

Biometrika Trust

Probability Estimation for Large-Margin Classifiers

Author(s): Junhui Wang, Xiaotong Shen and Yufeng Liu

Source: *Biometrika*, Vol. 95, No. 1 (Mar., 2008), pp. 149-167

Published by: Oxford University Press on behalf of Biometrika Trust

Stable URL: <https://www.jstor.org/stable/20441449>

Accessed: 23-03-2020 22:03 UTC

REFERENCES

Linked references are available on JSTOR for this article:

https://www.jstor.org/stable/20441449?seq=1&cid=pdf-reference#references_tab_contents

You may need to log in to JSTOR to access the linked references.

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <https://about.jstor.org/terms>



Biometrika Trust, Oxford University Press are collaborating with JSTOR to digitize, preserve and extend access to *Biometrika*

Probability estimation for large-margin classifiers

By JUNHUI WANG AND XIAOTONG SHEN

School of Statistics, University of Minnesota, Minneapolis, Minnesota 55455, U.S.A.
wangjh@stat.umn.edu xshen@stat.umn.edu

AND YUFENG LIU

*Department of Statistics and Operations Research, Carolina Center for Genome Sciences,
University of North Carolina, Chapel Hill, North Carolina 27599, U.S.A.*
yfliu@email.unc.edu

SUMMARY

Large margin classifiers have proven to be effective in delivering high predictive accuracy, particularly those focusing on the decision boundaries and bypassing the requirement of estimating the class probability given input for discrimination. As a result, these classifiers may not directly yield an estimated class probability, which is of interest itself. To overcome this difficulty, this article proposes a novel method for estimating the class probability through sequential classifications, by using features of interval estimation of large-margin classifiers. The method uses sequential classifications to bracket the class probability to yield an estimate up to the desired level of accuracy. The method is implemented for support vector machines and ψ -learning, in addition to an estimated Kullback–Leibler loss for tuning. A solution path of the method is derived for support vector machines to reduce further its computational cost. Theoretical and numerical analyses indicate that the method is highly competitive against alternatives, especially when the dimension of the input greatly exceeds the sample size. Finally, an application to leukaemia data is described.

Some key words: Function estimation; High dimension and low sample size; Interval estimate; Tuning; Weighting.

1. INTRODUCTION

In the statistics literature, classification is often treated as a problem of density estimation through regression; that is, the class probability given input is estimated, yielding classification by thresholding. This practice seems to undermine the fact that classification is generally easier than regression, because the former is a problem of interval estimation rather than a point estimation problem. This is evident from recent successes in large-margin classification such as support vector machines (Cortes & Vapnik, 1995) and ψ -learning (Shen et al., 2003), where many large-margin classifiers yield high performance by focusing directly on classification, bypassing the estimation of the class probability. However, knowledge about the class probability itself may be of significant scientific interest, indicating the strength or confidence of the outcome of classification. In this article, we bridge the gap by estimating the class probability through interval estimation in classification, allowing a large-margin classifier to enjoy the capability of regression while maintaining its high generalization ability and computational advantage.

In binary classification, a decision function f is estimated from a training sample $Z_i = (X_i, Y_i)$, $i = 1, \dots, n$, independent and identically distributed according to an unknown probability

distribution $P(x, y)$, where $X_i \in \mathbb{R}^d$ is a d -dimensional input and the output Y_i is labelled as ± 1 . For any input x , classifier $\text{sign}\{f(x)\}$ estimates the label of x . Within the framework of large-margin classification, estimation of the class probability has been investigated. Steinwart (2003) and Bartlett & Tewari (2007) show that replacing the large-margin loss with some differentiable loss leads asymptotically to conditional probability estimation. Platt (1999) assumes a sigmoid link function between the conditional distribution $p(x) = \text{pr}(Y = 1|X = x)$ and a large-margin classifier f , in the form of

$$p(x) = \frac{1}{1 + \exp\{Af(x) + B\}}, \quad (1)$$

with parameters A and B estimated by minimizing the cross-entropy error. Despite its empirical success, statistical properties of the approach have not yet been investigated. There is no solid evidence that the link function specified in (1) should be used to estimate $p(x)$ through $f(x)$. In fact, Lin (2002) shows that the optimal $f(x)$ estimated by support vector machines is $\text{sign}\{p(x) - 1/2\}$, which implies that the classifier f might only be concerned about whether $p(x)$ is greater than $1/2$ or not.

In this article, we estimate p for large-margin classifiers without imposing any assumption on the relationship between p and f as in (1). It is known that $\text{sign}\{\hat{f}(x)\} > 0$ estimates $\text{sign}\{p(x) - 1/2\} > 0$ for a large-margin classifier $\text{sign}(\hat{f})$ such as support vector machines (Lin, 2002). On this basis, we design a sequence of weighted classifications, corresponding to a refined partition of $[0, 1]$, to locate the subinterval that contains $p(x)$ for any fixed x . This approach is illustrated with data of high dimension but low sample size, typical of microarray experiments.

The proposed method is implemented for support vector machines and ψ -learning. To eliminate dependence of the method on a tuning parameter, we propose a method of model selection through the concept of a covariance penalty (Efron, 2004) and the technique of data perturbation (Shen & Huang, 2006). Moreover, we derive an efficient solution path algorithm for the proposed method via support vector machines to reduce its computational cost. We derive rates of convergence of the proposed estimator for large-margin classification, and show in an example that the accuracy of probability estimation for ψ -learning is of order $n^{-1/2}(\log n)^{3/2}$, whereas its classification accuracy is of order $n^{-1}(\log n)^3$. This confirms the aforementioned phenomenon that classification is usually easier than density estimation. Our numerical analyses suggest that the proposed method is highly competitive against alternatives.

2. ESTIMATION

2.1. Large-margin classifiers

A large-margin classifier minimizes a cost function in f over a decision function class \mathcal{F} :

$$\min_{f \in \mathcal{F}} n^{-1} \sum_{i=1}^n L\{y_i f(x_i)\} + \lambda J(f), \quad (2)$$

where $J(f)$ is a regularization term for penalizing model complexity, $\lambda > 0$ is the degree of penalization and $L(z)$ is a margin loss that is a function of the functional margin $yf(x)$; for instance, $L(z) = (1 - z)_+$ is the hinge loss for support vector machines, and $L(z) = \psi(z)$ with $\psi(z) = 1 - \text{sign}(z)$ if $z \geq 1$ or $z < 0$, and $2(1 - z)$ otherwise is the ψ -loss for ψ -learning (Shen et al., 2003). A margin loss $L(z)$ is said to be large-margin if $L(z)$ is nonincreasing in z , penalizing small margin values. In linear classification, f is linear; in kernel classification, f uses a kernel representation $(n\lambda)^{-1} \sum_{i=1}^n \theta_i y_i K(x_i, x) + \beta_0$ with $K(\cdot, \cdot)$ a kernel function. In the kernel case,

$J(f) = (2n^2\lambda^2)^{-1} \sum_{i=1}^n \sum_{i'=1}^n \theta_i \theta_{i'} y_i y_{i'} K(x_i, x_{i'})$, and \mathcal{F} is a reproducing kernel Hilbert space (Wahba, 1990) induced by $K(\cdot, \cdot)$. The weighted version of (2) is

$$\min_{f \in \mathcal{F}} n^{-1} \left[(1 - \pi) \sum_{y_i=1} L\{y_i f(x_i)\} + \pi \sum_{y_i=-1} L\{y_i f(x_i)\} \right] + \lambda J(f), \quad (3)$$

which reduces to (2) when $\pi = 1/2$. The loss in (3) permits a treatment of an unequal number of training samples or unequal costs for positive and negative misclassifications in margin classification, where $(\pi, 1 - \pi)$ are the known costs for the negative and positive classes with $0 \leq \pi \leq 1$; see Lin et al. (2002) for a discussion. Minimizing (3) with respect to $f \in \mathcal{F}$ yields $\hat{f}_\pi(x)$, and thus the classifier $\text{sign}\{\hat{f}_\pi(x)\}$, which is an estimate of the Bayes rule $\hat{f}_\pi(x) = \text{sign}\{f_\pi(x)\}$ with $f_\pi(x) = p(x) - \pi$.

Lemma 1 below constitutes a basis for our proposed method. In (2), when $n \rightarrow \infty$, the first component of (3) approaches

$$E[S(Y)L\{Yf(X)\}] = E[(1 - \pi)I(Y = 1)L\{Yf(X)\} + \pi I(Y = -1)L\{Yf(X)\}], \quad (4)$$

where $I(\cdot)$ is the indicator function, and $S(Y)$ is $1 - \pi$ if $Y = 1$, and π otherwise.

LEMMA 1. *With $L(z) = (1 - z)_+$ or $\psi(z)$, minimizing (4) with respect to f yields the Bayes rule $\hat{f}_\pi(x)$. Moreover, $\hat{f}_\pi(x)$ is nonincreasing in π .*

2.2. Estimation

Our proposed method is designed to estimate $p(x)$ at any x with x not necessarily being one of the observed values. First, we construct a uniform partition of $[0, 1]$ with the two endpoints 0 and 1 included; that is, $0 = \pi_1 < \dots < \pi_{m+1} = 1$ for any given integer $m > 0$ that determines the estimation precision. By construction, one and only one of the subintervals brackets $p(x)$. Ideally, one can use the monotonicity property of $\text{sign}\{\hat{f}_\pi(x)\}$, see Lemma 1, to compute $p(x)$ rapidly at one x -value. However, the monotonicity property may not hold empirically, and in addition it is desirable to compute $p(x)$ at multiple x -values simultaneously. We therefore examine one interval at a time and train $m + 1$ weighted margin classifiers with π_j , $j = 1, \dots, m + 1$, to identify the interval capturing $p(x)$. This is achieved by checking if $\text{sign}\{\hat{f}_{\pi_j}(x)\} > 0$, for $j = 1, \dots, m + 1$. Moreover, when the monotonicity property of $\text{sign}\{\hat{f}_\pi(x)\}$ does not hold for a specific set of data, there exists $1 \leq j \leq m$ such that $\text{sign}\{\hat{f}_{\pi_j}(x)\} = -1$ but $\text{sign}\{\hat{f}_{\pi_{j+1}}(x)\} = 1$, and hence that more than one interval captures $\hat{p}(x)$. This is especially so when the size of the training sample is not large. To overcome this difficulty, we define $\pi^* = \arg \max_{\pi_j} [\text{sign}\{\hat{f}_{\pi_j}(x)\} = 1]$ and $\pi_* = \arg \min_{\pi_j} [\text{sign}\{\hat{f}_{\pi_j}(x)\} = -1]$, with $0 \leq \pi_*, \pi^* \leq 1$. Then the proposed estimate $\hat{p}(x)$ is defined as $\frac{1}{2}(\pi_* + \pi^*)$.

The proposed estimator \hat{p} can be computed via Algorithm 1.

ALGORITHM 1.

Step 1. Initialize $\pi_j = (j - 1)/m$, for $j = 1, \dots, m + 1$.

Step 2. Train a weighted margin classifier for π_j as in (3), for $j = 1, \dots, m + 1$.

Step 3. Estimate labels of x by $\text{sign}\{\hat{f}_{\pi_j}(x)\}$.

Step 4. Sort $\text{sign}\{\hat{f}_{\pi_j}(x)\}$, $j = 1, \dots, m + 1$, to compute $\pi^ = \max[\pi_j : \text{sign}\{\hat{f}_{\pi_j}(x)\} = 1]$, $\pi_* = \min[\pi_j : \text{sign}\{\hat{f}_{\pi_j}(x)\} = -1]$. The estimated class probability is $\hat{p}(x) = \frac{1}{2}(\pi^* + \pi_*)$.*

Algorithm 1 is designed for any large-margin classifier, including weighted support vector machines (Lin et al., 2002) and weighted ψ -learning. To train weighted support vector machines, any software with a quadratic programming routine can be employed. To train weighted ψ -learning, we follow the technique of Liu, Shen & Wong (2005), who use the difference convex algorithm to solve the nonconvex optimization problem through sequential quadratic programming. The idea of the difference convex algorithm is to decompose a nonconvex objective function into a difference of two convex functions, and solve the nonconvex problem by solving sequential convex problems. According to Liu, Shen & Wong (2005), the difference convex algorithm is appropriate for the ψ -loss because of its encouraging numerical performance and its fast convergence speed.

Furthermore, a precision parameter m needs to be prespecified in Algorithm 1, achieving a trade-off between the precision of \hat{p} and the number of weighted classifiers to be trained. Evidently, a large value of m yields better precision but increases computational cost. In implementation, it is recommended that $m = \lfloor n^{1/2} \rfloor$, the largest integer not greater than $n^{1/2}$. Our simulation suggests that this choice is satisfactory. Of course, a data-driven choice of m can be derived by minimizing an estimated loss function, at an expense of increased computational cost, as discussed next.

3. ESTIMATING GENERALIZED KULLBACK–LEIBLER LOSS AND TUNING

3.1. Estimation of generalized Kullback–Leibler loss

The overall performance of \hat{p} in estimating p is evaluated by its closeness to p in terms of the generalized Kullback–Leibler loss,

$$\text{GKL}(p, \hat{p}) = E \left[p(X) \log \frac{p(X)}{\hat{p}(X)} + \{1 - p(X)\} \log \frac{1 - p(X)}{1 - \hat{p}(X)} \right], \quad (5)$$

where the expectation is taken with respect to randomness in X , which differs from Kullback–Leibler loss in that (5) is averaged over a random X that has the same distribution as X_i , $i = 1, \dots, n$. The corresponding comparative Kullback–Leibler loss, after terms in (5) unrelated to \hat{p} are omitted, is

$$\text{GKL}^c(p, \hat{p}) = -E[p(X) \log\{\hat{p}(X)\} + \{1 - p(X)\} \log\{1 - \hat{p}(X)\}].$$

Since $E\{\frac{1}{2}(Y + 1)|X\} = p(X)$, the empirical version of $\text{GKL}^c(p, \hat{p})$ is

$$\text{EGKL}(\hat{p}) = -n^{-1} \sum_{i=1}^n \left[\frac{1}{2}(Y_i + 1) \log \hat{p}(X_i) + \left\{ 1 - \frac{1}{2}(Y_i + 1) \right\} \log\{1 - \hat{p}(X_i)\} \right], \quad (6)$$

which measures the goodness-of-fit of \hat{p} . To penalize overfitting in $\text{EGKL}(\hat{p})$, $\text{GKL}^c(p, \hat{p})$ is estimated by choosing the optimal estimator from a class of candidate estimators of the form

$\text{EGKL}(\hat{p}) + \zeta(\hat{p}, X^n)$, where $\zeta(\hat{p}, X^n) \geq 0$ is a penalty depending on $X^n = \{X_i\}_{i=1}^n$, to be determined optimally by minimizing

$$E[\text{GKL}(p, \hat{p}) - \{\text{EGKL}(\hat{p}) + \zeta(\hat{p}, X^n)\}]^2. \quad (7)$$

Estimation of the Kullback–Leibler loss has been investigated in Shen & Huang (2006), but that of the generalized Kullback–Leibler loss involving random input X has not yet been explored in the literature. Breiman & Spector (1992) argued that ignoring randomness in linear regression could lead to highly biased estimation of the prediction error.

THEOREM 1. *The optimal $\zeta_o(\hat{p}, X^n)$ that minimizes (7) with respect to $\zeta(\hat{p}, X^n)$ is*

$$\zeta_o(\hat{p}, X^n) = n^{-1} \sum_{i=1}^n \text{cov}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)\}|X^n] + D_n(\hat{p}, X^n), \quad (8)$$

where $\phi(p) = \text{logit}(p)$ and $D_n(\hat{p}, X^n) = E\{\Delta(p, \hat{p}; X^n) - \bar{\Delta}(p, \hat{p})|X^n\}$, with $\bar{\Delta}(p, \hat{p}) = E[\{1 - p(X)\} \log\{1 - \hat{p}(X)\} + p(X) \log \hat{p}(X)]$ and $\Delta(p, \hat{p}; X^n) = n^{-1} \sum_{i=1}^n [\{1 - p(X_i)\} \log\{1 - \hat{p}(X_i)\} + p(X_i) \log \hat{p}(X_i)]$.

In (8), $n^{-1} \sum_{i=1}^n \text{cov}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)\}|X^n]$ evaluates the accuracy of estimating \hat{p} from X^n , which is a covariance penalty in Efron (2004) and the generalized degrees of freedom in Shen & Huang (2006). The term $D_n(\hat{p}, X^n)$, on the other hand, is a correction that adjusts the effect of random input X on prediction and needs to be estimated (Breiman & Spector, 1992; Breiman, 1992).

Therefore, we propose to estimate $\text{GKL}^c(p, \hat{p})$ by

$$\text{GKL}^c(p, \hat{p}) = \text{EGKL}^c(\hat{p}) + n^{-1} \sum_{i=1}^n \text{c\hat{ov}}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)\}|X^n] + \hat{D}_n(\hat{p}, X^n), \quad (9)$$

where $\text{c\hat{ov}}$ and \hat{D}_n are estimators of the covariance and D_n respectively. To construct approximately unbiased estimators for $\text{cov}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)\}|X^n]$ and $D_n(\hat{p}, X^n)$, we adopt the technique of data perturbation as in Wang & Shen (2006), as follows.

First, we perturb X_i , $i = 1, \dots, n$, via its empirical distribution \hat{F} , and then flip the corresponding label Y_i with a certain probability, given the perturbed X_i . This generates perturbations for assessing the accuracy of probability estimation. To be more precise, for $i = 1, \dots, n$, let

$$X_i^* = \begin{cases} X_i & \text{with probability } 1 - \tau, \\ \tilde{X}_i & \text{with probability } \tau, \end{cases} \quad Y_i^* = \begin{cases} Y_i & \text{with probability } 1 - \tau, \\ \tilde{Y}_i & \text{with probability } \tau, \end{cases} \quad (10)$$

in which \tilde{X}_i is sampled from \hat{F} , $0 \leq \tau \leq 1$ is the size of perturbation, and $\tilde{Y}_i \sim \text{Bi}\{1, \hat{p}(X_i^*)\}$, with $\hat{p}(X_i^*)$ an initial probability estimate of $E(Y_i|X_i^*)$. Here and in the sequel, we fix τ to be 0.5.

Denote respectively by E^* and cov^* the conditional expectation and covariance, given $X^{*n} = \{X_i^*\}_{i=1}^n$, $Y^n = \{Y_i\}_{i=1}^n$. Then we estimate $\text{cov}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)\}|X^n]$ by

$$\text{c\hat{ov}}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)\}|X^n] = \frac{1}{K\{Y_i, \hat{p}(X_i^*)\}} \text{cov}^*[(Y_i^* + 1)/2, \phi\{\hat{p}^*(X_i^*)\}|X^{*n}], \quad (11)$$

with \hat{p}^* obtained by applying the proposed probability estimation method to $\{X_i^*, Y_i^*\}_{i=1}^n$, and

$$K\{Y_i, \hat{p}(X_i^*)\} = \tau + \tau(1 - \tau) \frac{\{Y_i - \hat{p}(X_i^*)\}^2}{\hat{p}(X_i^*)\{1 - \hat{p}(X_i^*)\}}.$$

Meanwhile $D_n(X^n, \hat{p})$ is estimated by

$$\hat{D}_n(\hat{p}, X^n) = E^* \{ \Delta(\hat{p}, \hat{p}^*; X^{*n}) - \Delta(\hat{p}, \hat{p}^*; X^n) | X^{*n} \}. \quad (12)$$

With (11) and (12), we obtain $\text{GKL}^c(p, \hat{p})$ in (9), which can be computed by Monte Carlo approximations. First, generate D perturbed samples $X^{*ln} = \{X_i^{*l}\}_{i=1}^n$ according to (10) for $l = 1, \dots, D$. Secondly, for each sample $\{X_i^{*l}\}_{i=1}^n$, generate D perturbed samples $\{Y_i^{*lm}\}_{i=1}^n$ according to (10) for $m = 1, \dots, D$. Furthermore, for $l, m = 1, \dots, D$ and $i = 1, \dots, n$, compute

$$\text{cov}^*[(Y_i^* + 1)/2, \phi\{\hat{p}^*(X_i^*)|X^{*n}\}] = \frac{1}{D^2 - 1} \sum_{l,m=1}^D \phi\{\hat{p}^{*lm}(X_i^{*l})\} (Y_i^{*lm} - \bar{Y}_i^* + 1)/2,$$

in which \hat{p}^{*lm} is computed by applying the proposed probability estimation method to $\{X_i^{*l}, Y_i^{*lm}\}_{i=1}^n$, and $\bar{Y}_i^* = \frac{1}{D^2} \sum_{l,m=1}^D Y_i^{*lm}$. Now (11) and (12) are approximated by the corresponding Monte Carlo approximation; that is,

$$\begin{aligned} \text{cov}[(Y_i + 1)/2, \phi\{\hat{p}(X_i)|X^n\}] &\simeq \frac{1}{2(D^2 - 1)} \sum_{l,m=1}^D \frac{1}{K\{Y_i, \hat{p}(X_i^{*l})\}} \phi\{\hat{p}^{*lm}(X_i^{*l})\} \\ &\quad \otimes \{(Y_i^{*lm} - \bar{Y}_i^*) + 1\}, \end{aligned} \quad (13)$$

$$\hat{D}_n(\hat{p}, X^n) \simeq \frac{1}{D^2 - 1} \sum_{l,m=1}^D \{ \Delta(\hat{p}, \hat{p}^{*lm}; X^{*l,n}) - \Delta(\hat{p}, \hat{p}^{*lm}; X^n) \}. \quad (14)$$

By the law of large numbers, (13) and (14) converge to (11) and (12) respectively, as $D \rightarrow \infty$. In practice, we recommend D to be at least $\lfloor n^{1/2} \rfloor$ to ensure the precision of the Monte Carlo approximation. Plugging (13) and (14) into (9), we obtain the final estimate of $\text{GKL}^c(p, \hat{p})$.

3.2. Tuning

The performance of \hat{p}_λ depends on λ , and hence optimal selection of λ becomes important; here \hat{p} is written as \hat{p}_λ to indicate its dependence on λ . Minimization of (9) over the range of $\lambda > 0$ yields the optimal λ , denoted by $\hat{\lambda}$.

Conditions 1–3 concern optimality of selecting λ through (9). They are analogous to those in Wang & Shen (2006) but different in that consistency is not required here.

Condition 1. For any positive integers m, n and some $\delta > 0$, $E \sup_{\tau \in (0, \delta)} |\hat{\zeta}(\hat{p}_\lambda, X^n)| < +\infty$.

Condition 2. In probability, $\lim_{m \rightarrow \infty} \lim_{n \rightarrow \infty} \sup_\lambda |\text{GKL}(p, \hat{p}_\lambda) / E\{\text{GKL}(p, \hat{p}_\lambda)\} - 1| = 0$.

Condition 3. For any positive integers m and n , $\inf_\lambda E\{\text{GKL}(p, \hat{p}_\lambda)\} > 0$.

THEOREM 2. Under Conditions 1–3, for $\hat{\lambda}$ the minimizer of (9), we have

$$\lim_{m, n \rightarrow \infty} \left\{ \lim_{\tau \rightarrow 0+} \text{GKL}(p, \hat{p}_{\hat{\lambda}}) / \inf_{0 < \lambda < \infty} \text{GKL}(p, \hat{p}_\lambda) \right\} = 1.$$

Theorem 2 says that the ideal optimal performance $\inf_{0 < \lambda < \infty} \text{GKL}(p, \hat{p}_\lambda)$ can be realized by $\text{GKL}(p, \hat{p}_{\hat{\lambda}})$ when $\tau \rightarrow 0+$ and $m, n \rightarrow \infty$. Also, the proposed tuning method is optimal against other tuning methods in terms of the generalized Kullback–Leibler loss.

4. SOLUTION PATH FOR SUPPORT VECTOR MACHINES

This section develops a solution path algorithm for support vector machines to facilitate computation. One direct benefit of the algorithm is that, given an initial solution for the path algorithm, the m support vector machine classifiers with different weights in Step 2 there can be trained at essentially the cost of training one support vector machine classifier. Our solution path algorithm here differs from that of Hastie et al. (2004) in that it is with respect to π instead of λ .

To derive the solution path of (3) as a function of π , we express the solution of (3) with $L(z) = (1 - z)_+$ as $\hat{f}_\pi(x) = \beta_0 + (n\lambda)^{-1} \sum_{i=1}^n \theta_i(\pi) y_i K(x, x_i)$ by the reproducing kernel Hilbert space representation theorem of Kimeldorf & Wahba (1971). As is to be seen, $\theta(\pi) = (\theta_1(\pi), \dots, \theta_n(\pi))^T$ is piecewise linear in π for any fixed value of λ , where $\theta_i \equiv \theta_i(\pi) \in [0, S(y_i)]$ with $S(y_i) = 1 - \pi$ if $y_i = 1$ and $S(y_i) = \pi$ otherwise. On this basis, we derive an efficient algorithm for computing an exact solution path of $\theta(\pi)$, and thus of $\hat{f}_\pi(x)$, for $0 < \pi < 1$.

Before deriving an algorithm for computing the solution path, we rewrite (3), after introducing slack variables ξ_i , $i = 1, \dots, n$, as

$$\min_{\beta_0, \theta} \sum_{i=1}^n S(y_i) \xi_i + \frac{1}{2n\lambda} \theta^T \mathbb{K}_y \theta \quad (15)$$

subject to $1 - y_i f(x_i) \leq \xi_i$ and $\xi_i \geq 0$, $i = 1, \dots, n$, where \mathbb{K}_y is an $n \times n$ matrix with its (i, i') element $y_i y_{i'} K(x_i, x_{i'})$. Then (15) yields a primal function $L_p \equiv \sum_{i=1}^n S(y_i) \xi_i + (2n\lambda)^{-1} \theta^T \mathbb{K}_y \theta + \sum_{i=1}^n \alpha_i \{1 - y_i f(x_i) - \xi_i\} - \sum_{i=1}^n \gamma_i \xi_i$, where $\alpha_i \geq 0$ and $\gamma_i \geq 0$ are Lagrange multipliers. Setting the derivatives of L_p to be zero, we obtain

$$\frac{\partial L_p}{\partial \theta} : \theta_i = \alpha_i; \quad \frac{\partial L_p}{\partial \beta_0} : \sum_{i=1}^n \alpha_i y_i = 0; \quad \frac{\partial L_p}{\partial \xi_i} : \alpha_i = S(y_i) - \gamma_i, \quad (16)$$

with the Karush–Kuhn–Tucker conditions

$$\alpha_i \{1 - y_i f(x_i) - \xi_i\} = 0; \quad \gamma_i \xi_i = 0. \quad (17)$$

From (16), $0 \leq \alpha_i \leq (1 - \pi)$ if $y_i = 1$ and $0 \leq \alpha_i \leq \pi$ if $y_i = -1$, because $\gamma_i \geq 0$ for $i = 1, \dots, n$, implying that (1) $y_i f(x_i) > 1 \Rightarrow \xi_i = 0, \alpha_i = 0$; (2) $y_i f(x_i) < 1 \Rightarrow \xi_i \neq 0, \gamma_i = 0, \alpha_i = S(y_i)$; and (3) $y_i f(x_i) = 1 \Rightarrow \xi_i = 0, \alpha_i \in [0, S(y_i)]$. Note that $\theta_i = \alpha_i$ for all $0 \leq \pi \leq 1$.

Following an idea of Hastie et al. (2004), we define three sets to track the solution path in π based on preceding relationships: $\mathcal{E} = \{i : y_i f(x_i) = 1, 0 \leq \theta_i \leq S(y_i)\}$ represents an elbow; $\mathcal{L} = \{i : y_i f(x_i) < 1, \theta_i = S(y_i)\}$ represents left of the elbow; and $\mathcal{R} = \{i : y_i f(x_i) > 1, \theta_i = 0\}$ represents right of the elbow. For \mathcal{L} and \mathcal{R} , θ_i remains known for their elements. Therefore, the algorithm will focus on points resting at the elbow \mathcal{E} .

For our path algorithm, a value of π near the origin is initialized to compute the solution of $\theta(\pi)$ through Algorithm 1, and then the value of π increases towards 1. As π increases, points move from left of the elbow to the right of the elbow or vice versa. In this process, their corresponding θ_i 's change from $S(y_i)$ towards 0, implying that the points must linger on the elbow by continuity while their θ_i 's change from $S(y_i)$ to 0.

The algorithm thus tracks the elements in \mathcal{E} , satisfying $y_i f(x_i) = 1$ with $\theta_i \in [0, S(y_i)]$. As π increases, when one element begins to change, an event occurs. Such an event can be categorized as follows: an element from \mathcal{L} has just entered into \mathcal{E} with θ_i to be initially $S(y_i)$; an element from \mathcal{R} has just entered into \mathcal{E} with θ_i to be initially 0; and an element or elements from \mathcal{E} has or have just left \mathcal{E} to join either \mathcal{L} or \mathcal{R} .

In what follows, we use the superscript ℓ to index the preceding sets \mathcal{E}^ℓ , \mathcal{L}^ℓ and \mathcal{R}^ℓ , as well as parameter and function values $(\theta_i^\ell, \beta_0^\ell, \pi^\ell)$ and f^ℓ , immediately after the ℓ th event has

occurred. For convenience, write $\beta_{0,\lambda} = n\lambda\beta_0$ and $\beta_{0,\lambda}^\ell = n\lambda\beta_0^\ell$. Note that $f(x) = (n\lambda)^{-1}\{\beta_{0,\lambda} + \sum_{i=1}^n \theta_i y_i K(x, x_i)\}$. Then, for $\pi^\ell < \pi < \pi^{\ell+1}$,

$$\begin{aligned} f(x) &= \{f(x) - f^\ell(x)\} + f^\ell(x) = \frac{1}{n\lambda} \left\{ (\beta_{0,\lambda} - \beta_{0,\lambda}^\ell) + \sum_{i=1}^n (\theta_i - \theta_i^\ell) y_i K(x, x_i) \right\} + f^\ell(x) \\ &= \frac{1}{n\lambda} \left\{ (\beta_{0,\lambda} - \beta_{0,\lambda}^\ell) + \sum_{i \in \mathcal{E}^\ell} (\theta_i - \theta_i^\ell) y_i K(x, x_i) \right\} - \frac{1}{n\lambda} \sum_{i \in \mathcal{L}^\ell} (\pi - \pi^\ell) K(x, x_i) + f^\ell(x), \end{aligned}$$

where the second equality uses the fact that the θ_i 's are fixed for elements in \mathcal{R}^ℓ and are either $(1 - \pi)$ or π for elements in \mathcal{L}^ℓ , and all elements remain in their respective sets. Let $|\mathcal{E}^\ell| = n_{\mathcal{E}^\ell}^\ell$. Then, for any element k staying in \mathcal{E}^ℓ ,

$$\frac{y_k}{n\lambda} \left\{ (\beta_{0,\lambda} - \beta_{0,\lambda}^\ell) + \sum_{i \in \mathcal{E}^\ell} (\theta_i - \theta_i^\ell) y_i K(x_k, x_i) \right\} - \frac{y_k}{n\lambda} \sum_{i \in \mathcal{L}^\ell} (\pi - \pi^\ell) K(x_k, x_i) + y_k f^\ell(x_k) = 1.$$

This implies that $v_0 y_k + \sum_{i \in \mathcal{E}^\ell} v_i y_i K(x_k, x_i) = (\pi - \pi^\ell) y_k \sum_{i \in \mathcal{L}^\ell} K(x_k, x_i)$, for all $k \in \mathcal{E}^\ell$ with $v_i = (\theta_i - \theta_i^\ell)$ and $v_0 = (\beta_{0,\lambda} - \beta_{0,\lambda}^\ell)$. By (16), $\sum_{i \in \mathcal{E}^\ell} v_i y_i = (\pi - \pi^\ell) n_{\mathcal{L}^\ell}^\ell$. Thus we solve a system of $n_{\mathcal{E}^\ell}^\ell + 1$ linear equations involving the $n_{\mathcal{E}^\ell}^\ell + 1$ unknown variables v_i and v_0 .

Let \mathbb{K}_y^ℓ be an $n_{\mathcal{E}^\ell}^\ell \times n_{\mathcal{E}^\ell}^\ell$ matrix with its entries $y_k y_i K(x_k, x_i)$ for $i, k \in \mathcal{E}^\ell$, let $y_{\mathcal{E}^\ell}^\ell$ be the vector with components $y_k, k \in \mathcal{E}^\ell$, let v be a vector with components v_i for $i \in \mathcal{E}^\ell$, and let K_y^ℓ be a vector with components $y_k \sum_{i \in \mathcal{L}^\ell} K(x_k, x_i), k \in \mathcal{E}^\ell$. Then

$$v_0 y_{\mathcal{E}^\ell}^\ell + \mathbb{K}_y^\ell v = (\pi - \pi^\ell) K_y^\ell; \quad v' y_{\mathcal{E}^\ell}^\ell = (\pi - \pi^\ell) n_{\mathcal{L}^\ell}^\ell. \quad (18)$$

To simplify (18) further, we let

$$\mathbb{K}_y^* = \begin{pmatrix} 0 & (y_{\mathcal{E}^\ell}^\ell)' \\ y_{\mathcal{E}^\ell}^\ell & \mathbb{K}_y^\ell \end{pmatrix}, \quad v^* = \begin{pmatrix} v_0 \\ v \end{pmatrix}, \quad K_y^* = \begin{pmatrix} n_{\mathcal{L}^\ell}^\ell \\ K_y^\ell \end{pmatrix}.$$

Then equations in (18) can be combined to be $\mathbb{K}_y^* v^* = (\pi - \pi^\ell) K_y^*$. If \mathbb{K}_y^* has full rank, define $b^* = (\mathbb{K}_y^*)^{-1} K_y^*$ to yield

$$\beta_{0,\lambda} = \beta_{0,\lambda}^\ell + (\pi - \pi^\ell) b_0^*; \quad \theta_i = \theta_i^\ell + (\pi - \pi^\ell) b_i^*, \quad \text{for all } i \in \mathcal{E}^\ell. \quad (19)$$

Thus, for $\pi^\ell < \pi < \pi^{\ell+1}$, the θ_i and $\beta_{0,\lambda}$ proceed linearly in π . Also,

$$f(x) = f^\ell(x) + (\pi - \pi^\ell) h^\ell(x), \quad (20)$$

where $h^\ell(x) = (n\lambda)^{-1}\{b_0^* + \sum_{i \in \mathcal{E}^\ell} b_i^* y_i K(x, x_i) - \sum_{i \in \mathcal{L}^\ell} K(x, x_i)\}$.

Given π_ℓ , (19) and (20) permit computation of $\pi_{\ell+1}$, the π at which the next event occurs. This will be the smallest π greater than π_ℓ such that either θ_i for $i \in \mathcal{E}^\ell$ reaches $S(y_i)$ or 0, or one of the elements in \mathcal{R} or \mathcal{L} reaches the elbow. The latter event occurs for element x_k when $\pi = \pi^\ell + \{1 - y_k f^\ell(x_k)\} \{y_k h^\ell(x_k)\}^{-1}$, for all $k \in \mathcal{R}^\ell \cup \mathcal{L}^\ell$. Termination occurs when π has become sufficiently close to 1.

5. NUMERICAL RESULTS

5.1. Preamble

This section examines the effectiveness of the proposed method, and compares it to some popular competitors, mainly the Platt method (Platt, 1999), penalized logistic regression and

the nearest neighbour method, although these methods may have different objectives. A primary comparison is made with respect to accuracy of probability estimation. However, when a method is suited to classification, its accuracy with respect to the generalization error is examined as well.

In simulated examples, the generalized Kullback–Leibler loss over a test set is used for evaluating probability estimation when the true p is known. In benchmark examples when p is unknown, the cross entropy error over a test set is used, defined as

$$\text{CRE}(\hat{p}) = -\frac{1}{\#\{\text{test set}\}} \sum_{\text{test set}} \left[\frac{1}{2}(1 + y_i) \log\{\hat{p}(x_i)\} + \frac{1}{2}(1 - y_i) \log\{1 - \hat{p}(x_i)\} \right],$$

where $\#\{A\}$ is the cardinality of set A .

5.2. Simulation

The proposed method is examined for support vector machines and ψ -learning in the linear and Gaussian kernel cases, where support vector machines are trained using the svm routine in package e1071 of R2.1.1 and ψ -learning is carried out as in Liu et al. (2005) based on the difference convex algorithm. For penalized logistic regression, training is performed through routine StepPlr in R2.1.1. The nearest neighbour method is implemented as in R2.1.1.

For our method, the Platt method and penalized logistic regression, we seek the optimal λ by minimizing (9) through a grid search over the interval $[10^{-3}, 10^3]$ with ten equally-spaced points in each interval $(10^j, 10^{j+1}]$; $j = -3, \dots, 2$. For Gaussian kernel support vector machines, σ is set to be the median distance between the positive and negative classes (Jaakkola et al., 1999), because λ plays a similar role to that of σ^2 and it is easier to optimize with respect to λ with σ^2 fixed. For the nearest neighbour method, we examine methods based on 4, 9, 16, 25 nearest neighbours and report the best performance.

Example 1. Data $\{(X_{i1}, X_{i2}, Y_i); i = 1, \dots, 1000\}$ are generated as follows. First, $\{(X_{i1}, X_{i2}); i = 1, \dots, 1000\}$ are sampled from the uniform distribution over a unit disk $\{(X_1, X_2) : X_1^2 + X_2^2 \leq 1\}$. Next, we set $Y_i = 1$ if $X_{i1} \geq 0$ and $Y_i = -1$ otherwise, $i = 1, \dots, n$. Finally, we randomly choose 20% of the sample and flip their labels to generate the nonseparable case. This yields the first simulated example, in which 100 and 900 randomly selected cases are used for training and testing, respectively.

Example 2. Data $\{(X_{i1}, X_{i2}, Y_i); i = 1, \dots, 1000\}$ are generated as follows. First, we randomly assign ± 1 to $\{Y_i; i = 1, \dots, 1000\}$ with equal probability. Next, we generate X_{i1} from the uniform distribution over $[0, 2\pi]$, and set $X_{i2} = Y_i\{\sin(X_{i1}) + 1 + Z_i\}$, where $Z_i \sim N(0, 0.1^2)$. This yields the second simulated example with 100 randomly selected cases used for training and the remaining 900 for testing.

With regard to probability estimation, the value of generalized Kullback–Leibler loss in (5) is averaged over 100 simulation replications. Unfortunately, however, for penalized logistic regression and the nearest neighbour method, the value of generalized Kullback–Leibler loss is infinity when the estimated probability becomes exactly 0 or 1. To overcome this difficulty, we average over only 66 nondegenerate replications for penalized logistic regression, and leave a blank for the nearest neighbour method in Example 2 in the case of nondegenerate replications. With regard to classification, the test error is used to measure the performance, averaged over nondegenerate replications. Finally, the value of cross entropy error is given to see how well the cross entropy error estimates the generalized Kullback–Leibler loss. The simulation results are summarized in Tables 1 and 2.

Table 1. *Simulation study. Averaged true and estimated values of generalized Kullback–Leibler loss for the original and tuned Platt methods and our new method based on 100 simulations with estimated standard errors in parentheses. Tuning is performed as in § 3.2. Values quoted are the true generalized Kullback–Leibler loss, GKL and the cross entropy, CRE, error over the test set.*

Classifier	Platt		Tuned Platt			New method		
	GKL	CRE	GKL	CRE	Improv.	GKL	CRE	Improv.
<i>Example 1</i>								
SVM.G	0.581 (0.0028)	0.548 (0.0027)	0.569 (0.0015)	0.547 (0.0015)	2.06%	0.566 (0.0014)	0.540 (0.0013)	2.58%
SVM.L	0.587 (0.0014)	0.553 (0.0014)	0.585 (0.0013)	0.551 (0.0013)	0.34%	0.570 (0.0015)	0.547 (0.0017)	2.90%
ψ - G	0.582 (0.0031)	0.551 (0.0029)	0.569 (0.0015)	0.549 (0.0015)	2.23%	0.562 (0.0015)	0.539 (0.0014)	3.44%
ψ - L	0.586 (0.0014)	0.553 (0.0013)	0.584 (0.0013)	0.550 (0.0013)	0.34%	0.561 (0.0015)	0.536 (0.0018)	4.27%
<i>Example 2</i>								
SVM.G	0.158 (0.0016)	0.154 (0.0014)	0.153 (0.0013)	0.150 (0.0013)	3.16%	0.153 (0.0010)	0.148 (0.0010)	3.16%
SVM.L	0.172 (0.0010)	0.173 (0.0010)	0.171 (0.0009)	0.171 (0.0009)	0.58%	0.160 (0.0009)	0.158 (0.0009)	6.80%
ψ - G	0.167 (0.0024)	0.163 (0.0026)	0.157 (0.0015)	0.154 (0.0015)	5.99%	0.153 (0.0018)	0.151 (0.0019)	8.38%
ψ - L	0.175 (0.0018)	0.174 (0.0019)	0.171 (0.0010)	0.170 (0.0010)	2.29%	0.159 (0.0014)	0.157 (0.0015)	9.14%

Improv, improvement relative to the original Platt method; SVM.L and SVM.G, support vector machines with linear and Gaussian kernels respectively; ψ - L and ψ - G, ψ -learning with linear and Gaussian kernels respectively.

Table 2. *Simulation study. Averaged values of generalized Kullback–Leibler losses and prediction error rates for penalized logistic regression, nearest neighbour method and our new method with ψ -learning based on 66 simulation replications with estimated standard errors in parentheses. Tuning is performed as in § 3.2 and Wang & Shen (2006). Values quoted are the generalized Kullback–Leibler loss, GKL, and the prediction error rate, TE, over the test set.*

	GKL for probability estimation			TE for classification		
	NN	PLR	New	NN	PLR	New
Example 1	0.582 (0.0014)	0.579 (0.0021)	0.552 (0.0010)	0.232 (0.0015)	0.258 (0.0053)	0.217 (0.0021)
Example 2	–	0.138 (0.0024)	0.149 (0.0013)	0.089 (0.0020)	0.075 (0.0018)	0.069 (0.0014)

PLR, penalized logistic regression; NN, method of nearest neighbour; New, new method.

Our method outperforms the original and tuned forms of Platt’s method, in all the examples with the linear and Gaussian kernels. The amount of improvement of our method over the original Platt method ranges from 2.58% to 9.14%. In addition, our method outperforms the nearest neighbour method in probability estimation as well as in classification, and it outperforms penalized logistic regression in classification but yields comparable performance in probability estimation. This says that a method such as support vector machines or ψ -learning that targets classification is

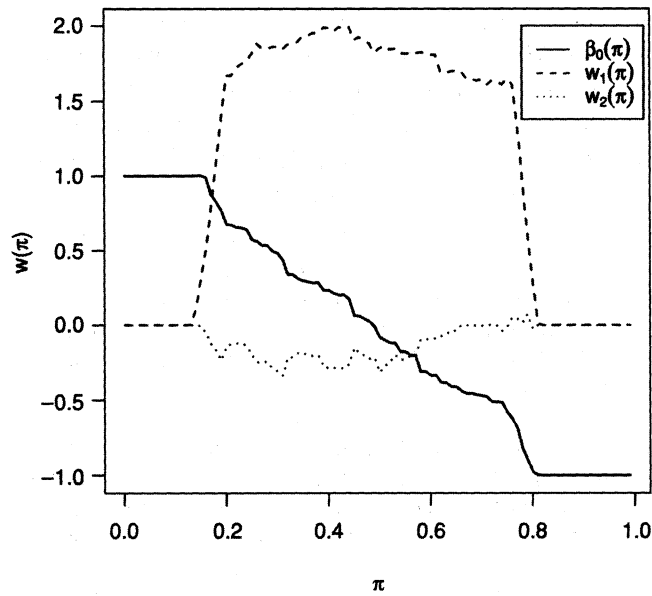


Fig. 1. The solution paths of $\beta_0(\pi)$, $w_1(\pi)$ and $w_2(\pi)$ as functions of π in Example 1, where $\hat{f}_\pi(x) = \beta_0(\pi) + x_{i1}w_1(\pi) + x_{i2}w_2(\pi)$ is the minimizer of (3) with a linear kernel.

able to achieve the performance of penalized logistic regression, which is designed for probability estimation. Also, it seems that the value of generalized Kullback–Leibler loss is reasonably well estimated by the cross entropy error.

Finally, Fig. 1 illustrates the piecewise-linear solution paths of the coefficients of the linearly weighted support vector machines in Example 1. Interestingly, there appear to be two roughly flat regions of the coefficients for π in $[0, 0.2]$ and $[0.8, 1]$. The corresponding solutions $\hat{f}_\pi(x)$ are approximately 1 and -1 for π in $[0, 0.2]$ and $[0.8, 1]$, respectively. This is because the true conditional probability in Example 1 is either 0.2 or 0.8.

5.3. Benchmark datasets

We now examine four benchmark examples, called Liver, Mushroom, Ionosphere and Diabetes and available from the University of California at Irvine repository of machine learning databases at <http://www.ics.uci.edu/~mllearn/MLRepository.html>. In each example, we randomly choose 100 cases for training and the remainder for testing.

For each pair of training and testing sets, tuning is conducted over λ for each method, using the same grid over the interval $[10^{-3}, 10^3]$ as in the simulated examples. For the Gaussian kernel case, σ is set to be the median distance between the positive and negative classes. Moreover, the original Platt method and the tuned Platt method are computed to illustrate that Platt's original proposal can be further enhanced by tuning.

The value of generalized Kullback–Leibler loss as estimated by the cross entropy error, averaged over 100 simulation replicates, is used for evaluation.

As suggested by Table 3, our method outperforms the Platt method in all cases except in the Ionosphere example. The improvement of our method over the original Platt method ranges from 3.49% to 48.2%. On average, the improvement is more substantial for the Gaussian kernel case than for the linear case. The performance of our method with ψ -learning appears to be slightly better than that of support vector machines. This indicates that the better classification perfor-

Table 3. *Benchmark datasets. Averaged true and estimated values of generalized Kullback–Leibler loss for the original and tuned Platt methods and our new method based on 100 simulation with estimated standard errors in parentheses. Tuning is performed as in § 3.2. Values quoted are the true generalized Kullback–Leibler loss, GKL, and the cross entropy error, CRE, over the test set.*

Classifier	Platt	Tuned Platt	Improv.	New	Improv.
<i>Mushroom</i>					
SVM.G	0.305(0.0035)	0.243(0.0038)	20.3%	0.234(0.0031)	23.3%
SVM.L	0.325(0.0066)	0.296(0.0054)	8.92%	0.223(0.0062)	31.4%
ψ . G	0.297(0.0038)	0.242(0.0029)	18.5%	0.232(0.0034)	21.9%
ψ . L	0.315(0.0072)	0.281(0.0052)	10.8%	0.215(0.0050)	31.7%
<i>Liver</i>					
SVM.G	0.724(0.0053)	0.655(0.0028)	9.53%	0.648(0.0025)	10.5%
SVM.L	0.663(0.0051)	0.647(0.0031)	2.41%	0.635(0.0021)	4.22%
ψ . G	0.690(0.0021)	0.650(0.0019)	5.80%	0.643(0.0017)	6.81%
ψ . L	0.672(0.0026)	0.665(0.0024)	1.04%	0.628(0.0019)	6.55%
<i>Diabetes</i>					
SVM.G	0.587(0.0053)	0.542(0.0038)	7.67%	0.536(0.0020)	8.69%
SVM.L	0.545(0.0026)	0.526(0.0024)	3.49%	0.526(0.0027)	3.49%
ψ . G	0.591(0.0057)	0.546(0.0041)	7.61%	0.536(0.0021)	9.31%
ψ . L	0.573(0.0028)	0.528(0.0029)	7.85%	0.528(0.0029)	7.85%
<i>Ionosphere</i>					
SVM.G	0.430(0.0062)	0.242(0.0052)	43.7%	0.250(0.0040)	41.9%
SVM.L	0.526(0.0102)	0.383(0.0045)	27.2%	0.384(0.0039)	27.0%
ψ . G	0.469(0.0067)	0.239(0.0046)	49.0%	0.243(0.0048)	48.2%
ψ . L	0.466(0.0089)	0.383(0.0045)	17.8%	0.382(0.0039)	18.0%

Improv, improvement relative to the original Platt method; SVM.L and SVM.G, support vector machines with linear and Gaussian kernels respectively; ψ . L and ψ .G, ψ -learning with linear and Gaussian kernels respectively.

Table 4. *Benchmark datasets. Averaged values of generalized Kullback–Leibler losses and prediction error rates for penalized logistic regression, the nearest neighbour method and our new method with ψ -learning based on nondegenerate simulation replications with estimated standard errors in parentheses. Tuning is performed as in § 3.2 and Wang & Shen (2006). Values quoted are the generalized Kullback–Leibler loss, GKL, and the prediction error rate, TE, over the test set.*

	CRE for probability estimation			TE for classification		
	NN	PLR	New	NN	PLR	New
Mushroom	—	—	0.215(0.0050)	0.482(0.0008)	0.085(0.0034)	0.065(0.0021)
Liver	—	0.665(0.0049)	0.628(0.0019)	0.578(0.0017)	0.335(0.0032)	0.316(0.0034)
Diabetes	—	0.553(0.0052)	0.528(0.0029)	0.348(0.0007)	0.252(0.0017)	0.232(0.0021)
Ionosphere	—	—	0.243(0.0048)	0.642(0.0014)	0.195(0.0033)	0.083(0.0024)

PLR, penalized logistic regression; NN, nearest neighbour method; New, new method.

mance in these examples translates into better estimation of the class probability. Furthermore, the tuned Platt method yields uniformly better performance than the original Platt method, with improvement ranging from 1.04% to 49.0%. Table 4 shows that our method with ψ -learning outperforms both the nearest neighbour method and penalized logistic regression in classification and probability estimation in these examples.

Table 5. *Leukaemia dataset. Estimated value of generalized Kullback–Leibler losses for the original and tuned Platt methods and our new method with $m = 19$ based on a testing set. Tuning is performed as in § 3.2.*

Classifier	Platt	Tuned Platt	Improv.	New	Improv.
SVM.L	0.343	0.343	0.00%	0.241	29.7%
ψ -L	0.171	0.171	0.00%	0.133	22.2%

Improv, improvement relative to the original Platt method;
SVM.L, support vector machine with linear kernel; ψ -L, ψ -learning with linear kernel.

6. AN APPLICATION TO MICROARRAY DATA

This section applies the proposed method to DNA microarray data concerning diagnosis of leukaemia (Golub et al., 1999) available at <http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi>. This dataset consists of 72 patients with 7129 genes expressed for each patient. Through patients' gene expressions, two types of acute leukaemia are discriminated, acute myeloid leukaemia and acute lymphoblastic leukaemia.

Clearly, the number of genes greatly exceeds the sample size, which is typical for microarray data. As a result, conventional methods cannot handle data of this type without the removal of some 'irrelevant' genes before discrimination; see Golub et al. (1999) for a prescreening analysis, and Guyon et al. (2002) and Guyon & Elisseeff (2003) for feature selection.

For discrimination, we apply linear support vector machines and ψ -learning to all 7129 genes for three reasons. First, support vector machines and ψ -learning are capable of processing such data efficiently because of the use of dual forms (Vapnik, 1998; Liu et al., 2005), whereas a conventional method cannot do so. Secondly, prescreening does not take into account the joint behaviour of genes. Thirdly, linear classification appears adequate here (Guyon et al., 2002).

To crossvalidate the performance of classification and probability estimation, we split the dataset into a training set of 38 patients and a test set of 34 patients. For the training and test sets, 11 and 27 patients, and 14 and 20 patients suffered acute myeloid leukaemia and acute lymphoblastic leukaemia, respectively. The accuracy of classification is measured by the test error, and the accuracy of probability estimation is measured by the estimated value of generalized Kullback–Leibler loss. Note that we do not include penalized logistic regression and the nearest neighbour method in this example since both of them yield degenerate estimates in this high-dimensional example.

For this dataset, support vector machines misclassify two samples, which is in contrast to one misclassified sample for ψ -learning. To see the strength of prediction, we compute the class probability at each observed input value in the test set. For support vector machines and ψ -learning, an estimated class probability of having acute myeloid leukaemia for each patient is near 0.975, except for patients 60 and 66, who are wrongly classified by support vector machines with estimated probabilities of 0.025 for them, and patient 66, who is wrongly classified by ψ -learning with an estimated probability of 0.025. This indicates strong confidence of the cancer discrimination.

We now examine the overall performance of our method and the Platt method in support vector machines and ψ -learning. As indicated in Table 5, our proposed method yields a more accurate class probability estimate than the Platt method in terms of the estimated generalized Kullback–Leibler loss. The amount of improvement of our method over the original Platt method

is 29.7% for support vector machines and 22.2% for ψ -learning. Moreover, ψ -learning yields better performance than support vector machines in all cases.

In summary, our method appears to perform well in this ‘high dimension but low sample size’ situation. In contrast, the Platt method deteriorates substantially, partly because the link function (1) breaks down.

7. ASYMPTOTIC THEORY

In the literature, fast convergence rates have been derived under various conditions for ψ -learning (Shen et al., 2003) and for support vector machines; see Steinwart & Scovel (2007), an unpublished manuscript by G. Blanchard, O. Bousquet and P. Massart and an unpublished University of Leiden technical report by B. Tarigan and S. van de Geer. However, asymptotic results about probability estimation for margin classification remain unavailable.

This section develops a novel theory for the proposed probability estimator \hat{p} as measured by the L_1 -norm $\|\hat{p} - p\|_1 = E|\hat{p}(X) - p(X)|$, in terms of the tuning parameter λ , the complexity of \mathcal{F} and (m, n) , where \mathcal{F} is the class of candidate functions and is allowed to depend on n . Here the generalized Kullback–Leibler loss is not considered because it suffers from the difficulty of degeneracy when $\hat{p} = 0$ or 1, thus requiring stronger assumptions.

Let $e_V(f, \tilde{f}_\pi) = E\{V(f, Z) - V(\tilde{f}_\pi, Z)\}$ with $V(f, z) = S(y)L\{yf(x)\}$ a weighted margin loss defined in (3). The following assumptions are made.

Assumption 1. (Approximation error.) For some positive sequence such that $s_n \rightarrow 0$ as $n \rightarrow \infty$, there exists $f_\pi^* \in \mathcal{F}$ such that $e_V(f_\pi^*, \tilde{f}_\pi) \leq s_n$.

Assumption 1 is analogous to Assumption A in Shen et al. (2003), and ensures that the Bayes rule \tilde{f}_π is well approximated by \mathcal{F} .

Define a truncated V by $V^T(f, z) = V(f, z)$ if $V(f, z) \leq T$ and $V^T(f, z) = T$ otherwise for any $f \in \mathcal{F}$ and some truncation constant T such that $\max\{V(\tilde{f}_\pi, z), V(f_\pi^*, z)\} \leq T$ almost surely, and $e_{VT}(f, \tilde{f}_\pi) = E\{V^T(f, Z) - V(\tilde{f}_\pi, Z)\}$.

Assumption 2. (Conversion formula.) There exist constants $0 \leq \alpha < \infty$, $0 \leq \beta \leq 1$, $a_1 > 0$ and $a_2 > 0$ such that, for any sufficiently small $\delta > 0$,

$$\sup_{\{f \in \mathcal{F}: e_{VT}(f, \tilde{f}_\pi) \leq \delta\}} \|\text{sign}(f) - \text{sign}(\tilde{f}_\pi)\|_1 \leq a_1 \delta^\alpha, \quad (21)$$

$$\sup_{\{f \in \mathcal{F}: e_{VT}(f, \tilde{f}_\pi) \leq \delta\}} \text{var}\{V^T(f, Z) - V(\tilde{f}_\pi, Z)\} \leq a_2 \delta^\beta. \quad (22)$$

Assumption 2 describes local smoothness of $\|\text{sign}(f) - \text{sign}(\tilde{f}_\pi)\|_1$ and $\text{var}\{V^T(f, Z) - V(\tilde{f}_\pi, Z)\}$ within a neighbourhood of f_π . The exponents α and β depend on the joint distribution of (X, Y) . The mean–variance relationship here is implied by Tsybakov’s assumption (Tsybakov, 2004), and thus is weaker (Shen & Wang, 2006). A similar assumption has been used in Shen & Wong (1994) in quantifying the rates of convergence for function estimation.

Next, we define the L_2 -metric entropy with bracketing that measures the cardinality of \mathcal{F} . Given any $\epsilon > 0$, define $\{(f_m^l, f_m^u)\}_{m=1}^M$ to be an ϵ -bracketing function set of \mathcal{F} if, for any $f \in \mathcal{F}$, there exists an m such that $f_m^l \leq f \leq f_m^u$ and $\|f_m^l - f_m^u\|_2 \leq \epsilon$ for $m = 1, \dots, M$. Then the L_2 -metric entropy with bracketing $H_B(\epsilon, \mathcal{F})$ is defined as the logarithm of the cardinality

of the smallest ϵ -bracketing function set of \mathcal{F} . Let $\mathcal{F}^V(k) = \{V^T(f, z) - V(f_\pi^*, z) : f \in \mathcal{F}(k)\}$, $\mathcal{F}(k) = \{f \in \mathcal{F} : J(f) \leq k\}$, $J(f) = \frac{1}{2}\|f\|_K^2$ and $J_\pi^* = \max\{J(f_\pi^*), 1\}$.

Assumption 3. (Metric entropy.) For some constants $a_i > 0$, $i = 3, \dots, 5$, and $\epsilon_n > 0$,

$$\sup_{k \geq 2} \phi(\epsilon_n, k) \leq a_5 n^{1/2}, \quad (23)$$

where $\phi(\epsilon, k) = \int_{a_4 L}^{a_3^{1/2} L^{B/2}} H_B^{1/2}\{w, \mathcal{F}^V(k)\} dw / L$, and $L = L(\epsilon, \lambda, k) = \min\{\epsilon^2 + \lambda(k/2 - 1)J_\pi^*, 1\}$.

THEOREM 3. Under Assumptions 1–3, for the estimator \hat{p} obtained from Algorithm 1, there exists a constant $a_6 > 0$ such that

$$\Pr \left\{ \|\hat{p} - p\|_1 \geq \frac{1}{2m} + \frac{1}{2} a_1 (m+1) \delta_n^{2\alpha} \right\} \leq 3.5 \exp\{-a_6 n (\lambda J_\pi^*)^{2-\beta}\},$$

provided that $\lambda^{-1} \geq 4\delta_n^{-2} J_\pi^*$, where $\delta_n^2 = \min\{\max\{\epsilon_n^2, s_n\}, 1\}$.

COROLLARY 1. Under the assumptions in Theorem 3,

$$\|\hat{p} - p\|_1 = O_p \left\{ \frac{1}{m} + a_1 (m+1) \delta_n^{2\alpha} \right\}, \quad E\|\hat{p} - p\|_1 = O \left\{ \frac{1}{m} + a_1 (m+1) \delta_n^{2\alpha} \right\},$$

provided that $n(\lambda J_\pi^*)^{2-\beta}$ is bounded away from 0.

Theorem 3 and Corollary 1 provide probability and risk bounds for $\|\hat{p} - p\|_1$. They also suggest the ideal m to be of order $O(\delta_n^{-\alpha})$, yielding the fast rate of $O(\delta_n^\alpha)$ for $E\|\hat{p} - p\|_1$.

To illustrate the phenomenon mentioned in §1, consider nonlinear classification by ψ -learning with the Gaussian kernel in Example 1. There $X = (X_1, X_2)$ is sampled from the uniform distribution over the unit disk $\{(X_1, X_2) : X_1^2 + X_2^2 \leq 1\}$, and $\Pr(Y = 1|X) = 0.8$ if $X_1 \geq 0$ and 0.2 otherwise. In this example, the candidate function class \mathcal{F} is $\{f : f(x) = \sum_{i=1}^n \alpha_i K(x_i, x) + b\}$ with Gaussian kernel $K(s, t) = \exp(-\|s - t\|^2 / \sigma^2)$.

To apply Corollary 1, we verify Assumptions 1–3. For Assumption 1, note that \mathcal{F} is rich for sufficiently large n in that, for any continuous function f , there exists a $\tilde{f} \in \mathcal{F}$ such that $\|f - \tilde{f}\|_\infty \leq \epsilon_n^2$ (Steinwart, 2001). This implies that there exists a function $\tilde{f}_\pi \in \mathcal{F}$ such that $\|f_\pi - \tilde{f}_\pi\|_\infty \leq \epsilon_n^2$. Choose $f_\pi^* = \tilde{f}_\pi \in \mathcal{F}$. Then $\|\text{sign}(f_\pi^*) - \text{sign}(f_\pi)\|_1 = 2\Pr(\text{sign}(\tilde{f}_\pi) \neq \text{sign}(f_\pi)) \leq 2\Pr(|f_\pi| \leq \epsilon_n^2) \leq 4\epsilon_n^2$, where ϵ_n is defined below. By construction of f_π^* , there exists a constant $c_1 > 0$ such that $e_V(f_\pi^*, \tilde{f}_\pi) \leq E|V(f_\pi^*, Z) - V(\tilde{f}_\pi, Z)| \leq c_1 \epsilon_n^2$.

For (21) in Assumption 2, we have that $|f_\pi| = |p(x) - \pi| \geq \min\{|\pi - 0.2|, |\pi - 0.8|\} > \eta$ and

$$\begin{aligned} e_\pi(f, \tilde{f}_\pi) &= E\{l(f, Z) - l(\tilde{f}_\pi, Z)\} = E|f_\pi| |\text{sign}(\tilde{f}_\pi) - \text{sign}(f)| \geq \eta E|\text{sign}(\tilde{f}_\pi) \\ &\quad - \text{sign}(f)| I(|f_\pi| \geq \eta) = \eta E|\text{sign}(f_\pi) - \text{sign}(f)| \end{aligned}$$

with $l(f, z) = S(y)[1 - \text{sign}\{yf(x)\}]$ for a sufficiently small constant $0 < \eta < \min\{|\pi - 0.2|, |\pi - 0.8|\}$. Thus, $E|\text{sign}(f) - \text{sign}(\tilde{f}_\pi)| \leq \eta^{-1} e_\pi(f, \tilde{f}_\pi) \leq \eta^{-1} e_{VT}(f, \tilde{f}_\pi)$, implying (21) with $\alpha = 1$. For (22) in Assumption 2, by the triangle inequality, $\text{var}\{V^T(f, Z) - V(f_\pi^*, Z)\} \leq TE|V^T(f, Z) - V(\tilde{f}_\pi, Z)| \leq T(\Lambda_1 + \Lambda_2)$, where

$$\begin{aligned} \Lambda_1 &= E|l(f, Z) - V(\tilde{f}_\pi, Z)| = E|S(Y)| |\text{sign}(f) - \text{sign}(\tilde{f}_\pi)| \leq \|\text{sign}(f) \\ &\quad - \text{sign}(\tilde{f}_\pi)\|_1 \leq \eta^{-1} e_{VT}(f, \tilde{f}_\pi), \end{aligned}$$

and

$$\begin{aligned}\Lambda_2 &= E\{V^T(f, Z) - l(f, Z)\} = E\{V^T(f, Z) - V(\bar{f}_\pi, Z)\} \\ &\quad + E\{l(\bar{f}_\pi, Z) - l(f, Z)\} \leq 2e_{VT}(f, \bar{f}_\pi).\end{aligned}$$

Therefore (22) holds with $\beta = 1$.

By Lemma A1, we have that $H_B\{\epsilon, \mathcal{F}^V(k)\} \leq O[\{\log(k/\epsilon)\}^3]$ for any k . Furthermore, let $\phi_1(\epsilon, k) = a_3\{\log(1/L^{1/2})\}^{3/2}/L^{1/2}$ with $L = L(\epsilon, \lambda, k)$. Solving (23) yields $\epsilon_n = \{n^{-1}(\log n)^3\}^{1/2}$ when $C_2/J_0 \sim \delta_n^{-2}n^{-1} \sim (\log n)^{-3}$.

By Corollary 1, $E\|\hat{p} - p\|_1 = O\{m^{-1} + a_1(m+1)n^{-1}(\log n)^3\}$. This also implies that $E\|\hat{p} - p\|_1 = O\{n^{-1/2}(\log n)^{3/2}\}$ with a choice of $m = n^{1/2}(\log n)^{3/2}$.

In summary, a fast rate $n^{-1/2}(\log n)^{3/2}$ is realized by our proposed estimator \hat{p} , whereas the classification accuracy of ψ -learning as measured by the generalization error in this example is of order $n^{-1}(\log n)^3$ (Liu & Shen, 2006). This confirms our discussion in § 1.

ACKNOWLEDGEMENT

This research is supported in part by grants from the U. S. National Science Foundation. The authors would like to thank Ji Zhu for helpful discussions, and thank the editor and three reviewers for comments and suggestions.

APPENDIX

Proofs

Proof of Lemma 1. We first show the case of $L(z) = (1 - z)_+$. The minimizer $\hat{f}(x)$ must take values in $[-1, +1]$, since $ES(Y)L\{Yf(X)\} \geq ES(Y)L\{Yf_{\pm 1}(X)\}$ with $f_{\pm 1} = f$ when $|f| \leq 1$ and $f_{\pm 1} = \text{sign}(f)$ otherwise. When $f(x)$ takes values in $[-1, +1]$, $\{1 - Yf(X)\}_+ = 1 - Yf(X)$. Thus minimization of (4) becomes $\min_f ES(Y)\{1 - Yf(X)\} = ES(Y) - \max_f E[E\{S(Y)Y|X\}f(X)]$. Furthermore, $E\{S(Y)Y|X\} = \text{pr}(Y = 1|X)(1 - \pi) - \{1 - \text{pr}(Y = 1|X)\}\pi = \text{pr}(Y = 1|X) - \pi$, yielding the minimizer of (4) to be $\text{sign}\{\text{pr}(Y = 1|X) - \pi\}$.

For the case of $L(z) = \psi(z)$, note that $\min_f ES(Y)[1 - Y\text{sign}\{f(X)\}]$ yields $\text{sign}\{\text{pr}(Y = 1|X) - \pi\}$ following the same argument as in case of $L(z) = (1 - z)_+$. The desired result follows because $ES(Y)\psi\{Yf(X)\} \geq ES(Y)[1 - Y\text{sign}\{f(X)\}]$ and $ES(Y)\psi[Y\text{sign}\{\text{pr}(Y = 1|X) - \pi\}] = ES(Y)[1 - Y\text{sign}\{\text{pr}(Y = 1|X) - \pi\}]$.

Proof of Theorem 1. It is easy to show that minimizing (7) with respect to ζ yields that $\zeta(\hat{p}, X^n) = E\{\text{GKL}^c(p, \hat{p})|X^n\} - E\{\text{EGKL}(\hat{p})|X^n\}$, which can be simplified to

$$\begin{aligned}E\left[-E\log\{1 - \hat{p}(X)\} - E\ p(X)\log\frac{\hat{p}(X)}{1 - \hat{p}(X)}\middle|X^n\right] &- E\left[-n^{-1}\sum_{i=1}^n\log\{1 - \hat{p}(X_i)\}\right. \\ &\quad \left.- n^{-1}\sum_{i=1}^n p(X_i)\log\frac{\hat{p}(X_i)}{1 - \hat{p}(X_i)} - n^{-1}\sum_{i=1}^n\left\{\frac{1}{2}(Y_i + 1) - p(X_i)\right\}\log\frac{\hat{p}(X_i)}{1 - \hat{p}(X_i)}\middle|X^n\right] \\ &= n^{-1}\sum_{i=1}^n\text{cov}\left\{\frac{1}{2}(Y_i + 1), \log\frac{\hat{p}(X_i)}{1 - \hat{p}(X_i)}\right\} + E\{\Delta(p, \hat{p}; X^n) - \bar{\Delta}(p, \hat{p})|X^n\},\end{aligned}$$

where $\bar{\Delta}(p, \hat{p})$ and $\Delta(p, \hat{p}; X^n)$ are as defined in §3.1.

Proof of Theorem 2. The proof is similar to that of Theorem 2 in Wang & Shen (2006), and thus is omitted.

Proof of Theorem 3. We first introduce some notation. Let $n^{-1} \sum_{i=1}^n \tilde{V}(f, Z_i)$ be the penalized cost function to be minimized with $\tilde{V}(f, z) = V(f, z) + \lambda J(f)$, and $\tilde{V}^T(f, z) = V^T(f, z) + \lambda J(f)$. We also define the scaled empirical process, $E_n\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\}$, as $n^{-1} \sum_{i=1}^n [\tilde{V}^T(f, Z_i) - \tilde{V}^T(f_\pi^*, Z_i) - E\{\tilde{V}^T(f, Z_i) - \tilde{V}^T(f_\pi^*, Z_i)\}] = E_n\{V^T(f, Z) - V^T(f_\pi^*, Z)\}$.

It follows from the definition of \hat{f}_π and $V^T \leq V$ that

$$\begin{aligned} \text{pr}\{e_{VT}(\hat{f}_\pi, \tilde{f}_\pi) \geq \delta_n^2\} &\leq \text{pr}^* \left[\sup_{e_{VT}(f, \tilde{f}_\pi) \geq \delta_n^2} n^{-1} \sum_{i=1}^n \{\tilde{V}(f_\pi^*, Z_i) - \tilde{V}(f, Z_i)\} \geq 0 \right] \\ &\leq \text{pr}^* \left[\sup_{e_{VT}(f, \tilde{f}_\pi) \geq \delta_n^2} n^{-1} \sum_{i=1}^n \{\tilde{V}(f_\pi^*, Z_i) - \tilde{V}^T(f, Z_i)\} \geq 0 \right] = \Gamma, \end{aligned}$$

where pr^* denotes the outer probability measure.

To bound Γ , we partition $\{f \in \mathcal{F} : e_{VT}(f, \tilde{f}_\pi) \geq \delta_n^2\}$ into a union of $A_{s,t}$, with $A_{s,t} = \{f \in \mathcal{F} : 2^{s-1}\delta_n^2 \leq e_{VT}(f, \tilde{f}_\pi) < 2^s\delta_n^2, 2^{t-1}J_\pi^* \leq J(f) < 2^tJ_\pi^*\}$ and $A_{s,0} = \{f \in \mathcal{F} : 2^{s-1}\delta_n^2 \leq e_{VT}(f, \tilde{f}_\pi) < 2^s\delta_n^2, J(f) < J_\pi^*\}$, for $s, t = 1, 2, \dots$. Then it suffices to bound the corresponding probability over each $A_{s,t}$. Towards this end, we need to bound the first and second moments of $\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)$ over $f \in A_{s,t}$. Without loss of generality, assume that $4s_n < \epsilon_n^2 < 1$, $J(f_\pi^*) \geq 1$, and thus $J_\pi^* = \max\{J(f_\pi^*), 1\} = J(f_\pi^*)$.

For the first moment, using the assumption that $\lambda J(f_\pi^*) \leq \delta_n^2/2$, we have that

$$\begin{aligned} \inf_{A_{s,t}} E\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\} &\geq M(s, t) = 2^{s-1}\delta_n^2 + \lambda(2^{t-1} - 1)J(f_\pi^*), \\ \inf_{A_{s,0}} E\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\} &\geq (2^{s-1} - 3/4)\delta_n^2 \geq M(s, 0) = 2^{s-3}\delta_n^2, \end{aligned}$$

for any $s, t = 1, 2, \dots$

For the second moment, it follows from Assumption 2 and the fact that

$$\text{var} V^T(f, Z) - V(f_\pi^*, Z) \leq 2 \left[\text{var}\{V^T(f, Z) - V(\tilde{f}_\pi, Z)\} + \text{var}\{V^T(f_\pi^*, Z) - V(\tilde{f}_\pi, Z)\} \right]$$

that

$$\sup_{A_{s,t}} \text{var}\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\} = \sup_{A_{s,t}} \text{var}\{V^T(f, Z) - V(f_\pi^*, Z)\} \leq a_3 M(s, t)^\beta = v^2(s, t),$$

for any $s, t = 1, 2, \dots$ and some constant $a_3 > 0$.

Now we obtain $\Gamma \leq \Gamma_1 + \Gamma_2$, with $\Gamma_1 = \sum_{s,t=1}^\infty \text{pr}^*[\sup_{A_{s,t}} E_n\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\} \geq M(s, t)]$ and $\Gamma_2 = \sum_{s=1}^\infty \text{pr}^*[\sup_{A_{s,0}} E_n\{\tilde{V}^T(f, Z) - \tilde{V}^T(f_\pi^*, Z)\} \geq M(s, 0)]$. Next we bound Γ_1 and Γ_2 separately using Theorem 3 of Shen & Wong (1994). For Γ_1 , we verify the conditions (4.5)–(4.7) there. Using the fact that $\int_{v(s,t)}^{aM(s,t)} H_B^{1/2}\{w, \mathcal{F}^V(2^w)\} dw / M(s, t)$ is nonincreasing in s and $M(s, t)$ for $s = 1, \dots$, we have

$$\int_{v(s,t)}^{aM(s,t)} H_B^{1/2}\{w, \mathcal{F}^V(2^t)\} dw / M(s, t) \leq \int_{aM(1,t)}^{a_3M(1,t)^{\beta/2}} H_B^{1/2}\{w, \mathcal{F}^V(2^t)\} dw / M(1, t) \leq \phi(\epsilon_n, 2^t),$$

with $a = 2a_4\epsilon$. Then Assumption 3 implies (4.5)–(4.7) in Shen & Wong (1994) with $\epsilon = 1/2$, the choices of $M(s, t)$ and $v(s, t)$ and some constants $a_i > 0$ for $i = 3, 4$. It follows from Theorem 3 of Shen & Wong

(1994) that, for some constant $0 < \xi < 1$,

$$\begin{aligned}\Gamma_1 &\leq \sum_{s,t=1}^{\infty} 3 \exp \left[-\frac{(1-\xi)nM^2(s,t)}{2\{4v^2(s,t) + M(s,t)T/3\}} \right] \leq \sum_{s,t=1}^{\infty} 3 \exp [-a_6 n \{M(s,t)\}^{2-\beta}] \\ &\leq \sum_{s,t=1}^{\infty} 3 \exp [-a_6 n \{2^{s-1} \delta_n^2 + \lambda(2^{t-1} - 1)J_n^*\}^{2-\beta}] \\ &\leq 3 \exp \{-a_6 n (\lambda J_n^*)^{2-\beta}\} / [1 - \exp\{-a_6 n (\lambda J_n^*)^{2-\beta}\}]^2.\end{aligned}$$

Similarly, $\Gamma_2 \leq 3 \exp\{-a_6 n (\lambda J_n^*)^{2-\beta}\} / [1 - \exp\{-a_6 n (\lambda J_n^*)^{2-\beta}\}]^2$. Combining the bounds for Γ_i , $i = 1, 2$, we have $\Gamma^{1/2} \leq (5/2 + \Gamma^{1/2}) \exp\{-a_6 n (\lambda J_n^*)^{2-\beta}\}$. Then $\Gamma = \text{pr}\{e_{VT}(\hat{f}_\pi, \bar{f}_\pi) \geq \delta_n^2\} \leq 3.5 \exp\{-a_6 n (\lambda J_n^*)^{2-\beta}\}$ because $\Gamma^{1/2} \leq 1$. It follows from (21) that

$$\text{pr}\{\|\text{sign}(\hat{f}_\pi) - \text{sign}(\bar{f}_\pi)\|_1 \geq a_1 \delta_n^{2\alpha}\} \leq 3.5 \exp\{-a_6 n (\lambda J_n^*)^{2-\beta}\}. \quad (\text{A1})$$

Next we establish a connection between $\|\hat{p} - p\|_1$ and $\|\text{sign}(\hat{f}_\pi) - \text{sign}(\bar{f}_\pi)\|_1$. For $j = 1, \dots, m+1$, let $\Delta_j = \{x : \text{sign}(\hat{f}_{\pi_j}(x)) \neq \text{sign}(\bar{f}_{\pi_j}(x))\}$. Then $\|\text{sign}(\hat{f}_{\pi_j}) - \text{sign}(\bar{f}_{\pi_j})\|_1 = 2EI(\Delta_j)$. It can be shown that $\{2EI(\Delta_j) \leq a_1 \delta_n^{2\alpha}, j = 1, \dots, m+1\} \subset \{EI(\bigcup_{j=1}^{m+1} \Delta_j) \leq \frac{1}{2}(m+1)a_1 \delta_n^{2\alpha}\}$. Moreover, $\{x : |\hat{p}(x) - p(x)| \geq (2m)^{-1}\}$ implies that $\{x : |\pi^* - \pi_*| > m^{-1} \text{ or } p(x) \notin [\pi_*, \pi^*]\}$ and $\{x : |\pi^* - \pi_*| > m^{-1} \text{ or } p(x) \notin [\pi_*, \pi^*]\}$ occurs only if there is some $1 \leq j \leq m+1$ such that $\text{sign}(\hat{f}_{\pi_j}(x)) \neq \text{sign}(\bar{f}_{\pi_j}(x))$. To be specific, we have $\bigcup_{j=1}^{m+1} \Delta_j \supset \{x : |\pi^* - \pi_*| > m^{-1} \text{ or } p(x) \notin [\pi_*, \pi^*]\} \supset \{x : |\hat{p}(x) - p(x)| \geq (2m)^{-1}\} = \mathbb{B}$. Therefore,

$$\begin{aligned}&\left\{EI\left(\bigcup_{j=1}^{m+1} \Delta_j\right) \leq \frac{1}{2}(m+1)a_1 \delta_n^{2\alpha}\right\} \\ &\subset \left\{EI(\mathbb{B}) \leq \frac{1}{2}(m+1)a_1 \delta_n^{2\alpha}\right\} \subset \left\{\|\hat{p} - p\|_1 \leq \frac{1}{2m} + \frac{1}{2}(m+1)a_1 \delta_n^{2\alpha}\right\}.\end{aligned}$$

Finally, $\text{pr}\{\|\hat{p} - p\|_1 \geq \frac{1}{2m} + \frac{1}{2}(m+1)a_1 \delta_n^{2\alpha}\} \leq \text{pr}\{\text{there exists } j : \|\text{sign}(\hat{f}_{\pi_j}) - \text{sign}(\bar{f}_{\pi_j})\|_1 \geq a_1 \delta_n^{2\alpha}\}$. The desired result follows from (A1).

LEMMA A1. Under the assumptions in §7, we have $H_B\{\epsilon, \mathcal{F}^V(k)\} \leq O[\{\log(k/\epsilon)\}^3]$.

Proof. From the result of Example 4 in Zhou (2002), $H_\infty\{\epsilon, \mathcal{F}(k)\} \leq O[\{\log(k/\epsilon)\}^3]$ under the L_∞ -metric, $\|f\|_\infty = \sup_{x \in \mathcal{R}^2} |f(x)|$. Note that, for functions f_l and f_u , $\|V^T(f_l, \cdot) - V^T(f_u, \cdot)\|_2 \leq \|f_l - f_u\|_2 \leq \|f_l - f_u\|_\infty$, implying that $H_B\{\epsilon, \mathcal{F}^V(k)\} \leq H_\infty\{\epsilon, \mathcal{F}(k)\}$. The desired result then follows.

REFERENCES

- BARTLETT, P. & TEWARI, A. (2007). Sparseness vs estimating conditional probabilities: some asymptotic results. *J. Mach. Learn. Res.* **8**, 775–90.
- BREIMAN, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *J. Am. Statist. Assoc.* **87**, 738–54.
- BREIMAN, L. & SPECTOR, P. (1992). Submodel selection and evaluation in regression—the X-Random case. *Int. Rev. Statist.* **3**, 291–319.
- CORTES, C. & VAPNIK, V. (1995). Support vector networks. *Mach. Learn.* **20**, 273–97.
- EFRON, B. (2004). The estimation of prediction error: covariance penalties and cross-validation (with Discussion). *J. Am. Statist. Assoc.* **99**, 619–42.
- GOLUB, T., SLONIM, D., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J., COLLIER, H., LOH, M., DOWNING, J. & CALIGIURI, M. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–6.
- GUYON, I. & ELISSEFF, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–82.

- GUYON, I., WESTON, J. & VAPNIK, V. (2002). Gene selection for cancer classification using support vector machine. *Mach. Learn.* **46**, 389–422.
- HASTIE, T., ROSSET, S., TIBSHIRANI, R. & HZ, J. (2004). The entire regularization path for the support vector machine. *J. Mach. Learn. Res.* **5**, 1391–415.
- JAAKKOLA, T., DIEKHANS, M. & HAUSSLER, D. (1999). Using the Fisher kernel method to detect remote protein homologies. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, Ed. T. Lengauer, R. Schneider, P. Bork., D. Brutlag, J. Glasgow, H. Mewes and R. Zimmer, pp. 149–58. Heidelberg, Germany: AAAI.
- KIMELDORF, G. & WAHBA, G. (1971). Some results on Tchebycheffian spline functions, *J. Math. Anal. Applic.* **33**, 82–95.
- LIN, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining Know. Disc.* **6**, 259–75.
- LIN, Y., LEE, Y. & WAHBA, G. (2002). Support vector machines for classification in nonstandard situations. *Mach. Learn.* **46**, 191–202.
- LIU, S., SHEN, X. & WONG, W. (2005). Computational development of ψ -learning. In *Proc. 2005 SIAM Int. Conf. Data Mining*, Ed. H. Kargupta, J. Srivastava, C. Kamath and A. Goodman, pp. 1–12. Philadelphia: SIAM.
- LIU, Y. & SHEN, X. (2006). Multicategory ψ -learning. *J. Am. Statist. Assoc.* **101**, 500–9.
- LIU, Y., SHEN, X. & DOSS, H. (2005). Multicategory ψ -learning and support vector machine: computational tools. *J. Comp. Graph. Statist.* **14**, 219–36.
- PLATT, J.C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, Ed. A. Smola, P. Bartlett, B. Scholkopf and D. Schuurmans, pp. 61–74. Cambridge, MA: MIT Press.
- SHEN, X. & HUANG, H-C. (2006). Optimal model assessment, selection and combination. *J. Am. Statist. Assoc.* **101**, 554–68.
- SHEN, X., TSENG, G.C., HANG, & X. WONG, W.H. (2003). On ψ -learning. *J. Am. Statist. Assoc.* **98**, 724–34.
- SHEN, X. & WANG, L. (2006). Discussion of ‘Local Rademacher complexities and oracle inequalities in risk minimization’ by V. Koltchinskii. *Ann. Statist.* **34**, 2677–80.
- SHEN, X. & WONG, W.H. (1994). Convergence rate of sieve estimates. *Ann. Statist.* **22**, 580–615.
- STEINWART, I. (2001). On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.* **2**, 67–93.
- STEINWART, I. (2003). Sparseness of support vector machines. *J. Mach. Learn. Res.* **4**, 1071–105.
- STEINWART, I. & SCOVEL, C. (2007). Fast rates for support vector machines using Gaussian kernels. *Ann. Statist.* **35**, 575–607.
- TSYBAKOV, A. (2004). Optimal aggregation of classifiers in statistical learning. *Ann. Statist.* **32**, 135–66.
- VAPNIK, V. (1998). *Statistical Learning Theory*. New York: Wiley.
- WAHBA, G. (1990). *Spline Models for Observational Data*. Philadelphia: SIAM.
- WANG, J. & SHEN, X. (2006). Estimation of generalisation error: random and fixed inputs. *Statist. Sinica* **16**, 569–88.
- ZHOU, D.X. (2002). The covering number in learning theory. *J. Complexity* **18**, 739–67.

[Received April 2006, Revised May 2007]