DEPARTMENT OF STATISTICS
University of Wisconsin
1210 West Dayton St.
Madison, WI 53706

TECHNICAL REPORT NO. 1045
October 24, 2001

Optimal Properties and Adaptive Tuning of Standard and Nonstandard Support Vector Machines

Grace Wahba, Yi Lin, Yoonkyung Lee and Hao Zhang

wahba, yilin, yklee, hzhang@stat.wisc.edu

http://www.stat.wisc.edu/~wahba, ~yilin, ~yklee, ~hzhang

This paper was prepared for the Proceedings of the Mathematical Sciences Research Institute, Berkeley Workshop on Nonlinear Estimation and Classification. Research partly supported by NSF Grant DMS0072292 and NIH Grant EY09946. This TR superceeds TR 1039.

Please Continue.

1

Optimal Properties and Adaptive Tuning of Standard and Nonstandard Support Vector Machines

Grace Wahba, Yi Lin, Yoonkyung Lee and Hao Zhang

October 24, 2001

Abstract

We review some of the basic ideas of Support Vector Machines (SVM's) for classification, with the goal of describing how these ideas can sit comfortably inside the statistical literature in decision theory and penalized likelihood regression. We review recent work on adaptive tuning of SVMs, discussing generalizations to the nonstandard case where the training set is not representative and misclassification costs are not equal. Mention is made of recent results in the multicategory case.

1.1 Introduction

This paper is an expanded version of the the talk given by one of the authors (GW) at the Mathematical Sciences Research Institute Berkeley Workshop on Nonlinear Estimation and Classification, March 20, 2001. In this paper we review some of the basic ideas of Support Vector Machines(SVMs) with the goal of describing how these ideas can sit comfortably inside the statistical literature in decision theory and penalized likelihood regression, and we review some of our own related research.

Support Vector Machines (SVM's) burst upon the classification scene in the early 90's, and soon became the method of choice for many researchers and practitioners involved in supervised machine learning. The talk of Tommi Poggio at the Berkeley workshop highlights some of the many interesting applications. The website `http://kernel-machines.org` is a popular repository for papers, tutorials, software, and links related to SVM's. A recent search in `http://www.google.com` for 'Support Vector Machines' leads to 'about 10,600' listings. Recent books on the topic include [23] [24] [5], and there is a section on SVM's in [10]. [5] has an incredible (for a technical book) ranking in amazon.com as one of the 4500 most popular books.

The first author became interested in SVM's at the AMS-IMS-SIAM Joint Summer Research Conference on Adaptive Selection of Models and Statistical Procedures, held at Mount Holyoke College in South Hadley MA in June 1996. There, Vladimir Vapnik, generally credited with the invention of SVM's, gave an interesting talk, and during the discussion after his talk it became evident that the SVM could be derived as the solution to an optimization problem in a Reproducing Kernel Hilbert Space (RKHS), [25], [29] [13], [27], thus bearing a resemblance to penalized likelihood and other regularization methods used in nonparametric regression. This served to link the rapidly developing SVM literature in supervised machine learning to the now obviously related statistics literature. Considering the relatively recent development of SVM's, compared to the 40 or so year history of other classification methods, it is of interest to question theoretically why SVM's work so well. This question was recently answered in [18], where it was shown that, provided a rich enough RKHS is used, the SVM is implementing the Bayes rule for classification. Convergence rates in some special cases can be found [19]. An examination of the form of the SVM shows that it is doing the implementation in a flexible and particularly efficient manner.

As with other regularization methods, there is always one, and sometimes several tuning parameters which must be chosen well in order to have efficient classification in nontrivial cases. Our own work has focused on the extension of the Generalized Approximate Cross Validation (GACV) [35] [17] [8] from penalized likelihood estimates to SVM's, see [21] [20] [32] [29]. At the Berkeley meeting, Bin Yu pointed GW to the $\xi\alpha$ method of Joachims [12], which turned out to be closely related to the GACV. Code for the $\xi\alpha$ estimate is available in SVM^{light} `http://ais.gmd.de/ thorsten/svm_light/`. At about this time there was a lot of activity in the development of tuning methods, and a number of them [26] [11] [22] [12] [2] turned out to be related under various circumstances.

We first review optimal classification in the two-category classification problem. We describe the standard case, where the training set is representative of the general population, and the cost of misclassification is the same for both categories, and then turn to the nonstandard case, where neither of these assumptions hold. We then describe the penalized likelihood estimate for Bernoulli data, and compare it with the standard SVM. Next we discuss how the SVM implements the Bayes rule for classification and then we turn to the GACV for tuning the standard SVM. The GACV and Joachims' $\xi\alpha$ method are then compared. Next we turn to the nonstandard case. We describe the nonstandard SVM, and show how both the GACV and the $\xi\alpha$ method can be generalized in that case, from [31]. A modest simulation shows that they behave similarly. Finally, we briefly mention that we have generalized the (standard and nonstandard) SVM to the multicategory case [15].

1.2 Optimal Classification and Penalized Likelihood

Let $h_{\mathcal{A}}(\cdot), h_{\mathcal{B}}(\cdot)$ be densities of x for class $\mathcal{A}$ and class $\mathcal{B}$, and let $\pi_{\mathcal{A}}$ = probability the next observation (Y) is an $\mathcal{A}$, and let $\pi_{\mathcal{B}} = 1 - \pi_{\mathcal{A}}$ = probability that the next observation is a $\mathcal{B}$. Then $p(x) \equiv prob\{Y = \mathcal{A}|x\} = \frac{\pi_{\mathcal{A}} h_{\mathcal{A}}(x)}{\pi_{\mathcal{A}} h_{\mathcal{A}}(x) + \pi_{\mathcal{B}} h_{\mathcal{B}}(x)}$.

Let $C_{\mathcal{A}}$ = cost to falsely call a $\mathcal{B}$ an $\mathcal{A}$ and $C_{\mathcal{B}}$ = cost to falsely call an $\mathcal{A}$ a $\mathcal{B}$. A classifier ϕ is a map $\phi(x) : x \rightarrow \{\mathcal{A}, \mathcal{B}\}$. The optimal (Bayes) classifier, which minimizes the expected cost is

$$\phi_{\mathrm{OPT}}(x) = \begin{cases} \mathcal{A} & \text{if } \frac{p(x)}{1-p(x)} > \frac{C_{\mathcal{A}}}{C_{\mathcal{B}}}, \\ \mathcal{B} & \text{if } \frac{p(x)}{1-p(x)} < \frac{C_{\mathcal{A}}}{C_{\mathcal{B}}}. \end{cases} \tag{1.1}$$

To estimate $p(x)$, or, alternatively the logit $f(x) \equiv \log p(x)/(1-p(x))$, we use a training set $\{y_i, x_i\}_{i=1}^n$, $y_i \in \{\mathcal{A}, \mathcal{B}\}$, $x_i \in \mathcal{T}$, where $\mathcal{T}$ is some index set. At first we assume that the relative frequency of $\mathcal{A}$'s in the training set is the same as in the general population. f can be estimated (nonparametrically) in various ways. If $C_{\mathcal{A}}/C_{\mathcal{B}} = 1$, and f is the logit, the optimal classifier is

$$\begin{array}{l} f(x) > 0 \text{ (equivalently, } p(x) - \frac{1}{2} > 0) \rightarrow \mathcal{A} \\ f(x) < 0 \text{ (equivalently, } p(x) - \frac{1}{2} < 0) \rightarrow \mathcal{B} \end{array}$$

In the usual penalized log likelihood estimation of f, the observations are coded as

$$y = \begin{cases} 1 & \text{if } \mathcal{A}, \\ 0 & \text{if } \mathcal{B}. \end{cases} \tag{1.2}$$

The probability distribution function for $y \mid p$ is then

$$\mathcal{L} = p^y(1-p)^{1-y} = \begin{cases} p & \text{if } y = 1 \\ (1-p) & \text{if } y = 0 \end{cases}.$$

Using $p = e^f/(1+e^f)$ gives the negative log likelihood $-\log\mathcal{L} = -yf + \log(1+e^f)$. For comparison with the support vector machine we will describe a somewhat special case (General cases are in [13], [17], [8], [34]). The penalized log likelihood estimate of f is obtained as the solution to the problem: Find $f(x) = b + h(x)$ with $h \in \mathcal{H}_K$ to minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left[-y_i f(x_i) + \log(1+e^{f(x_i)})\right] + \lambda\|h\|^2_{\mathcal{H}_K} \tag{1.3}$$

where $\lambda > 0$, and $\mathcal{H}_K$ is the reproducing kernel Hilbert space (RKHS) with reproducing kernel

$$K(s,t), \quad s,t \in \mathcal{T}. \tag{1.4}$$

For more on RKHS, see [1] [28]. RKHS may be tailored to many applications since any symmetric positive definite function on $\mathcal{T} \times \mathcal{T}$ has a unique RKHS associated with it.

Theorem: [13] f_λ, the minimizer of (1.3) has a representation of the form

$$f_\lambda(x) = b + \sum_{i=1}^{n} c_i K(x, x_i). \tag{1.5}$$

It is a property of RKHS that

$$\|h\|^2_{\mathcal{H}_K} \equiv \sum_{i,j=1}^{n} c_i c_j K(x_i, x_j). \tag{1.6}$$

To obtain the estimate f_λ, (1.5) and (1.6) are substituted into (1.3), which is then minimized with respect to b and $c = (c_1, \dots, c_n)$. Given positive λ, this is a strictly convex optimization problem with some nice features special to penalized likelihood for exponential families, provided that p is not too near 0 or 1. The smoothing parameter λ, and certain other parameters which may be inside K may be chosen by Generalized Approximate Cross Validation (GACV) for Bernoulli data, see ([17]) and references cited there. The target for GACV is to minimize the Comparative Kullback-Liebler (CKL) distance of the estimate from the true distribution:

$$CKL(\lambda) = E_{true} \sum_{i=1}^{n} -y_{new.i} f_\lambda(x_i) + \log(1 + e^{f_\lambda(x_i)}), \tag{1.7}$$

where $y_{new.i}$ is a new observation with attribute vector x_i.

1.3 Support Vector Machines (SVM's)

For SVM's, the data is coded differently:

$$y = \begin{cases} +1 & \text{if } \mathcal{A}, \\ -1 & \text{if } \mathcal{B}. \end{cases} \tag{1.8}$$

The support vector optimization problem is: Find $f(x) = b + h(x)$ with $h \in \mathcal{H}_K$ to minimize

$$\frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \lambda \|h\|^2_{\mathcal{H}_K} \tag{1.9}$$

where $(\tau)_+ = \tau$, if $\tau > 0$, and 0 otherwise. The original support vector machine (see e. g. ([26]) was obtained from a different argument, but it is well known that it is equivalent to (1.9), see ([29], [25]). As before, the SVM f_λ has the representation (1.5). To obtain the classifier f_λ for a fixed $\lambda > 0$, (1.5) and (1.6) are substituted into (1.9) resulting in a mathematical programming problem to be solved numerically. The classifier is then $f_\lambda(x) > 0 \rightarrow \mathcal{A}$, $f_\lambda(x) < 0 \rightarrow \mathcal{B}$.

We may compare the penalized log likelihood estimate of the logit $\log p/(1-p)$ and the SVM (the minimizer of (1.9)) by coding y in the likelihood as

$$\tilde{y} = \begin{cases} +1 & \text{if } \mathcal{A}, \\ -1 & \text{if } \mathcal{B}. \end{cases}$$

Then $-yf + \log(1 + e^f)$ becomes $\log(1 + e^{-\tilde{y}f})$, where f is the logit. Figure 1.1

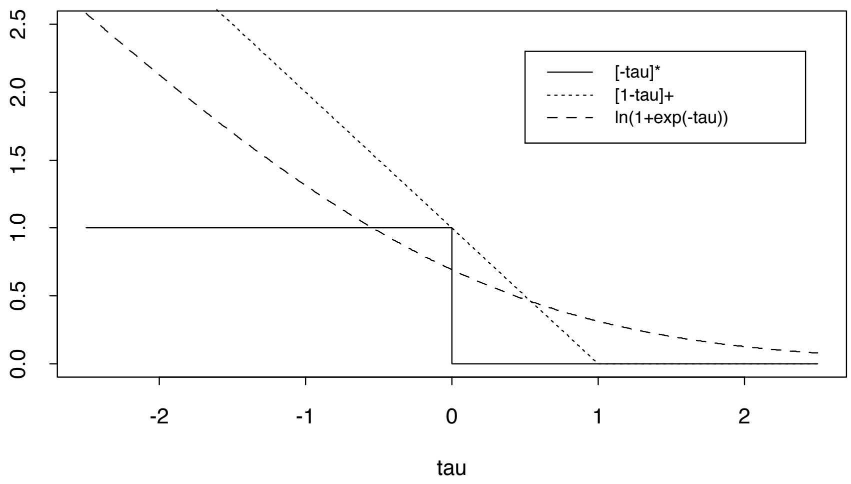


Figure 1.1. Adapted from [29]. Comparison of $[-\tau]_*$, $(1-\tau)_+$ and $log_e(1+e^{-\tau})$.

compares $\log(1+e^{-yf})$, $(1-yf)_+$ and $[-yf]_*$ as functions of $\tau = yf$ where

$$[\tau]_* = \begin{cases} 1 & \text{if } \tau \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $[-yf]_*$ is 1 or 0 according as y and f have the same sign or not. Calling $[-yf]_*$ the misclassification counter, one might consider minimizing the misclassification count plus some (quadratic) penalty functional on f but this is a nonconvex problem and difficult to minimize numerically. Numerous authors have replaced the misclassification counter by some convex upper bound to it. The support vector, or ramp function $(1-yf)_+$ is a convex upper bound to the misclassification counter, and Bin Yu observed that $log_2(1+e^{-\tau})$ is also a convex upper bound. Of course it is also possible to use a penalized likelihood estimate for classification see [33]. However, the ramp function (modulo the slope) is the 'closest' convex upper bound to the misclassification counter, which provides one heuristic argument why SVM's work so well in the classification problem.

Recall that the penalized log likelihood estimate was tuned by a criteria which chose λ to minimize a proxy for the CKL of (1.7) conditional on the same x_i. By analogy, for the SVM classifier we were motivated in [20] [21] [29] [32] to say that it is optimally tuned if λ minimizes a proxy for the Generalized Comparative

Kullback-Liebler distance (GCKL), defined as

$$GCKL(\lambda) = E_{true} \frac{1}{n} \sum_{i=1}^{n} (1 - y_{new \cdot i} f_{\lambda}(x_i))_+ . \tag{1.10}$$

That is, λ (and possibly other parameters in K) are chosen to minimize a proxy for an upper bound on the misclassification rate.

1.4 Why is the SVM so successful?

There is actually an important result which explains why the SVM is so successful: We have the Theorem:

Theorem [18]: The minimizer over f of $E_{true}(1 - y_{new} f(x))_+$ is sign $(p(x) - \frac{1}{2})$, which coincides with the sign of the logit.

As a consequence, if $\mathcal{H}_K$ is a sufficiently rich space, the minimizer of (1.9) where λ is chosen to minimize (a proxy for) $GCKL(\lambda)$, is estimating the sign of the logit. This is exactly what you need to implement the Bayes classifier! $E_{true}(1 - y_{new} f_{\lambda})_+$ is given by

$$E_{true}(1 - y_{new} f_{\lambda})_+ = \left\{ \begin{array}{lrl} p(1 - f_{\lambda}), & f_{\lambda} & < -1 \\ p(1 - f_{\lambda}) + (1 - p)(1 + f_{\lambda}), & -1 < \; f_{\lambda} & < +1 \\ (1 - p)(1 + f_{\lambda}), & f_{\lambda} & > +1. \end{array} \right\} \tag{1.11}$$

Since the true p is only known in a simulation experiment, $GCKL$ is also only known in experiments. The experiment to follow, which is reprinted from [18], demonstrates this theorem graphically. Figure 1.2 gives the underlying conditional probability function $p(x) = Prob\{y = 1|x\}$ used in the simulation. The function sign $(p(x) - 1/2)$ is 1, for $0.25 < x < 0.75$; -1 otherwise. A training set sample of $n = 257$ observations were generated with the x_i equally spaced on $[0, 1]$, and p according to Figure 1.2. The SVM was computed and f is given in Figure 1.3 for $n\lambda = 2^{-1}, 2^{-2}, \ldots, 2^{-25}$, in the plots left to right starting with the top row and moving down. We see that solution f is close to sign $(p(x) - 1/2)$ when $n\lambda$ is in the neighborhood of 2^{-18}. 2^{-18} was the minimizer of the $GCKL$, suggesting that it is necessary to tune the SVM to estimate sign $(p(x) - 1/2)$ well.

1.5 The GACV for choosing λ (and other parameters in K)

In [29], [32], [20], [21] we developed and tested the GACV for tuning SVM's. In [29] a randomized version of GACV was obtained using a heuristic argument

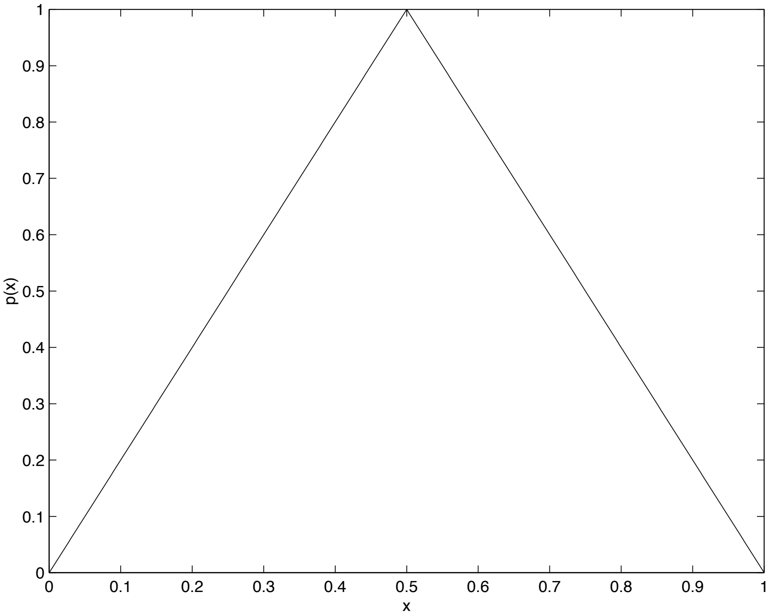


Figure 1.2. From [18]. The underlying conditional probability function $p(x) = Prob\{y = 1|x\}$ in the simulation.

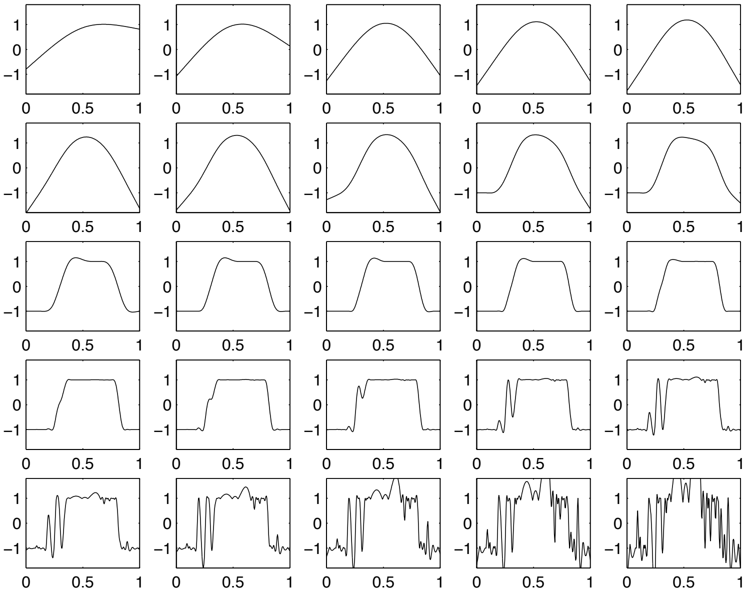


Figure 1.3. From [18]. Solutions to the SVM regularization for $n\lambda = 2^{-1}, 2^{-2}, \ldots, 2^{-25}$, left to right starting with the top row.

related to the derivation of the GCV [4], [9] for Gaussian observations and for the GACV for Bernoulli observations [35]. In [32], [20], [21] it was seen that a direct (non-randomized) version was readily available, easy to compute, and worked well. At about same time, there were several other tuning results [3] [11] [12] [22] [26] which are closely related to each other and to the GACV in one way or another. We will discuss these later. The arguments below follow [32]. The goal here is to obtain a proxy for the (unobservable) $GCKL(\lambda)$ of (1.10). Let $f_\lambda^{[-k]}$ be the minimizer of the form $f = b + h$ with $h \in \mathcal{H}_K$ to minimize

$$\frac{1}{n} \sum_{\substack{i = 1 \\ i \neq k}} (1 - y_i f(x_i))_+ + \lambda \|h\|_K^2.$$

Let

$$V_0(\lambda) = \frac{1}{n}\sum_{k=1}^{n}(1 - y_k f_\lambda^{[-k]}(x_k))_+.$$

We write

$$V_0(\lambda) \equiv \text{OBS}(\lambda) + D(\lambda), \tag{1.12}$$

where

$$\text{OBS}(\lambda) = \frac{1}{n}\sum_{k=1}^{n}(1 - y_k f_\lambda(x_k))_+. \tag{1.13}$$

and

$$D(\lambda) = \frac{1}{n}\sum_{k=1}^{n}[(1 - y_k f_\lambda^{[-k]}(x_k))_+ - (1 - y_k f_\lambda(x_k))_+] \tag{1.14}$$

Using a rather crude argument, [32] showed that $D(\lambda) \approx \hat{D}(\lambda)$ where

$$\hat{D}(\lambda) = \frac{1}{n}\left[\sum_{y_i f_\lambda(x_i) < -1} 2\frac{\partial f_\lambda(x_i)}{\partial y_i} + \sum_{y_i f_\lambda(x_i) \in [-1,1]} \frac{\partial f_\lambda(x_i)}{\partial y_i}\right]. \tag{1.15}$$

In this argument, y_i is treated as though it is a continuous variate, and the lack of differentiability is ignored. Then

$$V_0(\lambda) \approx \text{OBS}(\lambda) + \hat{D}(\lambda). \tag{1.16}$$

$\hat{D}(\lambda)$ may be compared to trace $A(\lambda)$ in GCV and unbiased risk estimates.

How shall we interpret $\frac{\partial f_\lambda(x_i)}{\partial y_i}$? Let $K_{n\times n} = \{K(x_i, x_j)\}, D_y = \begin{pmatrix} y_1 & & \\ & \ddots & \\ & & y_n \end{pmatrix}, \begin{pmatrix} f_\lambda(x_1) \\ \vdots \\ f_\lambda(x_n) \end{pmatrix} = Kc + eb \quad , \quad e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$. We will examine the optimization problem for (1.9): Find (b, c) to minimize $\frac{1}{n}\sum_{i=1}^{n}(1 -$

$y_i f_\lambda(x_i))_+ + \lambda c'Kc$. The dual problem for (1.9) is known to be: Find $\alpha = \begin{pmatrix}\alpha_1\\ \vdots \\ \alpha_n\end{pmatrix}$ to minimize $\frac{1}{2}\alpha'\left(\frac{1}{2n\lambda}D_yKD_y\right)\alpha - e'\alpha$ subject to $\begin{pmatrix}0\\ \vdots \\ 0\end{pmatrix} \leq \begin{pmatrix}\alpha_1\\ \vdots \\ \alpha_n\end{pmatrix} \leq \begin{pmatrix}1\\ \vdots \\ 1\end{pmatrix}$ and $y'\alpha = 0$, where $y = \begin{pmatrix}y_1\\ \vdots \\ y_n\end{pmatrix}$, and $c = \frac{1}{2n\lambda}D_y\alpha$. Then $\begin{pmatrix}f_\lambda(x_1)\\ \vdots \\ f_\lambda(x_n)\end{pmatrix} = \frac{1}{2n\lambda}KD_y\alpha + eb$, and we interpret $\frac{\partial f_\lambda(x_i)}{\partial y_i}$ as $\frac{\partial f_\lambda(x_i)}{\partial y_i} = \frac{1}{2n\lambda}K(x_i,x_i)\alpha_i$, resulting in

$$\hat{D}(\lambda) = \frac{1}{n}\left[2\sum_{y_i f_\lambda(x_i)<-1}\frac{\alpha_i}{2n\lambda}K(x_i,x_i) + \sum_{y_i f_\lambda(x_i)\in[-1,1]}\frac{\alpha_i}{2n\lambda}K(x_i,x_i)\right] \tag{1.17}$$

and

$$GACV(\lambda) = OBS(\lambda) + \hat{D}(\lambda). \tag{1.18}$$

Let $\theta_k = \frac{\alpha_k}{2n\lambda}K(x_k,x_k)$, and note that if $y_k f_\lambda(x_k) > 1$, then $\alpha_k = 0$. If $\alpha_k = 0$, leaving out the kth data point does not change the solution. Otherwise, the expression for $\hat{D}(\lambda)$ in (1.17) is equivalent in a leaving-out-one argument, to approximating $[y_k f_\lambda(x_k) - y_k f_\lambda^{[-k]}(x_k)]$ by θ_k if $y_k f_\lambda(x_k) \in [-1,1]$ and by $2\theta_k$ if $y_k f_\lambda(x_k) < -1$. Jaakkola and Haussler, [11] in the special case that b is taken as 0 proved that θ_k is an upper bound for $[y_k f_\lambda(x_k) - y_k f_\lambda^{[-k]}(x_k)]$ and Joachims [12] proved in the case considered here, that $[y_k f_\lambda(x_k) - y_k f_\lambda^{[-k]}(x_k)] \leq 2\theta_k$. Vapnik [26] in the case that b is set equal to 0, and $OBS = 0$, proposed choosing the parameters to minimize the so-called radius-margin bound. This works out to minimizing $\sum_i \theta_i$ when $K(x_i,x_i)$ is the same for all i. Chapelle and Vapnik [2] and Opper and Winther [22] have related proposals for choosing the tuning parameters. More details on some of these comparisons may be found in [3].

1.6 Comparing GACV and Joachims' $\xi\alpha$ method for choosing tuning parameters.

Let $\xi_i = (1 - y_i f_{\lambda i})_+$, and $K_{ij} = K(x_i,x_j)$. The GACV is then

$$GACV(\lambda) = \frac{1}{n}\left[\sum_{i=1}^{n}\xi_i + 2\sum_{y_i f_{\lambda i}<-1}\frac{\alpha_i}{2n\lambda}K_{ii} + \sum_{y_i f_{\lambda i}\in[-1,1]}\frac{\alpha_i}{2n\lambda}K_{ii}\right]. \tag{1.19}$$

A more direct target than $GCKL(\lambda)$ is the misclassification rate, defined (conditional on the observed set of attribute variables) as

$$MISCLASS(\lambda) = E_{true}\frac{1}{n}\sum_{i=1}^{n}[-y_i f_{\lambda i}]_* \equiv \frac{1}{n}\sum_{i=1}^{n}\{p_i[-f_{\lambda i}]_* + (1-p_i)[f_{\lambda i}]_*\}. \tag{1.20}$$

Joachims [12], Equation (7) proposed the $\xi\alpha$ (to be called XA here) proxy for MISCLASS as:

$$XA(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left[\xi_i + \rho\frac{\alpha_i}{2n\lambda}K - 1\right]_* \tag{1.21}$$

where $\rho = 2$ and here (with some abuse of notation) K is an upper bound on $K_{ii} - K_{ij}$. Letting $\theta_i = \rho\frac{\alpha_i}{2n\lambda}K$, it can be shown that the sum in $XA(\lambda)$ counts all of the samples for which $y_i f_{\lambda i} \le \theta_i$. Since $y_i f_{\lambda i} > 1 \Rightarrow \alpha_i = 0$, XA may also be written

$$XA(\lambda) = \frac{1}{n}\left[\sum_{i=1}^{n}[-y_i f_{\lambda i}]_* + \sum_{y_i f_{\lambda i}\le 1} I_{[\frac{\rho\alpha_i}{2n\lambda}K]}(y_i f_{\lambda i})\right], \tag{1.22}$$

where $I_{[\theta]}(\tau) = 1$ if $\tau \in (0,\theta]$ and 0 otherwise. Equivalently the sum in XA counts the misclassified cases in the training set plus all of the cases where $y_i f_{\lambda i} \in (0, \rho\frac{\alpha_i}{2n\lambda}K]$ (adopting the convention that if $f_{\lambda i}$ is exactly 0 then the example is considered misclassified). In some of his experiments Joachims (empirically) set $\rho = 1$ because it achieved a better estimate of the misclassification rate than did the XA with $\rho = 2$. Let us go over how estimates of the difference between a target and its leaving out one version may be used to construct estimates when the 'fit' is not the same as the target - here the 'fit' is $(1 - y_i f_{\lambda i})_+$, while the 'target' for the XA is $[-y_i f_{\lambda i}]_*$. We will use the argument in the next section to generalize the XA to the nonstandard case in the same way that the GACV is generalized to its nonstandard version.

Let $f_{\lambda i}^{[-i]} = f_{\lambda}^{[-i]}(x_i)$. Suppose we have the approximation $y_i f_{\lambda i} \approx y_i f_{\lambda i}^{[-i]} + \theta_i$, with $\theta_i \ge 0$. A leaving out one estimate of the misclassification rate is given by $V_0(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f_{\lambda i}^{[-i]}]_*$. Now $V_0(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f_{\lambda i}]_* + D(\lambda)$ where here

$$D(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\{[-y_i f_{\lambda i}^{[-i]}]_* - [-y_i f_{\lambda i}]_*\}. \tag{1.23}$$

Now, the ith term in $D(\lambda) = 0$ unless $y_i f_{\lambda i}^{[-i]}$ and $y_i f_{\lambda i}$ have different signs. For $\theta_i > 0$ this can only happen if $y_i f_{\lambda i} \in (0, \theta_i]$. Assuming the approximation

$$y_i f_{\lambda i} \approx y_i f_{\lambda i}^{[-i]} + \frac{\alpha_i}{2n\lambda}K_{ii} \tag{1.24}$$

tells us that $\frac{1}{n}\sum_{y_i f_{\lambda i}\le 1} I_{[\frac{\alpha_i}{2n\lambda}K_{ii}]}(y_i f_{\lambda i})$, can be taken as an approximation to $D(\lambda)$ of (1.23), resulting in (1.22). This provides an alternate derivation as well as an alternative interpretation of XA with $\rho = 1, K$ replaced by K_{ii}.

1.7 The Nonstandard SVM and the Nonstandard GACV

We now review the nonstandard case, from [21]. Let $\pi^s_{\mathcal{A}}$ and $\pi^s_{\mathcal{B}}$ be the relative frequencies of the $\mathcal{A}$ and $\mathcal{B}$ classes in the training (sample) set. Recall that $\pi_{\mathcal{A}}$ and $\pi_{\mathcal{B}}$ are the relative frequencies of the two classes in the target population, $C_{\mathcal{A}}$ and $C_{\mathcal{B}}$ are the costs of falsely calling a $\mathcal{B}$ an $\mathcal{A}$ and falsely calling an $\mathcal{A}$ a $\mathcal{B}$ respectively, and $h_{\mathcal{A}}(x)$ and $h_{\mathcal{B}}(x)$ are the the densities of x in the $\mathcal{A}$ and $\mathcal{B}$ classes, and that the probability that a subject from the target population with attribute x belongs to the $\mathcal{A}$ class is $p(x) = \frac{\pi_{\mathcal{A}} h_{\mathcal{A}}(x)}{\pi_{\mathcal{A}} h_{\mathcal{A}}(x) + \pi_{\mathcal{B}} h_{\mathcal{B}}(x)}$. However, the probability that a subject with attribute x chosen from a population with the same distribution as the training set, belongs to the $\mathcal{A}$ class, is $p_s(x) = \frac{\pi^s_{\mathcal{A}} h_{\mathcal{A}}(x)}{\pi^s_{\mathcal{A}} h_{\mathcal{A}}(x) + \pi^s_{\mathcal{B}} h_{\mathcal{B}}(x)}$. Letting $\phi(x)$ be the decision rule coded as a map from $x \in \mathcal{X}$ to $\{-1, 1\}$, where $1 \equiv \mathcal{A}$ and $-1 \equiv \mathcal{B}$, the expected cost, using $\phi(x)$ is $E_{x_{true}}\{C_{\mathcal{B}} p(x)[-\phi(x)]_* + C_{\mathcal{A}}(1 - p(x))[\phi(x)]_*\}$, where the expectation is taken over the distribution of x in the target population. The Bayes rule, which minimizes the expected cost is (from (1.1)) $\phi(x) = +1$ if $\frac{p(x)}{1-p(x)} > \frac{C_{\mathcal{A}}}{C_{\mathcal{B}}}$ and -1 otherwise. Since we don't observe a sample from the true distribution but only from the sampling distribution, we need to express the Bayes rule in terms of the sampling distribution p_s. It is shown in [21] that the Bayes rule can be written in terms of p_s as $\phi(x) = +1$ if $\frac{p_s(x)}{1-p_s(x)} > \frac{C_{\mathcal{A}}}{C_{\mathcal{B}}} \frac{\pi^s_{\mathcal{A}}}{\pi^s_{\mathcal{B}}} \frac{\pi_{\mathcal{B}}}{\pi_{\mathcal{A}}}$ and -1 otherwise. Let $L(-1) = C_{\mathcal{A}} \pi_{\mathcal{B}} / \pi^s_{\mathcal{B}}$ and $L(1) = C_{\mathcal{B}} \pi_{\mathcal{A}} / \pi^s_{\mathcal{A}}$. Then the Bayes rule can be expressed as $\phi(x) = sign\left[p_s(x) - \frac{L(-1)}{L(-1)+L(1)}\right]$. [21] proposed the nonstandard SVM to handle this nonstandard case as:

$$\min \frac{1}{n} \sum_{i=1}^{n} L(y_i)[(1 - y_i f(x_i))_+] + \lambda \|h\|^2_{H_K} \tag{1.25}$$

over all the functions of the form $f(x) = b + h(x)$, with $h \in H_K$. This definition is justified there by showing that, if the RKHS is rich enough and λ is chosen suitably, the minimizer of (1.25) tends to sign $\left[p_s(x) - \frac{L(-1)}{L(-1)+L(1)}\right]$. In [7] and references cited there, the authors considered the nonstandard case and proposed a heuristic solution, which is different than the one discussed here.

The minimizer of (1.25) has same form as in (1.5). [20] show that the dual problem becomes minimize $\frac{1}{2}\alpha' \left(\frac{1}{2n\lambda} D_y K D_y\right)\alpha - e'\alpha$ subject to $0 \leq \alpha_i \leq L(y_i), \;\; i = 1, 2, ..., n, \;\;$ and $\;\; y'\alpha = 0$, and $c = \frac{1}{2n\lambda} D_y \alpha$. The GACV for nonstandard problems was proposed there, in an argument generalizing the standard case, as:

$$GACV(\lambda) = \frac{1}{n}\left[\sum_{i=1}^{n} L(y_i)\xi_i + 2\sum_{y_i f_{\lambda i} < -1} L(y_i)\frac{\alpha_i}{2n\lambda} K_{ii} + \sum_{y_i f_{\lambda i} \in [-1,1]} L(y_i)\frac{\alpha_i}{2n\lambda} K_{ii}\right]. \tag{1.26}$$

It was shown to be a proxy for the nonstandard GCKL given by the nonstandard version of GCKL of (1.10), which can be written as:

$$GCKL(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\{L(1)p_s(x_i)(1-f_{\lambda i})_+ + L(-1)(1-p_s(x_i))(1+f_{\lambda i})_+\}. \tag{1.27}$$

(Compare (1.11).) We now propose a generalization, BRXA, of the XA as a computable proxy for the Bayes risk in the nonstandard case. Putting together the arguments which resulted in the the GACV of (1.19), the XA in the form that it appears in (1.22) and the nonstandard GACV of (1.26), we obtain the BRXA:

$$BRXA(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left[L(y_i)[-y_i f_{\lambda i}]_* + \sum_{y_i f_{\lambda i}\le 1} L(y_i) I_{[\frac{\alpha_i}{2n\lambda}K_{ii}]}(y_i f_{\lambda i})\right]. \tag{1.28}$$

The BRXA is a proxy for BRMISCLASS, given by

$$BRMISCLASS(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\{L(1)p_s(x_i)[-f_{\lambda i}]_* + L(-1)(1-p_s(x_i))[f_{\lambda i}]_*\}. \tag{1.29}$$

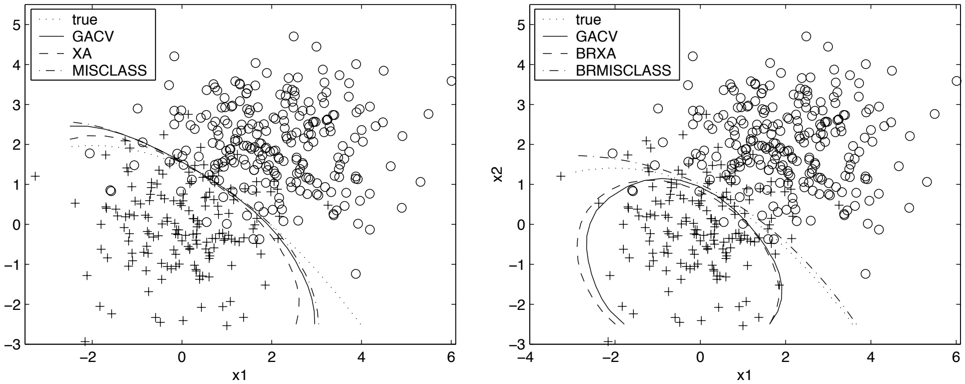


Figure 1.4. Observations, and true, GACV, XA and MISCLASS Decision Curves for the Standard Case (Left) and true, GACV, BRXA and BRMISCLASS Decision Curves for the Nonstandard Case (Right).

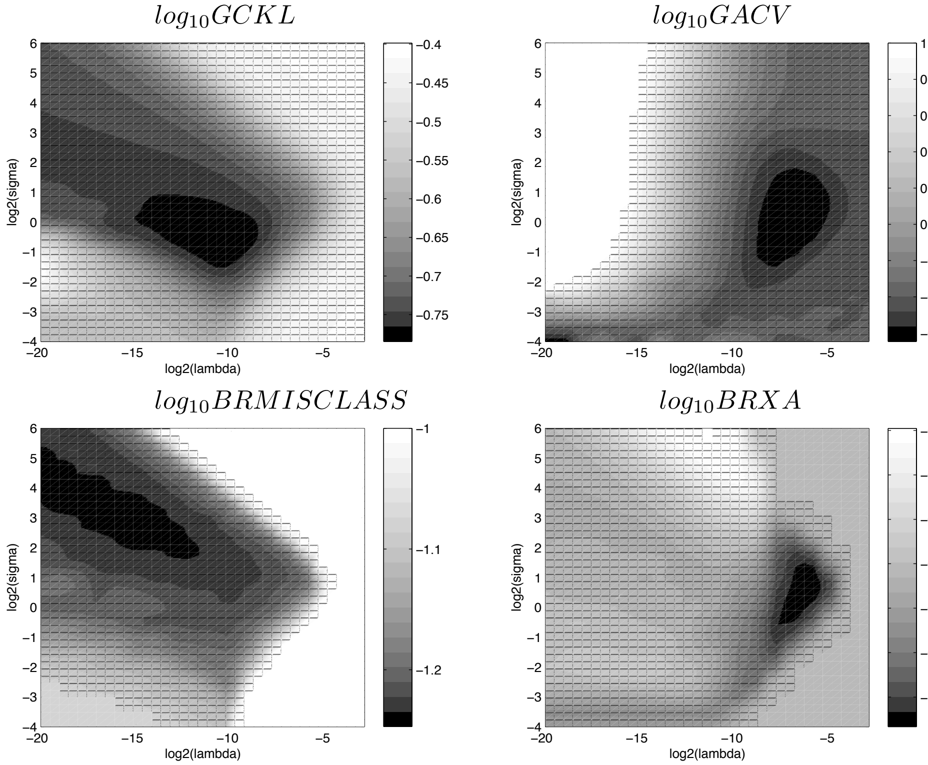
1.8 Simulation Results and Conclusions

The two panels of Figure 1.4 show the same simulated training set. The sample proportions of the $\mathcal{A}$ (+) and $\mathcal{B}$ (o) classes are .4 and .6 respectively. The conditional distribution of x given that the sample is from the $\mathcal{A}$ class is bivariate Normal with mean (0,0) and covariance matrix diag (1,1). The distribution for x from the $\mathcal{B}$ class is bivariate Normal with mean (2,2) and covariance diag (2,1). The left panel in Figure 1.4 is for the standard case, assuming that misclassification costs are the same for both kinds of misclassification, and the target population has the same proportions of the $\mathcal{A}$ and $\mathcal{B}$ as the sample. For the right panel, we assume that the costs of the two types of errors are different, and that the target population has different relative frequencies than the training set. We took $C_{\mathcal{A}} = 1$ $C_{\mathcal{B}} = 2, \pi_{\mathcal{A}} = 0.1, \pi_{\mathcal{B}} = 0.9$. As before, $\pi^s_{\mathcal{A}} = 0.4$, and $\pi^s_{\mathcal{B}} = 0.6$, yielding $L(-1) = C_{\mathcal{A}}\pi_{\mathcal{B}}/\pi^s_{\mathcal{B}} = 1.5$, and $L(1) = C_{\mathcal{B}}\pi_{\mathcal{A}}/\pi^s_{\mathcal{A}} = 0.5$. Since the distributions generating the data and the distributions of the target populations are known and involve Gaussians, the theoretical best decision rules (for an infinite future population) are known, and are given by the curves marked 'true' in both panels.

The Gaussian kernel $K(x, x') = \exp\{-\|x - x'\|^2/2\sigma^2\}$ was used, where $x = (x_1, x_2)$, and σ is to be tuned along with λ. The curves selected by the GACV of (1.19) and the XA of (1.22) in the standard case are shown in the left panel, along with MISCLASS of (1.20), which is only known in a simulation experiment. The right panel gives the curves chosen by the nonstandard GACV of (1.26), the BRXA of (1.28) and the BRMISCLASS of (1.29). The optimal (λ, σ) pair in each case for the tuned curves was chosen by a global search. It can be seen from both panels in Figure 1.4 that the MISCLASS curve, which is based on the (finite) observed sample is quite close to the theoretical true curve (based on an infinite future population), we make this observation because it will be easier to compare the GACV and the XA against MISCLASS than against the true, similarly for the BRMISCLASS curve. In both panels it can be seen that the decision curves determined by the GACV and the XA(BRXA) are very close.

We have computed the inefficiency of these estimates with respect to MISCLASS(BRMISCLASS), by inefficiency is meant the ratio of MISCLASS(BRMISCLASS) at the estimated (λ, σ) pair to its minimum value, a value of 1 means that the estimated pair is as accurate as possible, with respect to the (uncomputable) minimizer of MISCLASS(BRMISCLASS). The results for the standard case were: $GACV$: 1.0064, XA : $1.0062 - 1.0094$ (due to multiple neighboring minima in the grid search, the 1.0062 case is in Figure 1.4); and for the nonstandard case: $GACV : 1.151, BRXA : 1.166$.

Figure 1.5 gives contour plots for GCKL, GACV, BRMISCLASS and BRXA as a function of λ and σ in the nonstandard case. It can be seen that the GACV and BRXA curves have nearly the same minima. The GCKL and BRMISCLASS curves both have long, shallow, tilted cigar-shaped minima, and the GACV and BRXA minima are near the lower right end. For the standard case (not shown) the minima are somewhat more pronounced and the GACV and XA minima are closer to the MISCLASS minimum, and this is reflected in inefficiencies nearer to 1. (BR)MISCLASS curves in other simulation studies we have done show this same behavior. We have observed (as did Joachims) that the value of XA in the standard case is a good estimate of the value of MISCLASS at its minimizer, only slightly pessimistic. The GACV at its minimizer is an estimate of twice the misclassification rate. The value of one half the GACV is somewhat more pessimistic. We note that once one obtains the solution to the problem the computation of both GACV and (BR)XA are equally trivial.



$log_{10}GCKL$

$log_{10}GACV$

$log_{10}BRMISCLASS$

$log_{10}BRXA$

Figure 1.5. GCKL, GACV, BRMISCLASS, BRXA as functions of λ and σ^2, for the nonstandard example. Note different logarithmic scales in λ and σ.

The GACV in (quadratically) penalized likelihood cases generally scatters about the minimizer of its target (analogous to GCKL)(see [30]) but here, both the GACV and the BRXA (along with the standard case) appear to be biased towards larger λ. The (BR)MISCLASS surfaces are so flat in λ in our examples this does not seem to be a serious problem (less so in the standard case).

Recently we have obtained a generalization of the SVM to the k category case, which solves a single optimization problem to obtain a vector $f_\lambda(x) = (f_{1\lambda}(x), \ldots, f_{k\lambda}(x))$ where the category classifier is the component of f that is largest, see [14]. Usual muticategory classification schemes do one-vs-many or $\binom{k}{2}$ pairwise comparisons, and the multicategory SVM has advantages in certain examples. The GACV has been been extended to the nonstandard multicategory SVM case and it appears that the BRXA can also be extended. Penalized likelihood estimates which estimate a vector of logits simultaneously could also be used for classification, [16], but again, if classification is the only consideration, one can argue that an appropriate multicategory SVM is preferable.

Recently [6] compared the GACV, the XA, five-fold cross validation and several other methods for tuning, using the standard two-category SVM on four data sets with large validation sets available. It appears from the information given that the authors may not have always found the minimizing (λ, σ) pair. However, we note the authors' conclusions here. With regard to the comparison between the GACV and the XA, essentially similar conclusions were obtained as those here, namely that they behaved similarly, one slightly better on some examples the other slightly better on the other examples. However five-fold cross validation appeared to have a better accuracy record on three of the examples, and was tied with the GACV on the fourth. Several other methods were studied, none of which appeared to be related to any leaving out one argument, and those did not perform well. The five-fold cross validation will cost more in computer time, but with todays computing speeds, that is not a real consideration. In some of our own experiments we have found that the ten-fold cross validation beats or is tied with the GACV. It is of some theoretical interest to understand what appears to be a systematic overestimation of λ when using the Gaussian kernel and tuning σ^2 along with λ, by methods which are based on the leaving-out-one arguments around (1.24), especially since corresponding tuning parameter estimates in penalized likelihood estimation generally appear to be unbiased in numerical examples.

.

References

[1] N. Aronszajn.Theory of reproducing kernels.*Trans. Am. Math. Soc.*, 68:337–404, 1950.

[2] O. Chapelle and V. Vapnik.Model selection for support vector machines.In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 12*, pages 230–237. MIT Press, 2000.

[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee.Choosing multiple parameters for support vector machines.*Machine Learning*, xx:xx, 2001.

[4] P. Craven and G. Wahba.Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation.*Numer. Math.*, 31:377–403, 1979.

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

[6] K. Duan, S. Keerthi, and A. Poo.Evaluation of simple performance measures for tuning svm hyperparameters.Technical Report CD-01-11, Dept. of Mechanical Engineering, National University of Singapore, Singapore, 2001.

[7] T. Furey, N. Cristianini, N. Duffy, D Bednarski, M. Schummer, and D. Haussler.Support vector machine classification and validation of cancer tissue samples using microarray expression data.*Bioinformatics*, 16:906–914, 2001.

[8] F. Gao, G. Wahba, R. Klein, and B. Klein.Smoothing spline ANOVA for multivariate Bernoulli observations, with applications to ophthalmology data, with discussion.*J. Amer. Statist. Assoc.*, 96:127–160, 2001.

[9] G.H. Golub, M. Heath, and G. Wahba.Generalized cross validation as a method for choosing a good ridge parameter.*Technometrics*, 21:215–224, 1979.

[10] T. Hastie, R. Tibshirani, and J. Friedman.*The Elements of Statistical Learning*.Springer, 2001.

[11] T. Jaakkola and D. Haussler.Probabilistic kernel regression models.In *Proceedings of the 1999 Conference on AI and Statistics*, 1999.

[12] T. Joachims.Estimating the generalization performance of an SVM efficiently.In *Proceedings of the International Conference on Machine Learning*, San Francisco, 2000. Morgan Kaufman.

[13] G. Kimeldorf and G. Wahba.Some results on Tchebycheffian spline functions.*J. Math. Anal. Applic.*, 33:82–95, 1971.

[14] Y. Lee, Y. Lin, and G. Wahba.Multicategory support vector machines.Technical Report 1043, Department of Statistics, University of Wisconsin, Madison WI, 2001.

[15] Y. Lee, Y. Lin, and G. Wahba.Multicategory support vector machines (preliminary long abstract).Technical Report 1040, Department of Statistics, University of Wisconsin, Madison WI, 2001.

[16] X. Lin.Smoothing spline analysis of variance for polychotomous response data.Technical Report 1003, Department of Statistics, University of Wisconsin, Madison WI, 1998.Available via G. Wahba's website.

[17] X. Lin, G. Wahba, D. Xiang, F. Gao, R. Klein, and B. Klein.Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV.*Ann. Statist.*, 28:1570–1600, 2000.

[18] Y. Lin.Support vector machines and the Bayes rule in classification.Technical Report 1014, Department of Statistics, University of Wisconsin, Madison WI, to appear, *Data Mining and Knowledge Discovery*, 1999.

[19] Y. Lin.On the support vector machine.Technical Report 1029, Department of Statistics, University of Wisconsin, Madison WI, 2000.

[20] Y. Lin, Y. Lee, and G. Wahba.Support vector machines for classification in nonstandard situations.Technical Report 1016, Department of Statistics, University of Wisconsin, Madison WI, 2000.To appear, *Machine Learning*.

[21] Y. Lin, G. Wahba, H. Zhang, and Y. Lee.Statistical properties and adaptive tuning of support vector machines.Technical Report 1022, Department of Statistics, University of Wisconsin, Madison WI, 2000.To appear, *Machine Learning*.

[22] M. Opper and O. Winther.Gaussian processes and svm: Mean field and leave-out-one.In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 311–326. MIT Press, 2000.

[23] B. Scholkopf, C. Burges, and A. Smola.*Advances in Kernel Methods-Support Vector Learning*.MIT Press, 1999.

[24] A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans.*Advances in Large Marin Classifiers*.MIT Press, 1999.

[25] M. Pontil T. Evgeniou and T. Poggio.Regularization networks and support vector machines.*Advances in Computational Mathematics*, 13:1–50, 2000.

[26] V. Vapnik.*The Nature of Statistical Learning Theory*.Springer, 1995.

[27] G. Wahba.Estimating derivatives from outer space.Technical Report 989, Mathematics Research Center, 1969.

[28] G. Wahba.*Spline Models for Observational Data*.SIAM, 1990.CBMS-NSF Regional Conference Series in Applied Mathematics, v. 59.

[29] G. Wahba.Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV.In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*, pages 69–88. MIT Press, 1999.

[30] G. Wahba, X. Lin, F. Gao, D. Xiang, R. Klein, and B. Klein.The bias-variance tradeoff and the randomized GACV.In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Information Processing Systems 11*, pages 620–626. MIT Press, 1999.

[31] G. Wahba, Y. Lin, Y. Lee, and H. Zhang.On the relation between the GACV and Joachims' $\xi\alpha$ method for tuning support vector machines, with extensions to the nonstandard case.Technical Report 1039, Statistics Department University of Wisconsin, Madison WI, 2001.

[32] G. Wahba, Y. Lin, and H. Zhang.Generalized approximate cross validation for support vector machines.In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 297–311. MIT Press, 2000.

[33] G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein.Structured machine learning for 'soft' classification with smoothing spline ANOVA and stacked tuning, testing and evaluation.In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 415–422. Morgan Kauffman, 1994.

[34] G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein.Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy.*Ann. Statist.*, 23:1865–1895, 1995.Neyman Lecture.

[35] D. Xiang and G. Wahba.A generalized approximate cross validation for smoothing splines with non-Gaussian data.*Statistica Sinica*, 6:675–692, 1996.