Probability estimation for large margin classifiers

By JUNHUI WANG, XIAOTONG SHEN

School of Statistics, University of Minnesota, Minneapolis, MN 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, NC 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 to estimate the class probability through sequential classifications, by utilising 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 further reduce its computational cost. Theoretical and numerical analyses indicate that the method is highly competitive against alternatives, especially when the dimension of 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 practise seems to undermine the fact that classification is generally easier than regression, because the former is an interval instead of a point estimation problem. This is evident from recent success in large margin classification such as support vector machines (SVM; 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 requirement of estimating the class probability yielding classification. In classification, knowledge about the class probability itself may be of significant scientific interest, indicating the strength or confidence of 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 generalisation ability and computational advantage.

In binary classification, a decision function f is estimated based on a training sample $Z_i = (X_i, Y_i)$; $i = 1, \ldots, n$, 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 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 (2004) show that replacing the large margin loss with some differentiable loss leads to conditional probability estimation asyptotically. Platt (1999) assumes a sigmoid link function between the conditional distribution $p(x) = P(Y = 1|X = x)$ and large margin classifier f in the form of

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

with parameters A and B estimated by minimising the cross entropy error. Despite its empirical success, statistical properties of estimated $p(x)$ through classification and (1) have not yet been investigated. There is no solid evidence that a link function as 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 SVM is $\text{sign}(p(x) - 1/2)$, which implies that the classifier f might only concern about whether $p(x)$ greater than $1/2$ or not.

In this article, we take a novel approach to 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 $\text{sign}(p(x) - 1/2) > 0$ (Lin, 2002). On this basis, we design a sequence of weighted classifications, corresponding to a refined partition of $[0, 1]$, to locate which subinterval contains $p(x)$ for any fixed x. This approach is illustrated in high dimension but low sample size data, typical of microarray experiments.

The proposed method is implemented for SVM and ψ-learning. To eliminate dependency of the method on a tuning parameter, we propose a method of model selection through the concept of 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 SVM 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 of $n^{-1/2}(\log n)^{3/2}$, 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.

The rest of this article is organised as follows. Section 2 briefly discusses large margin classification as well as the proposed method. Section 3 constructs our loss estimator for tuning, together with a technique of data perturbation. Section 4 develops the solution path algorithm for our method via SVM. Section 5 compares the proposed method against one

top performer in the literature, followed by an application to leukaemia data in Section 6. Section 7 provides a theoretical justification. Section 8 contains a discussion. The appendix is devoted to technical proofs.

2. Estimation

This section reviews large margin classification, and introduces the proposed method involving sequential weighted classification.

2·1. *Large margin classifiers*

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

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

where $J(f)$ is a regularisation term for penalising model complexity, $\lambda > 0$ is the degree of penalisation, and $L(z)$ is a margin loss that is a function of functional margin $yf(x)$; for instance, $L(z) = (1-z)_+$ is the hinge loss for SVM, 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, c.f., Shen et al. (2003). A margin loss $L(z)$ is said to be large margin if $L(z)$ is nonincreasing in z, penalising small margin values. In linear classification, f is linear; in kernel classification, f uses a kernel representation $\frac{1}{n\lambda}\sum_{i=1}^{n}\theta_i y_i K(x_i, x) + \beta_0$ with $K(\cdot,\cdot)$ a kernel. In the kernel case, $J(f) = \frac{1}{2n^2\lambda^2}\sum_{i=1}^{n}\sum_{i'=1}^{n}\theta_i\theta_{i'}y_i y_{i'}K(x_i, x_{i'})$, and $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS; Wahba, 1990) induced by $K(\cdot,\cdot)$. The weighted version of (2) is

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

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 $(\pi, 1-\pi)$ is the known cost for the negative and positive classes

with $0 \leq \pi \leq 1$; c.f. Lin et al. (2002) for a discussion. Minimising (3) with respect to $f \in \mathcal{F}$ yields $\hat{f}_\pi(x)$, and thus classifier $\text{sign}(\hat{f}_\pi(x))$.

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

$$ES(Y)L(Yf(X)) = E\left((1-\pi)I(Y=1)L(Yf(X)) + \pi I(Y=-1)L(Yf(X))\right), \qquad (4)$$

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

Lemma 1 *With $L(z) = (1-z)_+$ or $\psi(z)$, minimising (4) with respect to f yields the Bayes rule $\bar{f}_\pi(x) = \text{sign}(f_\pi(x))$ with $f_\pi(x) = p(x) - \pi$. Moreover, $\bar{f}_\pi(x)$ is nonincreasing in π.*

2·2. *Estimation*

Our proposed method is designed for estimating $p(x)$ at any x with x not necessarily being one of the observed values. The method proceeds as follows. First, construct a uniform partition of $[0, 1]$ with the two end points 0 and 1 included, that is, $0 = \pi_1 < \pi_2 < \cdots < \pi_m < \pi_{m+1} = 1$ 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 utilise the monotonicity property (Lemma 1) of $\text{sign}(f_\pi(x))$ to rapidly compute $p(x)$ at one x value. However, the monotonicity property may not hold empirically, in addition that 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; $j = 1, \cdots, m+1$, to identify the interval capturing $p(x)$. This is achieved by checking if $\text{sign}(\hat{p}(x) - \pi_j) > 0$ through $\text{sign}(\hat{f}_{\pi_j}(x))$, for $j = 1, \cdots, m+1$. 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 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 $0 \le \pi_*, \pi^* \le 1$. Then the proposed estimate $\widehat{p}(x)$ is defined as $\frac{1}{2}(\pi_* + \pi^*)$.

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

Algorithm 1:

Step 1. Initialise $\pi_j = (j-1)/m$; $j = 1, \cdots, m+1$.

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

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

Step 4. Sort $\text{sign}(\hat{f}_{\pi_j}(x))$; $j = 1, \cdots, m+1$, 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 SVM (WSVM; Lin et al., 2002) and weighted ψ-learning. To train a WSVM, any software with quadratic programming (QP) routine can be employed. To train a weighted ψ-learning, we follow the technique of Liu, Shen & Wong (2005b), where difference convex algorithm (DCA) is used to solve the non-convex optimisation problem through sequential QP. The idea of DCA is to decompose a non-convex objective function into a difference of two convex functions, and solve the non-convex problem by sequential convex problems. According to Liu et al. (2005b), DCA is appropriate for the ψ-loss based on its encouraging numerical performance as well as its fast convergence speed.

Furthermore, a precision parameter m needs to be prespecified in Algorithm 1, balancing the trade-off between the precision of $\hat{p}$ and the number of weighted classifiers to be trained. Evidently, a large m value yields better precision but increases computational cost. In implementation, m is recommended to be $\lfloor n^{1/2} \rfloor$, the largest integer no greater than $n^{1/2}$. This choice seems to be satisfactory as suggested by our simulation. Of course, a data-driven choice of m can be derived at an expense of increased computational cost, obtained by minimising an estimated loss with regard to m, as to be discussed next.

3. Estimating generalised KL loss and tuning

This section develops a model selection method for estimating p, which uses an estimated generalised Kullback-Leibler (GKL) loss. This permits a performance evaluation of any estimation method.

Estimation of the KL loss has been investigated in Shen & Huang (2006), but that of the GKL 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.

3·1. *Estimation of GKL loss*

The overall performance of $\hat{p}$ in estimating p is evaluated by its closeness to p in terms of the GKL loss

$$GKL(p,\hat{p}) \;=\; E\left(p(X)\log\frac{p(X)}{\hat{p}(X)}+(1-p(X))\log\frac{1-p(X)}{1-\hat{p}(X)}\right), \tag{5}$$

where the expectation is taken with respect to randomness in X, which differs from the KL loss in that (5) is averaged over random X having the same distribution as X_i; $i = 1,\cdots,n$. The corresponding comparative KL loss, after omitting $\hat{p}$-unrelated term in (5), is $GKL^c(p,\hat{p}) = -E(p(X)\log(\hat{p}(X))+(1-p(X))\log(1-\hat{p}(X)))$. Using the fact that $E(\frac{1}{2}(Y+1)|X) = p(X)$, the empirical version of $GKL^c(p,\hat{p})$ is

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

which measures the goodness-of-fit of $\hat{p}$. To penalise overfitting due to $EGKL(\hat{p})$, $GKL^c(p,\hat{p})$ is estimated by choosing the optimal estimator from a class of candidate estimators of the form $EGKL(\hat{p})+\zeta(\hat{p},X^n)$, where $\zeta(\hat{p},X^n)\geq 0$ is an $X^n = \{X_i\}_{i=1}^n$-dependent penalty to be optimally determined by minimising

$$E\left(GKL^c(p,\hat{p})-(EGKL(\hat{p})+\zeta(\hat{p},X^n))\right)^2. \tag{7}$$

Theorem 1 *The optimal* $\zeta_o(\hat{p}, X^n)$ *that minimises (7) over* $\zeta(\hat{p}, X^n)$ *is*

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

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

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

Therefore, we propose our estimated $GKL^c(p, \hat{p})$ to be

$$\widehat{GKL}^c(p, \hat{p}) = EGKL^c(\hat{p}) + n^{-1} \sum_{i=1}^{n} \widehat{\mathrm{cov}}((Y_i + 1)/2, \phi(\hat{p}(X_i))|X^n) + \widehat{D}_n(\hat{p}, X^n), \tag{9}$$

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

Our data perturbation method proceeds as follows. First perturb X_i; $i = 1, \cdots, n$, via its empirical distribution $\hat{F}$, followed by flipping the corresponding label Y_i with a certain probability given the perturbed X_i. This generates perturbations for assessing the accuracy of probability estimation. More precisely, for $i = 1, \cdots, n$, let

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

with $\tilde{X}_i$ sampled from $\hat{F}$, $0 \le \tau \le 1$ is the size of perturbation, and $\tilde{Y}_i \sim \text{Bin}(1, \hat{p}(X_i^*))$ 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

$$\widehat{\text{cov}}((Y_i+1)/2, \phi(\hat{p}(X_i))|X^n) = \frac{1}{K(Y_i, \hat{p}(X_i^*))}\text{cov}^*((Y_i^*+1)/2, \phi(\hat{p}^*(X_i^*))|X^{*n}), \tag{11}$$

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

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

With (11) and (12), we obtain $\widehat{GKL}^c(p, \hat{p})$ in (9), which can be computed by Monte Carlo (MC) approximations. First, generate D perturbed samples $X^{*ln} = \{X_i^{*l}\}_{i=1}^n$ according to (10); $l = 1, \cdots, D$. Second, for each sample $\{X_i^{*l}\}_{i=1}^n$, generate D perturbed samples $\{Y_i^{*lm}\}_{i=1}^n$ according to (10); $m = 1, \cdots, D$. Furthermore, for $l, m = 1, \cdots, D$; $i = 1, \cdots, n$, compute $\widehat{\text{cov}}^*((Y_i^*+1)/2, \phi(\hat{p}^*(X_i^*))|X^{*n}) = \frac{1}{D^2-1}\sum_{l,m=1}^D \phi(\hat{p}^{*lm}(X_i^{*l}))(Y_i^{*lm} - \overline{Y}_i^* + 1)/2$, where $\hat{p}^{*lm}$ is trained by applying the proposed probability estimation method to $\{X_i^{*l}, Y_i^{*lm}\}_{i=1}^n$, and $\overline{Y}_i^* = \frac{1}{D^2}\sum_{l,m=1}^D Y_i^{*lm}$. Now (11) and (12) are approximated by the corresponding MC approximation, i.e.,

$$\widehat{\text{cov}}((Y_i+1)/2, \phi(\hat{p}(X_i))|X^n) \approx \frac{1}{2(D^2-1)}\sum_{l,m=1}^D \frac{1}{K(Y_i, \hat{p}(X_i^{*l}))}\phi(\hat{p}^{*lm}(X_i^{*l}))((Y_i^{*lm} - \overline{Y}_i^*) + 1); \tag{13}$$

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

By the law of large numbers, (13) and (14) converges to (11) and (12) respectively, as $D \to \infty$. In practise, we recommend D to be at least $\lfloor n^{1/2} \rfloor$ to ensure the precision of MC

approximation. Plugging (13) and (14) into (9), we obtain the final estimate of $GKL^c(p, \hat{p})$.

3·2. *Tuning*

The performance of $\hat{p}_\lambda$ depends on λ, and hence that optimal selection of λ becomes important, where $\hat{p}$ is written as $\hat{p}_\lambda$ to indicate its dependency on λ. Minimisation of (9) over the range of $\lambda > 0$ yields the optimal λ, denoted as $\hat{\lambda}$.

Assumptions (C.1)-(C.3) are made concerning optimality of selecting λ through(9), which are analogous to those in Wang & Shen (2006) but different in that consistency is not required here.

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

(C.2): (Loss and risk) $\lim_{m\to\infty} \lim_{n\to\infty} \sup_\lambda |GKL(p, \hat{p}_\lambda)/E(GKL(p, \hat{p}_\lambda)) - 1| = 0$ in probability.

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

Theorem 2 *Under Conditions C.1-C.3, for $\hat{\lambda}$ the minimiser of (9), we have*

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

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

4. Solution path for SVM

This section develops a solution path algorithm for SVM to facilitate computation. One direct benefit of the solution path algorithm is that given an initial solution for the path algorithm, the m SVM classifiers with different weights in Step 2 there can be trained at essentially the same cost of training one SVM 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 RKHS representation theorem of Kimeldorf & Wahba (1971). As to be seen, $\theta(\pi) = (\theta_1(\pi), \cdots, \theta_n(\pi))^T$ 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 π otherwise. On this basis, we derive an efficient algorithm for computing an *exact solution path* of $\theta(\pi)$, thus of $\hat{f}_\pi(x)$, for $0 < \pi < 1$.

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

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

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

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

with the Karush-Kuhn-Tucker conditions:

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

From (16), $0 \le \alpha_i \le (1-\pi)$ if $y_i = 1$ and $0 \le \alpha_i \le \pi$ if $y_i = -1$, because $\gamma_i \ge 0$; $i = 1, \cdots, n$, implying that (1) $y_i f(x_i) > 1 \Rightarrow \xi_i = 0, \alpha_i = 0$; (2) $y_i f(x_i) < 1 \Rightarrow \xi_i \ne 0, \gamma_i = 0$, $\alpha_i = S(y_i)$; 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 $0 \le \pi \le 1$.

Following an idea of Hastie et al. (2004), we define three sets to track the solution path in π based on preceding relationships: (1) $\mathcal{E} = \{i : y_i f(x_i) = 1, 0 \le \theta_i \le S(y_i)\}$ (Elbow); (2) $\mathcal{L} = \{i : y_i f(x_i) < 1, \theta_i = S(y_i)\}$ (Left of the elbow); (3) $\mathcal{R} = \{i : y_i f(x_i) > 1, \theta_i = 0\}$ (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 initialised to compute the solution of $\theta(\pi)$ through Algorithm 1, then the value of π increases toward 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 changes from $S(y_i)$ towards 0, implying 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 categorised as: (1) An element from $\mathcal{L}$ has just entered into $\mathcal{E}$ with θ_i to be initially $S(y_i)$; (2) An element from $\mathcal{R}$ has just entered into $\mathcal{E}$ with θ_i to be initially 0; (3) element(s) from $\mathcal{E}$ has/have just left $\mathcal{E}$ to join either $\mathcal{L}$ or $\mathcal{R}$.

In what follows, we use the subscript ℓ to index the preceding sets as well as parameter and function values $(\theta_i^\ell, \beta_0^\ell, \pi^\ell)$ and f^ℓ, immediately after the ℓth event has occurred. For convenience, write $\beta_{0,\lambda} = n\lambda \cdot \beta_0$ and $\beta_{0,\lambda}^\ell = n\lambda \cdot \beta_0^\ell$. Note that $f(x) = \frac{1}{n\lambda}\left(\beta_{0,\lambda} + \sum_{i=1}^n \theta_i y_i K(x, x_i)\right)$. Then for $\pi^\ell < \pi < \pi^{\ell+1}$,

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

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

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

This implies $\nu_0 y_k + \sum_{i\in\mathcal{E}^\ell} \nu_i y_k y_i K(x_k, x_i) = (\pi - \pi^\ell) y_k \sum_{i\in\mathcal{L}^\ell} K(x_k, x_i), \quad \forall k \in \mathcal{E}^\ell$ with $\nu_i = (\theta_i - \theta_i^\ell)$ and $\nu_0 = (\beta_{0,\lambda} - \beta_{0,\lambda}^\ell)$. By (16), $\sum_{i\in\mathcal{E}^\ell} \nu_i y_i = (\pi - \pi^\ell) n_{\mathcal{L}}^\ell$. Thus we solve a system of $n_{\mathcal{E}}^\ell + 1$ linear equations involving the $n_{\mathcal{E}}^\ell + 1$ unknown variables ν_i and ν_0.

Let $\mathbf{K}_y^\ell$ be an $n_{\mathcal{E}}^\ell \times n_{\mathcal{E}}^\ell$ matrix with its entries $y_k y_i K(x_k, x_i)$ for $i, k \in \mathcal{E}^\ell$, let $y_{\mathcal{E}}^\ell$ be the vector with components $y_k, k \in \mathcal{E}^\ell$, let ν be a vector with components ν_i for $i \in \mathcal{E}^\ell$, and 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

$$\nu_0 y_{\mathcal{E}}^\ell + \mathbf{K}_y^\ell \nu = (\pi - \pi^\ell) K_y^\ell; \quad \nu' y_{\mathcal{E}}^\ell = (\pi - \pi^\ell) n_{\mathcal{L}}^\ell. \tag{18}$$

To further simplify (18), we let $\mathbf{K}_y^* = \begin{pmatrix} 0 & (y_{\mathcal{E}}^\ell)' \\ y_{\mathcal{E}}^\ell & \mathbf{K}_y^\ell \end{pmatrix}$, $\nu^* = \begin{pmatrix} \nu_0 \\ \nu \end{pmatrix}$, and $K_y^* = \begin{pmatrix} n_{\mathcal{L}}^\ell \\ K_y^\ell \end{pmatrix}$, then equations in (18) can be combined to be $\mathbf{K}_y^* \nu^* = (\pi - \pi^\ell) K_y^*$. Then if $\mathbf{K}_y^*$ has full rank, define $b^* = (\mathbf{K}_y^*)^{-1} K_y^*$ to yield

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

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

$$f(x) = f^\ell(x) + (\pi - \pi^\ell) h^\ell(x), \tag{20}$$

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

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 $\pi = \pi^\ell + \frac{1 - y_k f^\ell(x_k)}{y_k h^\ell(x_k)}, \quad \forall k \in \mathcal{R}^\ell \cup \mathcal{L}^\ell$. Termination occurs when π has become sufficiently close to 1.

5. Numerical results

This section examines effectiveness of the proposed method, and compares it to some popular competitors: the Platt method (Platt, 1999), penalised logistic regression (PLR) and nearest neighbour (NN), in simulated and benchmark examples, 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 for classification, its accuracy with respect to the generalisation error is examined as well.

In simulated examples, the GKL loss over a test set is used for evaluating probability estimation when the true p is known. In the benchmark examples, the cross entropy error (CRE) over a test set is used when p is unknown, defined as $CRE(\hat{p}) = -\frac{1}{\#\{\text{test set}\}} \sum_{\text{test set}} \big(\frac{y_i+1}{2} \log(\hat{p}(x_i)) + \frac{1-y_i}{2} \log(1-\hat{p}(x_i))\big)$, where $\#\{\text{A}\}$ is the cardinality of set A.

5·1. *Simulation*

The proposed method is examined for SVM and ψ-learning in the linear and Gaussian kernel cases, where SVM is trained using the svm routine in package e1071 of R2.1.1 and ψ-learning is trained as in Liu et al. (2005b) based on DCA. For PLR, training is performed through routine StepPlr in R2.1.1. For NN, it is implemented in R2.1.1.

For our method, the Platt method and PLR, we seek the optimal λ by minimising (9) through a grid search over interval $[10^{-3}, 10^3]$ with ten equally-spaced points in each interval $(10^j, 10^{j+1}]$; $j = -3, \cdots, 2$. For Gaussian kernel SVM, σ is set to be the median distance between the positive and negative classes (Jaakkola et al., 1999), because λ plays a similar role as σ^2 and it is easier to optimise with respect to λ with σ^2 fixed. For NN, the best performance from $\{4NN, 9NN, 16NN, 25NN\}$ with different neighbourhood sizes is used.

Example 1: Data $\{(X_{i1}, X_{i2}, Y_i); i = 1, \cdots, 1000\}$ are generated as follows. First, $\{(X_{i1}, X_{i2});\ i = 1, \cdots, 1000\}$ are sampled from the uniform distribution over a unit disk $\{(X_1, X_2) : X_1^2 + X_2^2 \leq 1\}$. Second, $Y_i = 1$ if $X_{i1} \geq 0$ and -1 otherwise; $i = 1, \cdots, n$. Third, randomly choose 20% of the sample and flip their labels to generate the nonseparable case. This yields the first simulated example, where 100 and 900 randomly selected instances are

for training and testing, respectively.

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

With regard to probability estimation, the true GKL loss in (5) is averaged over 100 simulation replications. Unfortunately, however, for PLR and NN, the value of GKL could be infinity when the estimated probability becomes exactly 0 or 1. To overcome this difficulty, we average over only 66 nondegenerate replications for PLR, and leave a blank for NN in Example 2 in presence of nondegenerate replications. With regard to classification, the test error (TE) is used to measure the performance, averaged over nondegenerate replications. Finally, the result for CRE is given to see if CRE well estimates the GKL loss. The simulation results are summarised in Tables 1 and 2.

Tables 1 and 2 about here

Our method outperforms the Platt method, original and tuned, 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 NN in probability estimation as well as classification, and outperforms PLR in classification but yields a comparable performance in probability estimation. This says that a method targets to classification such as SVM or ψ-learning is able to achieve the performance of PLR that is designed for probability estimation. Also, it seems that the GKL loss is reasonably well estimated by CRE.

Figure 1 about here

Finally, we use Figure 1 to illustrate the piecewise linear solution paths of the coefficients of linear weighted SVM in Example 1. Interestingly, there appears to have 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 due to the fact that the true conditional probability in Example 1 is either 0.2 or 0.8.

5·2. *Benchmarks*

We now examine four benchmark examples: Liver, Mushroom, Ionosphere and Diabetes (the UCI Machine Learning Repository, Blake & Merz, 1998). In each example, we randomly choose 100 instances for training and the remaining for testing.

For each pair of training and testing sets, tuning is conducted over λ for each method. Particularly, the optimal tuning parameter λ is estimated using the same grids over interval $[10^{-3}, 10^3]$ as in the simulated examples. For the Gaussian kernel case, σ^2 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 estimated GKL loss through CRE, averaged over 100 simulation replicates, is used for evaluation.

Tables 3 and 4 about here

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%. The amount of improvement over the Platt method varies over different types of classifiers. On average, the improvement is more substantial for the Gaussian kernel case than the linear case. The performance of our method with ψ-learning appears to be slightly better than that of SVM. This indicates that the better classification performance 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 NN and PLR in classification and probability estimation in these examples.

6. An application to microarray data

This section applies the proposed method to DNA microarray data – leukaemia (Golub et al., 1999) available at `http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi`. This leukaemia data consists of 72 patients with 7,129 genes expressed for each patient. Through patients' gene expressions, two types of acute leukaemia are discriminated, acute myeloid leukaemia (AML) and acute lymphoblastic leukaemia (ALL).

One characteristic of this dataset is that the number of genes greatly exceeds the sample size, which is typical for microarray data. As a result, a conventional method is incapable of handling data of this type without removal of some "irrelevant" genes before discrimination. See Golub et al. (1999) for a prescreening analysis, and Guyon et al. (2002), and Guyon & Elisseff (2003) for feature selection.

For discrimination, we apply linear SVM and ψ-learning to all 7129 genes for three reasons. First, SVM and ψ-learning are capable of processing this type of data efficiently because of the usage of dual forms (Vapnik, 1998; Liu et al., 2005), whereas a conventional method fails to do so. Second, prescreening does not take the joint behaviour of genes into account. Third, linear classification appears adequate here, c.f., Guyon et al. (2002).

To cross-validate 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 AML and ALL, respectively. The accuracy of classification is measured by the test error, where the accuracy of probability estimation is measured by the estimated GKL loss. Note that we don't include PLR and NN in this example since both of them yield degenerate estimates in this high dimensional example.

Table 5 about here

For this dataset, SVM misclassifies 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 SVM and ψ-learning, an estimated class probability of AML for each patient is near 0.975, except for patients 60 and 66 who are wrongly classified by SVM with estimated probabilities of 0.025 for them and for 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 SVM 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 GKL loss. The amount of improvement of our method over the original Platt method is 29.7% for SVM and is 22.2% for ψ-learning. Moreover, ψ-learning yields better performance than SVM in all cases.

In summary, our method appears to perform well in this "high dimension but low sample size" situation. On the contrary, the Platte method deteriorates substantially, partly because the link function (1) breaks down.

7. Asymptotic theory

In literature, fast convergence rates have been derived under various conditions for SVM (Blanchard et al., 2004; Tarigan & Van de Geer, 2004; Steinwart & Scovel, 2005) and ψ-learning (Shen et al., 2003). However, asymptotic results about probability estimation for margin classification remain unavailable.

This section develops a novel theory for the proposed probability estimate $\hat{p}$ as measured by the L_1-norm $\|\hat{p} - p\|_1 = E|\hat{p}(X) - p(X)|$, in terms of tuning parameter λ, 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 GKL loss is not considered because it suffers from the difficulty of degeneracy

when $\hat{p} = 0$ or 1, thus requiring stronger assumptions.

7·1. *Theory*

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

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

Assumption A is analogous to Assumption A in Shen et al. (2003), which ensures that the Bayes rule $\bar{f}_\pi$ is well approximated by $\mathcal{F}$.

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

Assumption B. (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 $\delta > 0$,

$$\sup_{\{f \in \mathcal{F}:\ e_{V^T}(f, \bar{f}_\pi) \leq \delta\}} \| \operatorname{sign}(f) - \operatorname{sign}(\bar{f}_\pi) \|_1 \leq a_1 \delta^\alpha, \tag{21}$$

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

Assumption B describes local smoothness of $\| \operatorname{sign}(f) - \operatorname{sign}(\bar{f}_\pi) \|_1$ and $\operatorname{var}(V^T(f, Z) - V(\bar{f}_\pi, Z))$ 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 condition, thus is weaker, c.f., Shen & Wang (2006). A similar assumption has been used in Shen & Wong (1994) in quantifying the rates of convergence for classification.

Next, we define the L_2-metric entropy with bracketing that measures the cardinality of $\mathcal{F}$. Given any $\epsilon > 0$, denote $\{(f_m^l, f_m^u)\}_{m=1}^M$ as 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 $\|f_m^l - f_m^u\|_2 \leq \epsilon; m = 1, \cdots, M$. 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 $\mathcal{F}^V(k) = \{V^T(f,z) - V(f_\pi^*, z) : f \in \mathcal{F}(k)\}$, $\mathcal{F}(k) = \{f \in \mathcal{F} : J(f) \le k\}$, $J(f) = \frac{1}{2}\|f\|_K^2$ and $J_\pi^* = \max(J(f_\pi^*), 1)$.

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

$$\sup_{k \ge 2} \phi(\epsilon_n, k) \le a_5 n^{1/2}, \tag{23}$$

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

Theorem 3 *Under Assumptions A-C, for the estimator $\hat{p}$ obtained from Algorithm 1, there exists a constant $a_6 > 0$ such that*

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

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

Corollary 1 *Under the assumptions in Theorem 3,*

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

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 $O(\delta_n^\alpha)$ for $E\|\hat{p} - p\|_1$.

7·2. *A nonlinear example*

To illustrate the phenomenon mentioned in the Introduction, 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 $\{(X_1, X_2) : X_1^2 + X_2^2 \le 1\}$, and

$P(Y=1|X)=0.8$ 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 A-C. For Assumption A, 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 $\|f - \tilde{f}\|_\infty \leq \epsilon_n^2$, c.f. 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 $\|\operatorname{sign}(f_\pi^*) - \operatorname{sign}(f_\pi)\|_1 = 2P(\operatorname{sign}(\tilde{f}_\pi) \neq \operatorname{sign}(f_\pi)) \leq 2P(|f_\pi| \leq \epsilon_n^2) \leq 4\epsilon_n^2$, where ϵ_n is defined below. By construction of f_π^*, there exists a constant $c_1 > 0$ such that $e_V(f_\pi^*, \bar{f}_\pi) \leq E|V(f_\pi^*, Z) - V(\bar{f}_\pi, Z)| \leq c_1\epsilon_n^2$.

For (21) in Assumption B, we have $|f_\pi| = |p(x) - \pi| \geq \min\{|\pi - 0.2|, |\pi - 0.8|\} > \eta$ and $e_\pi(f, \bar{f}_\pi) = E(l(f,Z) - l(\bar{f}_\pi, Z)) = E|f_\pi||\operatorname{sign}(\bar{f}_\pi) - \operatorname{sign}(f)| \geq \eta E|\operatorname{sign}(\bar{f}_\pi) - \operatorname{sign}(f)| I(|f_\pi| \geq \eta) = \eta E|\operatorname{sign}(f_\pi) - \operatorname{sign}(f)|$ with $l(f,z) = S(y)(1 - \operatorname{sign}(yf(x)))$ for a sufficiently small constant $0 < \eta < \min\{|\pi - 0.2|, |\pi - 0.8|\}$. Thus, $E|\operatorname{sign}(f) - \operatorname{sign}(\bar{f}_\pi)| \leq \eta^{-1} e_\pi(f, \bar{f}_\pi) \leq \eta^{-1} e_{V^T}(f, \bar{f}_\pi)$, implying (21) with $\alpha = 1$. For (22) in Assumption B, by the triangle inequality, $\operatorname{var}(V^T(f,Z) - V(f_\pi^*, Z)) \leq TE|V^T(f,Z) - V(\bar{f}_\pi, Z)| \leq T(\Lambda_1 + \Lambda_2)$, where $\Lambda_1 = E|l(f,Z) - V(\bar{f}_\pi, Z)| = E|S(Y)||\operatorname{sign}(f) - \operatorname{sign}(\bar{f}_\pi)| \leq \|\operatorname{sign}(f) - \operatorname{sign}(\bar{f}_\pi)\|_1 \leq \eta^{-1} e_{V^T}(f, \bar{f}_\pi)$, and $\Lambda_2 = E(V^T(f,Z) - l(f,Z)) = E(V^T(f,Z) - V(\bar{f}_\pi, Z)) + E(l(\bar{f}_\pi, Z) - l(f,Z)) \leq 2e_{V^T}(f, \bar{f}_\pi)$. Therefore (22) holds with $\beta = 1$.

By Lemma 2, we have $H_B(\epsilon, \mathcal{F}^V(k)) \leq O((\log(k/\epsilon))^3)$ 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 $\epsilon_n = (\frac{(\log n)^3}{n})^{1/2}$ when $C_2/J_0 \sim \delta_n^{-2} n^{-1} \sim (\log n)^{-3}$.

By Corollary 1, $E\|\hat{p} - p\|_1 = O\left(\frac{1}{m} + a_1(m+1)n^{-1}(\log n)^3\right)$. This also implies that $E\|\hat{p} - p\|_1 = O\left(n^{-1/2}(\log n)^{3/2}\right)$ 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 realised by our proposed estimator $\hat{p}$, whereas the classification accuracy of ψ-learning as measured by the generalisation error in this example is of order $n^{-1}(\log n)^3$ (Liu & Shen, 2006). This confirms our discussion in the Introduction.

8. Discussion

This article proposes a novel methodology for estimating the conditional class probability for margin classification. In contrast to existing methods assuming a link function between the class probability and the classification decision function, our proposed method estimates the probability through interval estimation of classification. The theoretical and numerical analyses demonstrate that the proposed method has a good rate of convergence and outperforms several other competitors in numerical examples. The generalisation of our proposed method to multicategory classification is under way, while the Platt's method has been generalised by Passerini et al. (2004).

Acknowledgement

This research is supported in part by NSF grants IIS-0328802, DMS-0604394 and DMS-0606577. The authors would like to thank Ji Zhu for helpful discussions, and thank the editor and three reviewers for comments and suggestions.

Appendix

Proof of Lemma 1: We first show the case of $L(z) = (1-z)_+$. The minimiser $\hat{f}(x)$ 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 $\text{sign}(f)$ otherwise. When $f(x)$ takes values in $[-1, +1]$, $(1 - Yf(X))_+ = 1 - Yf(X)$. Thus minimisation of (4) becomes $\min_f ES(Y)(1 - Yf(X)) = ES(Y) - \max_f E(E(S(Y)Y|X)f(X))$. Furthermore, $E(S(Y)Y|X) = P(Y = 1|X)(1-\pi) - (1-P(Y = 1|X))\pi = P(Y = 1|X) - \pi$, yielding the minimiser of (4) to be $\text{sign}(P(Y = 1|X) - \pi)$.

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

Proof of Theorem 1: It is easy to show that minimising (7) with respect to ζ yields that

$\zeta(\hat{p}, X^n) = E(GKL^c(p, \hat{p})|X^n) - E(EGKL(\hat{p})|X^n)$, which can be simplified to

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

where $\bar{\Delta}(p, \hat{p})$ and $\Delta(p, \hat{p}; X^n)$ are as defined in Section 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 notations to be used. Let $n^{-1}\sum_{i=1}^{n}\widetilde{V}(f, Z_i)$ be the penalised cost function to be minimised with $\widetilde{V}(f, z) = V(f, z) + \lambda J(f)$, and $\widetilde{V}^T(f, z) = V^T(f, z) + \lambda J(f)$. We also define the scaled empirical process, $E_n(\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z))$, as $n^{-1}\sum_{i=1}^{n}\left(\widetilde{V}^T(f, Z_i) - \widetilde{V}(f_\pi^*, Z_i) - E(\widetilde{V}^T(f, Z_i) - \widetilde{V}(f_\pi^*, Z_i)\right) = E_n(V^T(f, Z) - V(f_\pi^*, Z))$. It follows from the definition of $\hat{f}_\pi$ and $V^T \le V$ that

$$
\begin{aligned}
P(e_{V^T}(\hat{f}_\pi, \bar{f}_\pi) \ge \delta_n^2) &\le P^*\left(\sup_{e_{V^T}(f,\bar{f}_\pi)\ge\delta_n^2} n^{-1}\sum_{i=1}^{n}(\widetilde{V}(f_\pi^*, Z_i) - \widetilde{V}(f, Z_i)) \ge 0\right) \\
&\le P^*\left(\sup_{e_{V^T}(f,\bar{f}_\pi)\ge\delta_n^2} n^{-1}\sum_{i=1}^{n}(\widetilde{V}(f_\pi^*, Z_i) - \widetilde{V}^T(f, Z_i)) \ge 0\right) = \Gamma,
\end{aligned}
$$

where P^* denotes the outer probability measure.

To bound Γ, we partition $\{f \in \mathcal{F} : e_{V^T}(f, \bar{f}_\pi) \ge \delta_n^2\}$ into a union of $A_{s,t}$, with $A_{s,t} = \{f \in \mathcal{F} : 2^{s-1}\delta_n^2 \le e_{V^T}(f, \bar{f}_\pi) < 2^s\delta_n^2, 2^{t-1}J_\pi^* \le J(f) < 2^t J_\pi^*\}$ and $A_{s,0} = \{f \in \mathcal{F} : 2^{s-1}\delta_n^2 \le e_{V^T}(f, \bar{f}_\pi) < 2^s\delta_n^2, J(f) < J_\pi^*\}$, for $s, t = 1, 2, \cdots$. Then it suffices to bound the corresponding probability over each $A_{s,t}$. Toward this end, we need to bound the first and second moment of $\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z)$ 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 $\lambda J(f_\pi^*) \leq \frac{1}{2}\delta_n^2$, we have $\inf_{A_{s,t}} E(\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z)) \geq M(s,t) = 2^{s-1}\delta_n^2 + \lambda(2^{t-1}-1)J(f_\pi^*)$ and $\inf_{A_{s,0}} E(\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z)) \geq (2^{s-1} - 3/4)\delta_n^2 \geq M(s,0) = 2^{s-3}\delta_n^2$, for any $s, t = 1, 2, \cdots$.

Similarly, for the second moment, it follows from Assumption B and the fact that $\mathrm{var}(V^T(f, Z) - V(f_\pi^*, Z)) \leq 2\big(\mathrm{var}(V^T(f, Z) - V(\bar{f}_\pi, Z)) + \mathrm{var}(V^T(f_\pi^*, Z) - V(\bar{f}_\pi, Z))\big)$ that $\sup_{A_{s,t}} \mathrm{var}(\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z)) = \sup_{A_{s,t}} \mathrm{var}(V^T(f, Z) - V(f_\pi^*, Z)) \leq a_3 M(s,t)^\beta = v^2(s,t)$ for any $s, t = 1, 2, \cdots$ and some constant $a_3 > 0$.

Now we obtain $\Gamma \leq \Gamma_1 + \Gamma_2$, with $\Gamma_1 = \sum_{s,t=1}^\infty P^*(\sup_{A_{s,t}} E_n(\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z)) \geq M(s,t))$ and $\Gamma_2 = \sum_{s=1}^\infty P^*(\sup_{A_{s,0}} E_n(\widetilde{V}^T(f, Z) - \widetilde{V}(f_\pi^*, Z)) \geq M(s,0))$. 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 non-increasing in s and $M(s,t); s = 1, \cdots$, we have

$$\int_{v(s,t)}^{aM(s,t)} H_B^{1/2}(w, \mathcal{F}^V(2^t))dw/M(s,t) \leq \int_{aM(1,t)}^{a_3 M(1,t)^{\beta/2}} H_B^{1/2}(w, \mathcal{F}^V(2^t))dw/M(1,t) \leq \phi(\epsilon_n, 2^t),$$

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

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

Similarly, $\Gamma_2 \leq 3\exp(-a_6 n(\lambda J_\pi^*)^{2-\beta})/(1 - \exp(-a_6 n(\lambda J_\pi^*)^{2-\beta}))^2$. Combining the bounds for $\Gamma_i;\ i = 1, 2$, we have $\Gamma^{1/2} \leq (5/2 + \Gamma^{1/2})\exp(-a_6 n(\lambda J_\pi^*)^{2-\beta})$. Then $\Gamma = P(e_{V^T}(\hat{f}_\pi, \bar{f}_\pi) \geq$

$\delta_n^2) \leq 3.5\exp(-a_6 n(\lambda J_\pi^*)^{2-\beta})$ because $\Gamma^{1/2} \leq 1$. It follows from (21) that,

$$P(\|\operatorname{sign}(\hat{f}_\pi) - \operatorname{sign}(\bar{f}_\pi)\|_1 \geq a_1\delta_n^{2\alpha}) \leq 3.5\exp(-a_6 n(\lambda J_\pi^*)^{2-\beta}). \tag{24}$$

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

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

Finally, $P(\|\hat{p} - p\|_1 \geq \frac{1}{2m} + \frac{1}{2}(m+1)a_1\delta_n^{2\alpha}) \leq P(\exists j : \|\operatorname{sign}(\hat{f}_{\pi_j}) - \operatorname{sign}(\bar{f}_{\pi_j})\|_1 \geq a_1\delta_n^{2\alpha})$. The desired result follows from (24).

Lemma 2 *(Metric Entropy in Section 6.2.2) Under the assumptions in Section 6.2, we have* $H_B(\epsilon, \mathcal{F}^V(k)) \leq O((\log(k/\epsilon))^3)$.

Proof: Using the result of Example 4 in Zhou (2002), $H_\infty(\epsilon, \mathcal{F}(k)) \leq O(\log(k/\epsilon)^3)$ under the $L_\infty-$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 $H_B(\epsilon, \mathcal{F}^V(k)) \leq H_\infty(\epsilon, \mathcal{F}(k))$. The desired result then follows.

References

[1] Bartlett, P. and Tewari, A. (2004). Sparseness vs estimating conditional probabilities: Some asymptotic results. In *Proceedings of the 17th Annual Conference on Learning Theory*, **3120**, 564-578.

[2] Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. University of California, Irvine, Department of Information and Computer Science.

[3] Blanchard, G., Bousquet,O. & Massart, P. (2004) Statistical performance of support vector machines. Manuscript.

[4] Breiman, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *J. Ameri. Statist. Assoc.*, **87**, 738-754.

[5] Breiman, L. & Spector, P. (1992). Submodel selection and evaluation in regression – the X - Random case. *International Review Statist.*, **3**, 291 - 319.

[6] Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning*, **20**, 273-297.

[7] Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation (with discussion). *J. Ameri. Statist. Assoc.*, **99**, 619-632.

[8] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J. & Caligiuri, M. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531-536.

[9] Guyon, I. & Elisseff, A. (2003). An introduction to variable and feature selection. *J. Machine Learning Res.*, **3**, 1157-1182.

[10] Guyon, I., Weston, J. & Vapnik, V. (2002). Gene selection for cancer classification using support vector machine. *Machine Learning*, **46**, 389-422.

[11] Hastie, T., Rosset, S., Tibshirani, R. & Hz, J. (2004). The entire regularization path for the support vector machine. *J. Machine Learning Res.*, **5**, 1391-1415.

[12] Jaakkola, T., Diekhans, M. & Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. In *Proc. the Seventh International Conf. on Intelligent Systems for Molecular Biology*, 149-158.

[13] Kimeldorf, G. & Wahba, G. (1971). Some results on Tchebycheffian spline functions, *J. Math. Analysis and Applications*, **33**,82-95.

[14] Lin, Y. (2002). Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery*, **6**, 259-275.

[15] Lin, Y., Lee, Y. & Wahba, G. (2002). Support vector machines for classification in non-standard situations. *Machine Learning*, **46**, 191-202.

[16] Liu, S., Shen, X. & Wong, W. (2005). Computational development of ψ-learning. In *The SIAM 2005 International Data Mining Conf.*, 1-12.

[17] Liu, Y. & Shen, X. (2006). Multicategory ψ-learning. *J. Ameri. Statist. Assoc.*, **101**, 500-509.

[18] Liu, Y., Shen, X. & Doss, H. (2005). Multicategory ψ-learning and support vector machine: computational tools. *J. Computa. & Graphical Statist.*, **14**, 219-236.

[19] Passerini, A., Pontil, M. & Frasconi, P. (2004). New results on error correcting output codes of kernel machines. *IEEE Transaction on Neural Networks*, **15**, 45-54.

[20] Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans (Eds.), MIT Press, 61–74.

[21] Shen, X. & Huang, H-C. (2006). Optimal model assessment, selection and combination. *J. Ameri. Statist. Assoc.*, **101**, 554-568.

[22] Shen, X., Tseng, G.C., Hang, X. & Wong, W.H. (2003). On ψ-learning. *J. Ameri. Statist. Assoc.*, **98**, 724-734.

[23] Shen, X. & Wang, L. (2006). Discussion of 2004 IMS Medallion Lecture: "Local Rademacher complexities and oracle inequalities in risk minimization". *Ann. Statist.*, in press.

[24] Shen, X. & Wong, W.H. (1994). Convergence rate of sieve estimates. *Ann. Statist.*, **22**, 580-615.

[25] Steinwart, I (2001). On the influence of the kernel on the consistency of support vector machines. *J. Machine Learning Res.*, **2**, 67-93.

[26] Steinwart, I (2003). Sparseness of Support Vector Machines. *J. Machine Learning Res.*, **4**, 1071-1105.

[27] Steinwart, I. & Scovel, C. (2005). Fast rates for support vector machines using Gaussian kernels. Manuscript.

[28] Tarigan, B. & van de Geer, S. (2004). Adaptivity of support vector machines with L_1 penalty. Technical Report MI 2004-14, University of Leiden.

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

[30] Wahba, G. (1990). Spline models for observational data society for industrial and applied mathematics. Philadelphia.

[31] Wang, J. & Shen, X. (2006). Estimation of generalization error: random and fixed inputs. *Statistica Sinica*, **16**, 569-588.

[32] Zhou, D.X. (2002). The covering number in learning theory. *J. Complexity*, **18**, 739-767.

Table 1: Averaged true and estimated GKL losses of our method, the original and tuned Platt method in the simulated examples over 100 simulation. GKL and CRE denote the true GKL loss and the CRE over the test set, with tuning performed as in Section 3.2. Improv. denotes the percentage of improvements of our method and the tuned Platt method over the original Platt method.

Data	Classifier	Platt's		Tuned Platt's			Our		
		GKL	CRE	GKL	CRE	Improv.	GKL	CRE	Improv.
Example 1	SVM_G	.581	.548	.569	.547	2.06%	.566	.540	2.58%
		(.0028)	(.0027)	(.0015)	(.0015)		(.0014)	(.0013)	
	SVM_L	.587	.553	.585	.551	0.34%	.570	.547	2.90%
		(.0014)	(.0014)	(.0013)	(.0013)		(.0015)	(.0017)	
	ψ_G	.582	.551	.569	.549	2.23%	.562	.539	3.44%
		(.0031)	(.0029)	(.0015)	(.0015)		(.0015)	(.0014)	
	ψ_L	.586	.553	.584	.550	0.34%	.561	.536	4.27%
		(.0014)	(.0013)	(.0013)	(.0013)		(.0015)	(.0018)	
Example 2	SVM_G	.158	.154	.153	.150	3.16%	.153	.148	3.16%
		(.0016)	(.0014)	(.0013)	(.0013)		(.0010)	(.0010)	
	SVM_L	.172	.173	.171	.171	0.58%	.160	.158	6.80%
		(.0010)	(.0010)	(.0009)	(.0009)		(.0009)	(.0009)	
	ψ_G	.167	.163	.157	.154	5.99%	.153	.151	8.38%
		(.0024)	(.0026)	(.0015)	(.0015)		(.0018)	(.0019)	
	ψ_L	.175	.174	.171	.170	2.29%	.159	.157	9.14%
		(.0018)	(.0019)	(.0010)	(.0010)		(.0014)	(.0015)	

Table 2: Averaged GKL losses and prediction errors of our method with ψ-learning, penalised logistic regression and nearest neighbour in Examples 1 and 2 over 66 simulation replications. Here TE denotes the test error, with tuning performed as in Wang & Shen (2006), and PLR and NN represent the penalised logistic regression and the method of nearest neighbour respectively.

Method	GKL for probability estimation			TE for classification		
	NN	PLR	Our	NN	PLR	Our
Example 1	.582(.0014)	.579(.0021)	.552(.0010)	.232(.0015)	.258(.0053)	.217(.0021)
Example 2	-	.138(.0024)	.149(.0013)	.089(.0020)	.075(.0018)	.069(.0014)

Table 3: Averaged estimated GKL loss of our method, the original Platt method and the tuned Platt method in the benchmark examples over 100 simulation replicates, with tuning performed as in Section 3.2. Improv. denotes the percentage of improvement over the original Platt method.

Data	Classifier	Platt's	Tuned Platt's	Improv.	Our	Improv.
Mushroom	SVM_G	.305(.0035)	.243(.0038)	20.3%	.234(.0031)	23.3%
	SVM_L	.325(.0066)	.296(.0054)	8.92%	.223(.0062)	31.4%
	ψ_G	.297(.0038)	.242(.0029)	18.5%	.232(.0034)	21.9%
	ψ_L	.315(.0072)	.281(.0052)	10.8%	.215(.0050)	31.7%
Liver	SVM_G	.724(.0053)	.655(.0028)	9.53%	.648(.0025)	10.5%
	SVM_L	.663(.0051)	.647(.0031)	2.41%	.635(.0021)	4.22%
	ψ_G	.690(.0021)	.650(.0019)	5.80%	.643(.0017)	6.81%
	ψ_L	.672(.0026)	.665(.0024)	1.04%	.628(.0019)	6.55%
Diabetes	SVM_G	.587(.0053)	.542(.0038)	7.67%	.536(.0020)	8.69%
	SVM_L	.545(.0026)	.526(.0024)	3.49%	.526(.0027)	3.49%
	ψ_G	.591(.0057)	.546(.0041)	7.61%	.536(.0021)	9.31%
	ψ_L	.573(.0028)	.528(.0029)	7.85%	.528(.0029)	7.85%
Ionosphere	SVM_G	.430(.0062)	.242(.0052)	43.7%	.250(.0040)	41.9%
	SVM_L	.526(.0102)	.383(.0045)	27.2%	.384(.0039)	27.0%
	ψ_G	.469 (0067)	.239(.0046)	49.0%	.243(.0048)	48.2%
	ψ_L	.466(.0089)	.383(.0045)	17.8%	.382(.0039)	18.0%

Table 4: Averaged estimated GKL losses and prediction errors of our method with ψ-learning, penalised logistic regression and nearest neighbour in the benckmark examples over nondegenerate simulation replications. Here TE denotes the test error, with tuning performed as in Wang & Shen (2006).

Method	CRE for probability estimation			TE for classification		
	NN	PLR	Our	NN	PLR	Our
Mushroom	-	-	.215(.0050)	.482(.0008)	.085(.0034)	.065(.0021)
Liver	-	.665(.0049)	.628(.0019)	.578(.0017)	.335(.0032)	.316(.0034)
Diabetes	-	.553(.0052)	.528(.0029)	.348(.0007)	.252(.0017)	.232(.0021)
Ionosphere	-	-	.243(.0048)	.642(.0014)	.195(.0033)	.083(.0024)

Table 5: Estimated GKL losses of our method with $m = 19$, the original Platt method and the tuned Platt method for the leukaemia data over a testing set, with tuning performed as in Section 3.2. Improv. denotes the percentage of improvement over the original Platt method.

Classifier	Platt's	Tuned Platt's	Improv.	Our	Improv.
SVM_L	.343	.343	0.00%	.241	29.7%
ψ_L	.171	.171	0.00%	.133	22.2%

Figure 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 minimiser of (3) with linear kernel.