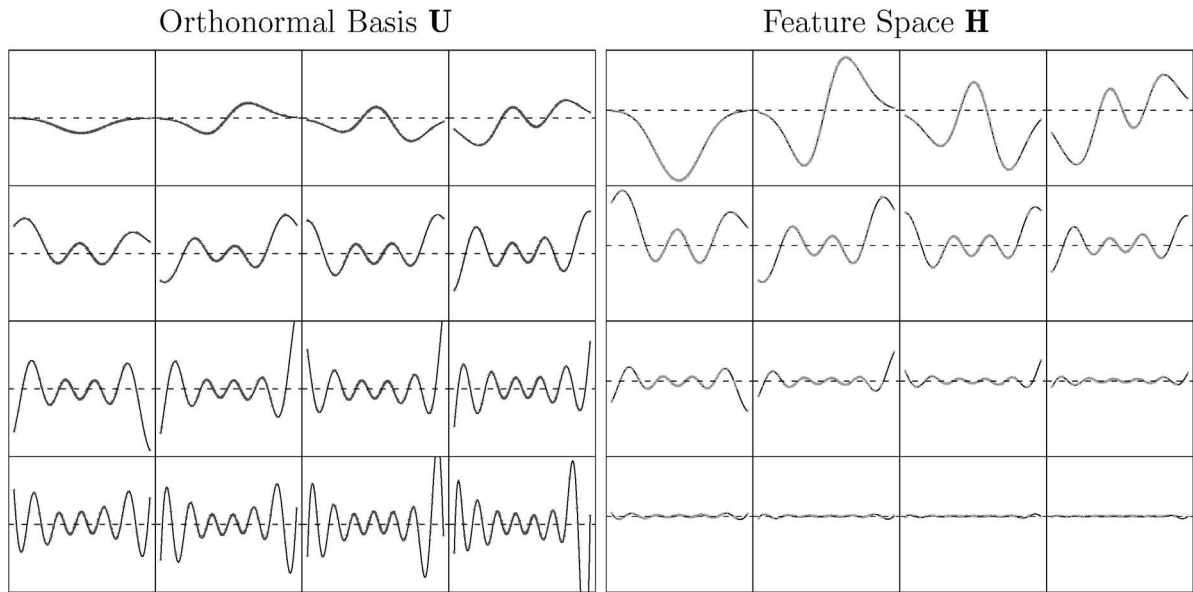


FIG. 2. The left panel shows the eigenvectors of the data kernel matrix  $\mathbf{K}$  and the right panel shows the corresponding features (eigenvectors scaled by the corresponding eigenvalues).

TABLE I  
Results for the mixture simulation example

$\gamma$	5	1	0.5	0.1
Training errors	0	12	21	33
Effective rank	200	177	143	76

This emphasizes again the fact that not all features in the feature map implied by  $K(\cdot, \cdot)$  are of equal stature (see the right panel in Figure 2); many of them are

shrunk way down to zero. The regularization in (3) and (5) penalizes unit-norm eigenvectors by the inverse of their eigenvalues, which effectively annihilates some, depending on  $\gamma$ . Small  $\gamma$  implies wide, flat kernels and a suppression of wiggly, rough functions. Figure 3 shows the eigenvalues of  $\mathbf{K}$  for the four values of  $\gamma$ . The larger eigenvalues correspond in this case to smoother eigenfunctions, the smaller ones to rougher. The rougher eigenfunctions get penalized exponentially more than the smoother ones. Hence, for smaller values of  $\gamma$ , the effective dimension of the feature space is truncated.

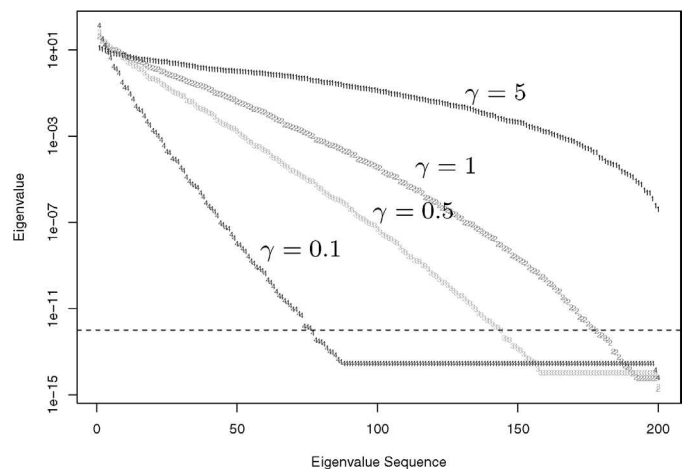


FIG. 3. The eigenvalues (on the log scale) for the data kernel matrices  $\mathbf{K}$  that correspond to the four values of  $\gamma$ .

## THE NEED FOR CAREFUL REGULARIZATION

The SVM has been very successful for the classification problem and gained a lot of attention in the machine learning community in the past ten years. Many papers have been published to explain why it performs so well. Most of this literature concentrates on the concept of margin. Various misclassification error bounds have been derived based on the margin (Vapnik, 1995; Bartlett and Shawe-Taylor, 1999; Shawe-Taylor and Cristianini, 1999).

However, our view is a little different from that based on the concept of margin. Several researchers have noted the relationship between the SVM and regularized function estimation in RKHS (Evgeniou, Pontil and Poggio, 2000; Wahba, 1999). The regularized function estimation problem contains two parts: a loss function and a penalty term [e.g., (1)]. The SVM uses the so-called hinge loss function (see Figure 5). The margin maximizing property of the SVM derives from the hinge loss function. Hence margin maximization is by nature a nonregularized objective, and solving it in high-dimensional space is likely to lead to overfitting and bad prediction performance. This has been observed in practice by many researchers, in particular Breiman (1999) and Marron and Todd (2002).

The *loss + penalty* formulation emphasizes the role of regularization. In many situations we have sufficient features (e.g., gene expression arrays) to guarantee separation. We may nevertheless avoid the maximum margin separator ( $\lambda \downarrow 0$ ) in favor of a more regularized solution.

Figure 4 shows the test error as a function of  $\lambda$  for the mixture data example. Here we see a dramatic range in the correct choice of  $\lambda$ . When  $\gamma = 5$ , the most regularized model is called for. On the other hand, when  $\gamma = 0.1$ , we would want to choose among the least regularized models. Depending on the value of  $\gamma$ , the optimal  $\lambda$  can occur at either end of the spectrum or anywhere in between, emphasizing the need for careful selection.

## CONNECTION WITH OTHER STATISTICAL TOOLS

Last, we would like to comment on the connection between the SVM and some statistical tools that statisticians are familiar with.

As we have seen in previous sections, what is special with the SVM is not the regularization term, but is rather the loss function, that is, the hinge loss. Lin (2002) pointed out that the hinge loss is Bayes consistent, that is, the population minimizer of the loss function agrees with the Bayes rule in terms of classification. This is important in explaining the success of the SVM, because it implies that the SVM is trying to implement the Bayes rule.

On the other hand, notice that the hinge loss and the binomial deviance have very similar shapes (see Figure 5): both increase linearly as  $yf$  gets very small (negative), and both encourage  $y$  and  $f$  to have the same sign. Hence it is reasonable to conjecture that by replacing the hinge loss of the SVM with the binomial

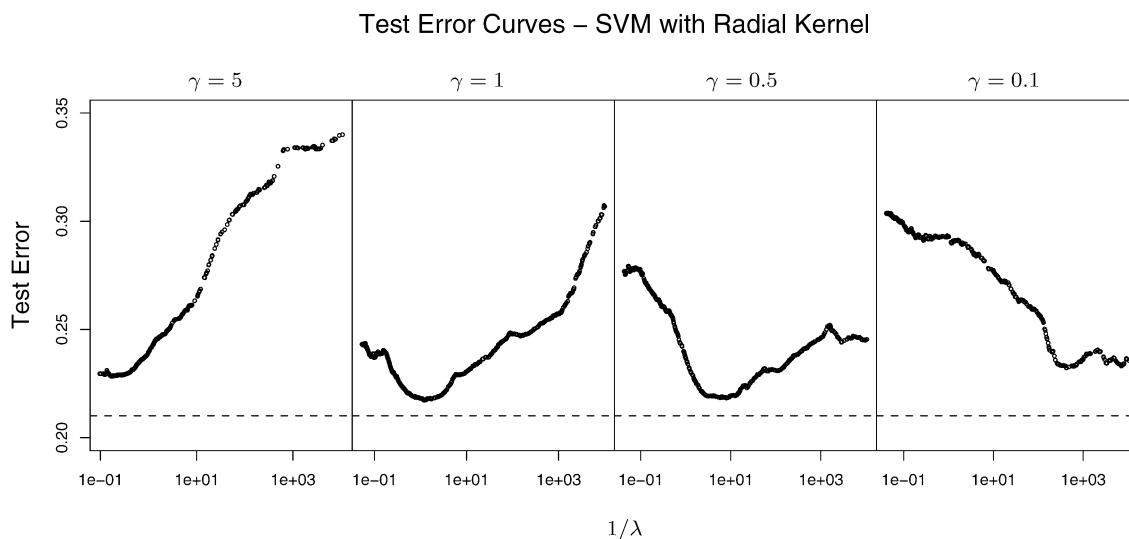


FIG. 4. Test error curves for the mixture example, using four different values for the radial kernel parameter  $\gamma$ . Large values of  $\lambda$  correspond to heavy regularization, small values of  $\lambda$  to light regularization. Depending on the value of  $\gamma$ , the optimal  $\lambda$  can occur at either end of the spectrum or anywhere in between, emphasizing the need for careful selection.

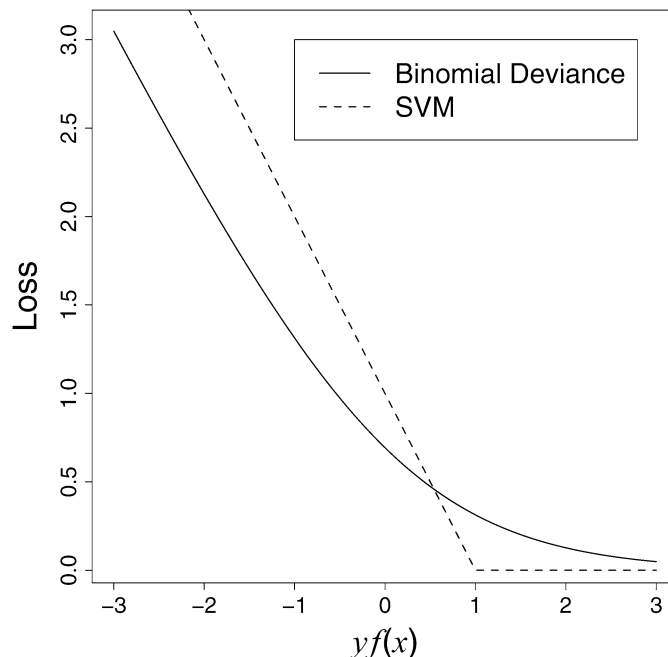


FIG. 5. Comparing the hinge loss and the binomial deviance,  $y \in \{-1, 1\}$ .

deviance, which is also Bayes consistent, we should be able to get a fitted model that performs similarly to the SVM. In fact, in Zhu and Hastie (2005), we show that under certain conditions, the classification boundary of the resulting penalized logistic regression (using the binomial deviance) and that of the SVM coincide. Penalized logistic regression has been studied by many statisticians (see Green and Silverman, 1994; Wahba, Gu, Wang and Chappell, 1995; and Lin et al., 2000, for details). We understand why it can work well. The same reasoning could be applied to the SVM.

Penalized logistic regression is not the only model that performs similarly to the SVM; replacing the hinge loss with any sensible loss function will give a similar result, for example, the exponential loss function of boosting (Freund and Schapire, 1997) and the squared error loss (Zhang and Oles, 2001; Bühlmann and Yu, 2003). These loss functions are all Bayes consistent. The binomial deviance and the exponential loss are margin-maximizing loss functions, but the squared error loss is not. The distance weighted discrimination (Marron and Todd, 2002) is designed specifically for *not* maximizing the margin and works well with high-dimensional data, which in a way also justifies that margin maximization is not the key to the success of the SVM.

## ACKNOWLEDGMENTS

Hastie is partially supported by NSF Grant DMS-05-05676. Zhu is partially supported by NSF Grant DMS-05-05432.

## REFERENCES

- BACH, F. and JORDAN, M. (2002). Kernel independent component analysis. *J. Mach. Learn. Res.* **3** 1–48. [MR1966051](#)
- BARTLETT, P. and SHAWE-TAYLOR, J. (1999). Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 43–54. MIT Press, Cambridge, MA.
- BREIMAN, L. (1999). Prediction games and arcing algorithms. *Neural Computation* **11** 1493–1517.
- BÜHLMANN, P. and YU, B. (2003). Boosting with the  $L_2$  loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–340. [MR1995709](#)
- BURGES, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **2** 121–167.
- EVGENIOU, T., PONTIL, M. and POGGIO, T. (2000). Regularization networks and support vector machines. *Adv. Comput. Math.* **13** 1–50. [MR1759187](#)
- FREUND, Y. and SCHAPIRE, R. (1997). A decision theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139. [MR1473055](#)
- GREEN, P. and SILVERMAN, B. (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman and Hall, London. [MR1270012](#)
- 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.

- HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*. Springer, New York. [MR1851606](#)
- KIMELDORF, G. and WAHBA, G. (1971). Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.* **33** 82–95. [MR0290013](#)
- LIN, X., WAHBA, G., XIANG, D., GAO, F., KLEIN, R. and KLEIN, B. (2000). Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *Ann. Statist.* **28** 1570–1600. [MR1835032](#)
- LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Min. Knowl. Discov.* **6** 259–275. [MR1917926](#)
- MARRON, J. and TODD, M. (2002). Distance weighted discrimination. Technical report, School Operations Res. and Industrial Engineering, Cornell Univ.
- SHAWE-TAYLOR, J. and CRISTIANINI, N. (1999). Margin distribution bounds on generalization. In *Computational Learning Theory. Lecture Notes in Artificial Intelligence* **1572** 263–273. Springer, Berlin.
- VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York. [MR1367965](#)
- WAHBA, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia. [MR1045442](#)
- WAHBA, G. (1999). Support vector machines, reproducing kernel Hilbert space and randomized GACV. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 69–88. MIT Press, Cambridge, MA.
- WAHBA, G., GU, C., WANG, Y. and CHAPPELL, R. (1995). Soft classification, a.k.a. risk estimation, via penalized log likelihood and smoothing spline analysis of variance. In *The Mathematics of Generalization* (D. Wolpert, ed.) 329–360. Addison–Wesley, Reading, MA.
- WILLIAMS, C. and SEEGER, M. (2000). The effect of the input density distribution on kernel-based classifiers. In *Proc. Seventeenth International Conference on Machine Learning* 1159–1166. Morgan Kaufmann, San Francisco.
- ZHANG, T. and OLES, F. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval* **4** 5–31.
- ZHU, J. and HASTIE, T. (2005). Kernel logistic regression and the import vector machine. *J. Comput. Graph. Statist.* **14** 185–205. [MR2137897](#)

# Rejoinder

Javier M. Moguerza and Alberto Muñoz

## 1. INTRODUCTION

We are very grateful to the Executive Editors George Casella and Edward George for their active interest in our paper and for organizing this challenging discussion. We also thank all the discussants for their insightful and stimulating comments.

When we submitted the original manuscript in 2003, we were tempted to go for a more general paper on kernel methods. We decided to focus on support vector machines (SVMs), waiting for a mature development of the new and exciting ideas related to kernel methods, such as manifold learning and other related topics. We will refer to some of these methods below. Let us begin, first, with some general considerations.

Regarding the question of the dimensionality induced by the feature space, Hastie and Zhu remark in their comment that usual kernels do not automatically lead to infinite-dimensional feature spaces. They give a nice example that involves the radial (Gaussian) kernel function. This agrees with results in Keerthi and Lin [8], where an explanation of the performance of the Gaussian kernel is given when, according to the notation in the comment by Hastie and Zhu,  $\gamma \rightarrow 0$  and  $\lambda$  is chosen in the appropriate way. In this case, the SVM classifier converges to a linear SVM classifier, and the effective dimension of the kernel is finite, agreeing with the empirical conclusion provided by the discussants.

We also agree with the assertions of some of the discussants regarding the probabilistic interpretability of the SVM output (the sign of some estimated function). Our comment was rather along the line of Sollich [18], who proposed to make Bayesian methods available for the support vector methodology, while leaving as much as possible of the standard SVM framework intact. This is not an easy task. In fact, as Bartlett, Jordan and McAuliffe remark, sparseness and the precise estimation of conditional probabilities are hard to reconcile.

Regarding the role of differentiability in SVMs (misplaced in the opinion of Bartlett, Jordan and McAuliffe), it is convenient to recall that the differentiable formulation of the SVM problem allows its solution by the use of standard Newton-type methods

for convex optimization. Under the availability of second order derivatives (and this is the case for SVMs), these methods are known to be the most efficient ones for the solution of smooth problems.

We thank some of the discussants for turning the attention of the reader to general kernel methods. In particular, we appreciate the Bartlett, Jordan and McAuliffe effort to make clearer the potential impact of reproducing kernel Hilbert space (RKHS) methods. Regarding the origins of RKHS in statistics, for the sake of completeness, we strongly recommend reading the conversation with Emanuel Parzen in [14].

Given the history of SVMs, perfectly outlined by Wahba in the introduction of her comment, we do not like to think of SVMs as a “modest” variant of some standard statistical methodology (as suggested by Bartlett, Jordan and McAuliffe). Using a similar (a posteriori) reasoning, some strict mathematicians might think that RKHS methods in statistics are just a small variation on the general theory of Hilbert spaces. Of course, this is far from true. We rather think that the support vector methodology, followed closely by kernel methods, has been able to synthesize a variety of techniques from different fields, leading to a more unified framework for learning theory [5]. In addition, the geometrical viewpoint of SVMs allows new approaches to long-familiar problems, as illustrated in the next section.

## 2. KERNEL METHODS REVISITED

One interesting point regarding the geometrical interpretation of SVMs is that they have stirred the development of new techniques driven by the geometrical properties of the kernel. Some of these techniques have not so far been mentioned in the discussion. We now briefly describe two relevant examples.

### 2.1 One-Class SVMs

An example of a new method that has arisen from a geometrical point of view is one-class SVMs [16]. One-class SVMs deal with a problem related to estimating high density regions from data samples. The method computes a binary function that takes the value +1 in “small” regions that contain most data points and

takes the value  $-1$  elsewhere. The strategy of the one-class support vector method is to map the data points into the feature space determined by a kernel function and to calculate a hyperplane that separates the mapped data  $\{\Phi(\mathbf{x}_i)\}_{i=1}^n$  from the origin, where  $\Phi$  is the mapping induced by the kernel function. With this aim, the one-class SVM algorithm solves the quadratic optimization problem

$$(2.1) \quad \begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 - b + \frac{1}{\nu n} \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & \mathbf{w}^T \Phi(\mathbf{x}_i) \geq b - \xi_i, \quad i = 1, \dots, n, \\ & \xi_i \geq 0, \quad i = 1, \dots, n, \end{aligned}$$

where  $\xi_i$  are slack variables,  $\nu \in [0, 1]$  is an a priori fixed constant which represents the fraction of outlying points and  $b$  is the decision value which determines whether a given point belongs to the estimated high density region. The decision function will take the form  $h(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \Phi(\mathbf{x}) - b^*)$ , where  $\mathbf{w}^*$  and  $b^*$  are the values of  $\mathbf{w}$  and  $b$  at the solution of problem (2.1). The hyperplane  $\mathbf{w}^{*T} \Phi(\mathbf{x}) - b^* = 0$  separates from the origin the mapped data for which the decision function  $h(\mathbf{x}) = +1$ . Problem (2.1) is smooth and convex, and follows the SVM idea of building a hyperplane in a feature space.

It is apparent that solving the problem of estimating high density regions by building a separating hyperplane in a feature space is not trivial. Next, we provide an original statistical explanation of one-class SVMs. Consider the class of real-valued functions

$$(2.2) \quad \begin{aligned} \mathcal{G} = \{g > 0 \mid \forall \mathbf{x}, \mathbf{y} \in X, \\ g(\mathbf{x}) > g(\mathbf{y}) \iff f(\mathbf{x}) > f(\mathbf{y})\}, \end{aligned}$$

where  $X$  is the input space and  $f$  is the data density function. To estimate the outlying points that correspond to the proportion  $\nu$ , all we have to do is use the order induced by any function  $g \in \mathcal{G}$  on the data sample  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . This is equivalent to solving the optimization problem

$$(2.3) \quad \begin{aligned} \max_{\lambda} \quad & - \sum_{i=1}^n \lambda_i g(\mathbf{x}_i) \\ \text{s.t.} \quad & \sum_{i=1}^n \lambda_i = 1, \\ & 0 \leq \lambda_i \leq \frac{1}{\nu n}, \quad i = 1, \dots, n, \end{aligned}$$

where, at the solution  $\lambda^* = (\lambda_1^*, \dots, \lambda_n^*)^T$ ,  $\lambda_i^* > 0$  if  $\mathbf{x}_i$  is an outlying point [i.e.,  $\lambda_i^* > 0$  for small values of

$g(\mathbf{x}_i)$ ] and  $\lambda_i^* = 0$  otherwise. The dual of this linear problem is

$$(2.4) \quad \begin{aligned} \min_{b, \xi} \quad & -b + \frac{1}{\nu n} \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & g(\mathbf{x}_i) \geq b - \xi_i, \quad i = 1, \dots, n, \\ & \xi_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

It can be shown that, at the solution, the values of the objective functions of problems (2.3) and (2.4) coincide (see, e.g., [1]). Moreover, the solution of problem (2.3) can be straightforwardly calculated from the solution of problem (2.4).

The one-class SVM problem (2.1) and problem (2.4) are very similar. It becomes clear now that the solution of problem (2.1) (the one-class SVM solution) tries to estimate a function  $g \in \mathcal{G}$  by  $\hat{g}(\mathbf{x}) = \mathbf{w}^{*T} \Phi(\mathbf{x})$ , that is, by estimating in the feature space the weights  $\mathbf{w}^*$  of a hyperplane with minimum norm. This is achieved through the inclusion of the term  $1/2 \|\mathbf{w}\|^2$  in the objective function of problem (2.1).

Appropriate mappings and kernels to solve the problem of estimating high density regions (density level sets) using one-class SVMs are derived and can be consulted in [13].

## 2.2 Combination of Kernels

Another example of a technique developed from geometric considerations of the kernel is now illustrated. This method falls in the category of “further advances” mentioned by Bousquet and Schölkopf at the end of their comment. In particular, we build, for classification purposes, what they call a joint kernel (mixing inputs and outputs). This joint kernel is built by the combination of a set of kernels. A key point of our proposal is that the constructed kernel tries to capture the “right” notion of similarity. This agrees with the comment by Bousquet and Schölkopf about the relationship between the good performance of SVMs in practice and the appropriate prior knowledge about the problems incorporated by kernels. Thus, we will work with similarity matrices instead of kernel matrices. In fact, as Wahba points out, Euclidean distances (and therefore similarities) can be derived from positive definite kernels.

The idea, in geometric terms, is introduced next. If kernels are being used, points in a sufficiently small neighborhood in the feature space should belong to the same class (excluding points very close to the decision surface). As a consequence, if we are going to classify a data set by relying on a given similarity matrix, points close to each other using such similarities

should, in general, be in the same class. Therefore, we have to construct a similarity matrix  $K^*$  with entries  $K^*(x_i, x_j)$  that are large for  $x_i$  and  $x_j$  in the same class (i.e.,  $y_i = y_j$ ), and small for  $x_i$  and  $x_j$  in different classes. For instance, if two kernels  $K_1$  and  $K_2$  are to be combined, a possible choice is

$$(2.5) \quad K^*(x_i, x_j) = \begin{cases} \max(K_1(x_i, x_j), K_2(x_i, x_j)), & \text{if } y_i = y_j, \\ \min(K_1(x_i, x_j), K_2(x_i, x_j)), & \text{otherwise.} \end{cases}$$

It is immediate to show that (2.5) is equivalent to

$$(2.6) \quad K^* = \frac{1}{2}(K_1 + K_2) + \frac{1}{2}Y|K_1 - K_2|Y,$$

where  $Y = \text{diag}(y)$  is a diagonal matrix whose nonzero elements are the data labels, that is,  $y_i \in \{-1, +1\}$ .

Let  $K_1, K_2, \dots, K_M$  be the available set of  $M$  input kernel matrices, all of which are obtained from the same data sample  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . The extension of the previous idea to the combination of more than two kernel matrices is

$$(2.7) \quad K^* = \bar{K} + Y \sum_{i < j} g(K_i - K_j)Y,$$

where  $\bar{K}$  is the average of the kernel matrices and  $g$  is a function that quantifies the difference of information between kernel matrices. The function  $g$  must have the property that if  $K_i$  and  $K_j$  tend to produce the same classification results, then  $g(K_i - K_j)$  should be almost null. A particular case of the previous equation is

$$(2.8) \quad K^* = \bar{K} + Y \sum_{m=1}^M |K_m - \bar{K}|Y.$$

This and other choices for  $g$  can be consulted in [12].

Next, we show how this method (denoted AV for absolute value) can be used to improve the performance of single kernels. With this aim, we will use the breast cancer data set, made up of 683 observations with 9 features each [11]. We have considered three kernels: a polynomial kernel  $K_1(x, z) = (1 + x^T z)^2$ , a Gaussian kernel  $K_2(x, z) = \exp(-\|x - z\|^2)$  and a linear kernel  $K_3(x, z) = x^T z$ . We will compare SVMs using these kernels with the AV combination method and a semidefinite programming (SDP) technique for building linear combinations of kernels developed by Lanckriet et al. [9]. The data set has been randomly partitioned ten times into a training set and a test set, and for each method, a run of the experiment has been done over each partition. The average results are shown

TABLE 1  
Percentage of misclassified data and support vectors for the cancer data. Standard deviations in parentheses

	Training error	Test error	% SV
$K_1$ : Polynomial	0.1 (0.1)	7.8 (2.5)	8.3 (0.8)
$K_2$ : Gaussian	0.0 (0.0)	10.8 (1.7)	65.6 (1.0)
$K_3$ : Linear	2.6 (0.5)	3.7 (1.8)	7.1 (0.8)
AV	2.4 (0.3)	3.1 (1.3)	2.9 (0.4)
SDP	0.0 (0.0)	6.2 (1.6)	65.5 (1.9)

in Table 1. The AV method provides the best results (a test error of 3.1%), using significantly less support vectors than the other methods. The SDP method improves only the results of the Gaussian and the polynomial kernel.

**2.2.1 Parameter selection.** Techniques for the combination of kernels can be successfully applied to the problem of parameter selection in kernel methods. This links with the comment of Bousquet and Schölkopf about the need for further research on satisfactory alternatives, other than cross-validation, to choose the parameters in kernel methods. We illustrate this situation using a collection of Gaussian kernels on the cancer data set. Let  $\{K_1, \dots, K_{12}\}$  be a set of Gaussian kernels  $K_c(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/c)$  with parameters  $c = 0.1, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90$  and 100, respectively. This wide set covers a realistic range of possible values for the kernel parameter. The test errors for 12 SVMs using these Gaussian kernels range from 3.1% to 24.7%. In this case, the AV method, combining the 12 Gaussian kernels, gives the best result obtained using only one of the Gaussian kernels under consideration (with a test error of 3.1%). It is important to note that the performance of the AV method (which is parameter-free) is not affected by the inclusion of kernels with a bad generalization performance. Since, in general, the best parameter choice is not known in advance, the methodology just described provides an alternative that minimizes the effect of bad parameter selection.

### 3. THE BIAS-VARIANCE PROBLEM

Regarding the comments on statistical consistency provided by Bartlett, Jordan and McAuliffe, also pointed out by Bousquet and Schölkopf, we agree that the Vapnik-Chervonenkis (VC) dimension is not central in the analysis of SVMs. In fact, in [6], the bias-variance problem is analyzed in the context of regularization for the quadratic loss function (the analysis by

Steinwart cited by the discussants is for the  $L_1$ -SVM, as Bousquet and Schölkopf remark). Cucker and Smale [6] replaced the VC dimension by the radius  $r$  of a ball in a RKHS space ( $r$  is the norm in the RKHS of the minimizer of the empirical risk). Since the regularization parameter  $\lambda$  (using the notation of Bartlett, Jordan and McAuliffe) is inversely proportional to  $r$ , large values of  $\lambda$  correspond to large bias, while small values of  $\lambda$  lead to large variance. The Cucker and Smale paper [6] also contains a theorem (Corollary 2) that is in agreement with the discussants' comment about the fact that the regularization coefficient must decrease with the sample size. In addition, statistical consistency can be derived from the results in the paper if a rich enough kernel is used (i.e., a universal kernel, in the sense of Steinwart).

#### 4. DIFFERENTIAL GEOMETRY METHODS AND KERNEL METHODS

In her comment, Wahba introduces a particular method for learning the kernel data matrix for the purpose of manifold unfolding. As Wahba and her co-authors remark in [10], the manifold unfolding problem is closely related to the construction of a kernel. In fact, Ham, Lee, Mika and Schölkopf [7] showed that several of the proposed techniques for manifold learning, namely ISOMAP [19], graph Laplacian eigenmap [2] and locally linear embedding [15], can be interpreted as particular cases of kernel principal component analysis [17]. As Ham and co-authors point out, these techniques can be viewed as a “warping of the input space into a feature space where the manifold is flat.”

In this regard, Burges [4] described the intrinsic geometry of the manifold which arises for a particular choice of the kernel. In particular, he shows that the Riemannian metric induced on the manifold by its embedding can be expressed, in terms of the kernel, in closed form. A closely related approach can be found in [20]. It is worth mentioning that the implicit geometric assumption within manifold unfolding is that the decision surface (for the case of classification) is smooth with respect to the underlying geometry [3].

Finally, we would like to thank David Rios and Francisco J. Prieto for their careful reading of the manuscript and suggestions. We hope that our paper and this interesting discussion encourage the statistical community to pursue further research on support vector machines and other related methodologies.

#### REFERENCES

- [1] BAZARAA, M. S., JARVIS, J. J. and SHERALI, H. D. (1990). *Linear Programming and Network Flows*, 2nd ed. Wiley, New York. [MR1029024](#)
- [2] BELKIN, M. and NIYOGI, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* **15** 1373–1396.
- [3] BELKIN, M. and NIYOGI, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning* **56** 209–239.
- [4] BURGES, C. J. C. (1999). Geometry and invariance in kernel based methods. *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 89–116. MIT Press, Cambridge, MA.
- [5] CUCKER, F. and SMALE, S. (2002). On the mathematical foundations of learning. *Bull. Amer. Math. Soc. (N.S.)* **39** 1–49. [MR1864085](#)
- [6] CUCKER, F. and SMALE, S. (2002). Best choices for regularization parameters in learning theory: On the bias-variance problem. *Found. Comput. Math.* **2** 413–428. [MR1930945](#)
- [7] HAM, J., LEE, D. D., MIKA, S. and SCHÖLKOPF, B. (2004). A kernel view of the dimensionality reduction of manifolds. In *Proc. Twenty-First International Conference on Machine Learning (ICML-04)* (R. Greiner and D. Schuurmans, eds.) 369–376. ACM Press, New York.
- [8] KEERTHI, S. S. and LIN, C.-J. (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation* **15** 1667–1689.
- [9] LANCKRIET, G. R. G., CRISTIANINI, N., BARTLETT, P., EL GHAOU, L. and JORDAN, M. I. (2004). Learning the kernel matrix with semi-definite programming. *J. Mach. Learn. Res.* **5** 27–72.
- [10] LU, F., LIN, Y. and WAHBA, G. (2005). Robust manifold unfolding with kernel regularization. Technical Report 1108, Dept. Statistics, Univ. Wisconsin.
- [11] MANGASARIAN, O. L. and WOLBERG, W. H. (1990). Cancer diagnosis via linear programming. *SIAM News* **23** 1, 18.
- [12] MARTIN, I., MOGUERZA, J. M. and MUÑOZ, A. (2004). Combining kernel information for support vector classification. *Multiple Classifier Systems. Lecture Notes in Comput. Sci.* **3077** 102–111. Springer, Berlin.
- [13] MUÑOZ, A. and MOGUERZA, J. M. (2006). Estimation of high-density regions using one-class neighbor machines. *IEEE Trans. Pattern Analysis and Machine Intelligence* **28** 476–480.
- [14] NEWTON, H. J. (2002). A conversation with Emanuel Parzen. *Statist. Sci.* **17** 357–378. [MR1962489](#)
- [15] ROWEIS, S. and SAUL, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science* **290** 2323–2326.
- [16] SCHÖLKOPF, B., PLATT, J. C., SHAWE-TAYLOR, J., SMOLA, A. J. and WILLIAMSON, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation* **13** 1443–1471.
- [17] SCHÖLKOPF, B., SMOLA, A. J. and MÜLLER, K.-R. (1999). Kernel principal component analysis. In *Advances in Kernel Methods—Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.) 327–352. MIT Press, Cambridge, MA.

- [18] SOLLICH, P. (2002). Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning* **46** 21–52.
- [19] TENENBAUM, J. B., DE SILVA, V. and LANGFORD, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science* **290** 2319–2323.
- [20] WU, S. and AMARI, S.-I. (2002). Conformal transformation of kernel functions: A data-dependent way to improve support vector machine classifiers. *Neural Processing Letters* **15** 59–67.