ADAPTIVE LEARNING MACHINES FOR NONLINEAR CLASSIFICATION AND BAYESIAN INFORMATION CRITERIA

Ando, Tomohiro
Graduate School of Mathematics, Kyushu University

Imoto, Seiya
Institute of Medical Science, University of Tokyo

Konishi, Sadanori
Graduate School of Mathematics, Kyushu University

https://doi.org/10.5109/12706

出版情報：Bulletin of informatics and cybernetics. 36, pp.147-162, 2004-12. Research Association of Statistical Sciences
バージョン：
権利関係：

ADAPTIVE LEARNING MACHINES FOR NONLINEAR CLASSIFICATION AND BAYESIAN INFORMATION CRITERIA

by

Tomohiro Ando, **Seiya Imoto** and **Sadanori Konishi**

Reprinted from the Bulletin of Informatics and Cybernetics
Research Association of Statistical Sciences, Vol.36

FUKUOKA, JAPAN
2004

Bulletin of Informatics and Cybernetics, Vol. 36, 2004

ADAPTIVE LEARNING MACHINES FOR NONLINEAR CLASSIFICATION AND BAYESIAN INFORMATION CRITERIA

By

Tomohiro Ando[*], **Seiya Imoto**[†] and **Sadanori Konishi**[‡]

Abstract

Regularization is a well-known method for the treatment of mathematically ill-posed problems. By using the method of regularization, we propose a new machine learning algorithm, adaptive learning machine, to classify the high-dimensional data with complex structure. A crucial issue in the model constructing process is the choice of a suitable model among candidates. We present a Bayesian information criterion to evaluate models estimated by regularization. Real data analysis and Monte Carlo experiments show that our proposed method performs well in various situations.

Key Words and Phrases: Bayes approach, Classification, Genetic algorithm, Regularization theory.

1. Introduction

Recently, the regularization theory (Poggio and Girosi (1990)) has received considerable attention in various fields of statistical science. It provides a unified theoretical framework for designing the widely applied machine learning technique based on nonlinear models such as neural networks, support vector machines, regularization networks and splines. Since these nonlinear models are generally characterized by a large number of parameters, the method of regularization plays an important role to balance the fitting of the data with constraints on the model flexibility, producing a robust model that generalizes successfully.

Among many different varieties of machine learning techniques, support vector machines have been advocated as a useful tool for understanding the system with complex structure (Cortes and Vapnik (1995), Vapnik (1998), Wahba *et al.* (2000)). A remarkable point of support vector machines is the truncation property of loss function. Due to this property, many parameters are automatically estimated as zero and the estimated model can avoid overfitting, and make good generalization capability.

However, despite its success, there are a lot of practical disadvantages in the support vector machines. First, support vector machine makes unnecessarily kernel functions in

[*] Graduate School of Mathematics, Kyushu University 6–10–1, Hakozaki, Higashi-Ku Fukuoka 812-8581 Japan. tel +81–92–642–2779 ando@math.kyushu-u.ac.jp

[†] Institute of Medical Science, University of Tokyo 4–6–1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan. +81-3-5449-5615 imoto@ims.u-tokyo.ac.jp

[‡] Graduate School of Mathematics, Kyushu University 6–10–1, Hakozaki, Higashi-Ku Fukuoka 812-8581 Japan. tel +81–92–642–2779 konishi@math.kyushu-u.ac.jp

machine learning process. Since the number of support vectors is usually much smaller than the sample size, the computational cost increases as the number of data increases. Second, the tuning parameters are generally determined by a cross-validation, which is wasteful both of training data and computational cost. Additionally, the weakness of support vector machine significantly appears to lie in classification problem; the conditional probabilities of class membership are not available, and its extension to multi-class case is still under going issue.

Utilizing dramatically fewer kernel functions than support vector machines, Zhu and Hastie (2001) proposed import vector machines for probabilistic multi-class classification problems. Although import vector machine gives a computational and practical advantage over the support vector machine, it still remains critical problems to be solved. First, the regularization parameter and the import points are chosen by using the misclassification error on the tuning set, which results in an unstable prediction since the evaluation is significantly depending on how the tuning set is made. Second, although the model is dependent on the kernel parameter such as width parameter in Gaussians, Zhu and Hastie (2001) do not take this critical problem into account. Moreover, they used a simple greedy approach for searching the optimal model, which needs to a large computational cost and may result in local minima.

In order to choose the tuning parameters in the import vector machine, we take a Bayesian approach. Schwarz (1978) proposed the Bayesian information criterion, BIC. However, note that, theoretically, the BIC covers only models estimated by the maximum likelihood method. It still remains to construct a criterion for evaluating models estimated by regularization. By using the general result given by Konishi *et al.* (2004), we derive Schwarz's (1978) Bayesian information criterion so that it can be applied to the evaluation of models estimated by regularization. In order to obtain the optimal model in the candidate space, we use the genetic algorithms, originally developed by Holland (1975) and Goldberg (1989). By combining the model structure of import vector machine, Bayesian approach and genetic algorithms, we develop adaptive learning machines for constructing probabilistic multi-class classification models in the regularization framework.

In Section 2, we briefly describe a general framework of regularization theory and show the relationship among several machine learning techniques proposed previously. Section 3 presents adaptive learning machines for treating probabilistic multi-class classification problems. In Section 4, we conduct real data analyses and Monte Carlo experiments to examine the efficiency of the proposed machine learning algorithms. Some concluding remarks are given in Section 5.

2. Preliminaries

Regularization is the most popular and preferred method for the treatment of mathematically ill-posed problems. It has been applied successfully to numerous machine learning problems in constructing flexible models with generalization capacity. When a set of data $\{(\boldsymbol{x}_\alpha, y_\alpha); \boldsymbol{x}_\alpha \in R^p, y_\alpha \in R, \alpha = 1, ..., n\}$ is given, regularization theory formulates the machine learning problem as a functional variational problem of finding the function $h(\boldsymbol{x})$ that minimizes a functional of the form

$$\ell_\lambda(h) = \frac{1}{n}\sum_{\alpha=1}^{n} \psi(y_\alpha, h(\boldsymbol{x}_\alpha)) + \frac{\lambda}{2}||h||^2_{\mathcal{H}}, \tag{1}$$

where $\psi(\cdot,\cdot)$ represents the loss function that measures the fitness of the model responses to each data, $||h||^2_{\mathcal{H}}$ corresponds to the regularization functional given by a norm in the Hilbert Space $\mathcal{H}$ of the functions. The regularization parameter λ is used to control the tradeoff between fitness of the model's responses to the data and the complexity of the model (Hastie and Tibshirani (1990), Poggio and Girosi (1990)).

When the space of models considered is a reproducing kernel Hilbert space (RKHS) defined by the positive definite function ϕ, the minimizer of equation (1) establishes a representation of the function $h(\boldsymbol{x})$ as a linear combination of kernels centered in each data point (Kimeldorf and Wahba (1971))

$$h(\boldsymbol{x}) = \sum_{j=1}^{n} \gamma_j \phi(\boldsymbol{x}_j; \boldsymbol{x}) + b, \tag{2}$$

where $\phi(\boldsymbol{x}_j; \boldsymbol{x})$ is a certain symmetric positive definite function, and $\{b, \gamma_1, ..., \gamma_n\}$ is a set of parameters to be estimated.

This regularization framework reduces to different machine learning techniques by specifying the loss function ψ, each leading to a different estimation procedure. In particular, we will review four machine learning techniques which have recently been proposed for both classification and regression problems:

Support vector classification machine:	$\psi(y, h(\boldsymbol{x})) = \lvert 1 - yh(\boldsymbol{x})\rvert_+$,
Support vector regression machine:	$\psi(y, h(\boldsymbol{x})) = \lvert y - h(\boldsymbol{x})\rvert_\varepsilon$,
Regularized networks:	$\psi(y, h(\boldsymbol{x})) = (y - h(\boldsymbol{x}))^2$,
Kernel logistic regression machine:	$\psi(y, h(\boldsymbol{x})) = \log[1 + \exp\{-yh(\boldsymbol{x})\}]$,

where $[a]_+ = \max(0, a)$ is a truncated loss function, and $|a|_\varepsilon = 0$ ($|a| \leq \varepsilon$), $= |a| - \varepsilon$ (otherwise) is an ε -insensitive loss function.

The loss function of support vector classification machines (and also regression machines) has a highly effective mechanism, the truncation property. Because only the points on the outside of the margin and those on the inside but near the maximum margin have a large influence in constructing the machines, we can construct a sparse machine with good generalization. However, we can identify a number of significant disadvantages of support vector classification methodology. (1) Predictions are not probabilistic. (2) The extension to the multi-class case is still under going issue. (3) The tuning parameters that include the regularization parameter λ are generally optimized by a cross-validation procedure, which is wasteful both of data and computation.

In the next section, to overcome these critical problems, we propose adaptive learning machines, so that we can easily consider the probabilistic multi-class classification problem.

3. Adaptive learning machine

In this section, we first describe the basic idea of proposed adaptive learning machines in the two-class classification framework. Then we naturally generalize the adaptive learning machines to the multi-class case.

3.1. Two-class classification model

Consider the two class classification problem $Y \in \{0, 1\}$ based on measurements of a set of characteristics $\boldsymbol{x} \in R^p$. As pointed out in the last part of foregoing section, the absence of probabilistic property in support vector machine classification is one of the critical problems. In other words, support vector machine only estimates $\text{sign}[P(Y = 1|\boldsymbol{x}) - 1/2]$, regardless of the conditional probability $P(Y = 1|\boldsymbol{x})$ that is often of interest itself. Although the kernel logistic regression machine overcomes this problem by using the negative log likelihood function of the binomial distribution, it compromises the truncated loss function of the support vector machine. It therefore no longer has the "support points" property, and the number of parameters included in the model always exceeds the number of observation, which we call over-parameterization.

To avoid over-parameterization problem, Zhu and Hastie (2001) approximated the full model (2) by using a linear combination of fewer kernels

$$h(\boldsymbol{x}) = \sum_{j \in S} \gamma_j \phi(\boldsymbol{x}_j; \boldsymbol{x}) + b = \sum_{j=1}^{m} \gamma_j \phi(\boldsymbol{x}_j; \boldsymbol{x}) + b, \tag{3}$$

where S is a subset of the training data that consists of important points, and $m < n$ is the number of observations included in S. The advantage of sub-model (3) is that the computational cost is reduced, especially for large data sets (Zhu and Hastie (2001)).

The RKHS formulation provides for a very rich structure for a collection of kernels and their properties. By using the Gaussian kernel functions

$$\phi(\boldsymbol{z}; \boldsymbol{x}) = \exp\left\{ -\sum_{k=1}^{p} \frac{(x_k - z_k)^2}{2\sigma^2} \right\}, \tag{4}$$

we obtain a solution $h(\boldsymbol{x})$ in (3). Here $\boldsymbol{z} = (z_1, ..., z_p)'$ is the p-dimensional vector determining the centers of Gaussians, and the width parameter σ controls the shape of Gaussian kernel function.

Substituting the negative log likelihood of the binomial distribution for the loss function ψ in (1), the unknown parameters b and $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_m)^T$ are estimated by minimizing the penalized negative log likelihood function

$$\begin{aligned} \ell_\lambda(\boldsymbol{w}) &= \frac{1}{n} \sum_{\alpha=1}^{n} \left(\log\left[1 + \exp\left\{h(\boldsymbol{x}_\alpha)\right\}\right] - y_\alpha h(\boldsymbol{x}_\alpha) \right) + \frac{\lambda}{2} ||h||^2_{\mathcal{H}} \\ &= \frac{1}{n} \sum_{\alpha=1}^{n} \left(\log\left[1 + \exp\left\{\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_\alpha)\right\}\right] - y_\alpha \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_\alpha) \right) + \frac{\lambda}{2} \boldsymbol{\gamma}^T K \boldsymbol{\gamma}, \end{aligned} \tag{5}$$

where $\boldsymbol{w} = (b, \boldsymbol{\gamma}^T)^T$, $\boldsymbol{\phi}(\boldsymbol{x}) = (1, \phi(\boldsymbol{x}_1, \boldsymbol{x}), ..., \phi(\boldsymbol{x}_m, \boldsymbol{x}))^T$ and K is an m-dimensional kernel matrix constructed by a subset of the training data S with (i, j)-th element $K_{ij} = \phi(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Note that the nonlinear function $h(\boldsymbol{x})$ in (3) estimates the log-odds ratio of the conditional probability $\log\{\Pr(Y = 1|\boldsymbol{x}) / \Pr(Y = 0|\boldsymbol{x})\} = h(\boldsymbol{x})$ and we can formulate the probabilistic classification procedure.

The optimization process with respect to the unknown parameters is nonlinear and the score function $\partial \ell_\lambda(\boldsymbol{w}) / \partial \boldsymbol{w} = \boldsymbol{0}$ does not have an explicit solution, hence we use the

Fisher's scoring algorithm. Initializing $\boldsymbol{w}^0$, the solution $\hat{\boldsymbol{w}}$ can be obtained by the Fisher scoring algorithm that updates $\boldsymbol{w}$ by

$$\boldsymbol{w}^{new} = \left(\Phi^T W \Phi + n\lambda R\right)^{-1} \Phi^T W \boldsymbol{\zeta}, \quad R = \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & K \end{pmatrix}, \tag{6}$$

until a suitable convergence criterion is satisfied. Here $\Phi = (\boldsymbol{\phi}(\boldsymbol{x}_1), ..., \boldsymbol{\phi}(\boldsymbol{x}_n))^T$, W is an $n \times n$ diagonal matrix and $\boldsymbol{\zeta}$ is an n-dimensional vector with α-th elements are given by

$$W_{\alpha\alpha} = \pi(\boldsymbol{x}_\alpha)(1 - \pi(\boldsymbol{x}_\alpha)) \quad \text{and} \quad \zeta_\alpha = \frac{y_\alpha - \pi(\boldsymbol{x}_\alpha)}{\pi(\boldsymbol{x}_\alpha)(1 - \pi(\boldsymbol{x}_\alpha))} + \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_\alpha),$$

respectively. The conditional probability of a sample being in class 1, $\pi(\boldsymbol{x}) := P(Y = 1|\boldsymbol{x})$, is given by

$$\pi(\boldsymbol{x}) = \frac{\exp\{\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})\}}{1 + \exp\{\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})\}}. \tag{7}$$

Replacing the unknown parameter vector $\boldsymbol{w}$ in (7) by its sample estimate $\hat{\boldsymbol{w}}$ yields the two-class classification model as the logistic density of the form:

$$f(y|\boldsymbol{x}; \hat{\boldsymbol{w}}) = \hat{\pi}(\boldsymbol{x})^y (1 - \hat{\pi}(\boldsymbol{x}))^{1-y}, \tag{8}$$

where $\hat{\pi}(\boldsymbol{x}) = 1/[1 + \exp\{-\hat{\boldsymbol{w}}^T \boldsymbol{\phi}(\boldsymbol{x})\}]$ is the estimated conditional probability. Using the constructed model (8), the future observation $\boldsymbol{x}$ is classified into $\hat{Y} = 1$ if $\hat{\pi}(\boldsymbol{x}) > 0.5$ and $\hat{Y} = 0$ if $\hat{\pi}(\boldsymbol{x}) < 0.5$.

The essential problem is how to find a subset S such that the model (3) is a good approximation of the full model (2). Moreover, we have to choose appropriate values of regularization parameter λ and the width parameter σ in the Gaussian kernel function (4). We therefore give a Bayesian information criterion for evaluating the models estimated by regularization.

Several methods have been proposed for selecting the subset S to approximate the full model. Smola and Schölkopf (2000) developed a greedy algorithm to sequentially select m columns of full kernel matrix K $(n \times n)$, such that the span of these m columns approximates the span of whole kernel matrix K well in the Frobenius norm. Lin *et al.* (2000) divided the observed data into several clusters, then randomly selected the inputs from each cluster to make up S. However, these algorithms only use the information of input vectors $\boldsymbol{x}$, and does not take the information of the class label Y into account. In order to adopt the both information of $\boldsymbol{x}$ and Y, Zhu and Hastie (2001) randomly divided all the data into a training set and a tuning set, and used the misclassification error on the tuning set as a criterion. However, the cross validation procedure produces the high variability results and the prediction ability of a constructed model would be unstable. In addition, they used a simple greedy approach, which needs to a large computational cost.

3.2. Bayesian information criterion

The two-class classification model introduced in the previous section can be constructed when we fix a subset S, regularization parameter λ and the width parameter σ. In order to choose these tuning parameters, we derive a Bayesian information criterion by using the general result due to Konishi *et al.* (2004).

Suppose we are interested in selecting a machine from a set of candidate machines $M_1, \cdots, M_r$ for a set of given observation $D = \{(\boldsymbol{x}_\alpha, y_\alpha); \alpha = 1, ..., n\}$. Let $\log \pi_k(\boldsymbol{w}_k|\lambda_k) = O(n)$ be the prior density for the q_k-dimensional parameter vector $\boldsymbol{w}_k$ under a model M_k, where λ_k is a hyperparameter. The posterior probability of the model M_k for the dataset D is then given by

$$P(M_k|D) = \frac{P(M_k)\int L(D|\boldsymbol{w}_k)\pi_k(\boldsymbol{w}_k|\lambda_k)d\boldsymbol{w}_k}{\sum_{\alpha=1}^{r} P(M_\alpha)\int L(D|\boldsymbol{w}_\alpha)\pi_\alpha(\boldsymbol{w}_\alpha|\lambda_\alpha)d\boldsymbol{w}_\alpha},$$

where $L(D|\boldsymbol{w}_k) = \prod_{\alpha=1}^{n} f(y_\alpha|\boldsymbol{x}_\alpha; \boldsymbol{w}_k)$ and $P(M_k)$ are the likelihood function and the prior probability for model M_k, respectively.

The Bayes approach for selecting a model is to choose the model with the largest posterior probability among a set of candidate models for given values of λ_k. This is equivalent to choosing the model that maximizes the product of the prior probability, and the marginal probability of the data under model M_k,

$$P(M_k|D, \lambda_k) = P(M_k)\int L(D|\boldsymbol{w}_k)\pi_k(\boldsymbol{w}_k|\lambda_k)d\boldsymbol{w}_k. \tag{9}$$

Then using the Laplace approximation (Davison (1986), Tierney and Kadane (1986)) for integrals, we have

$$\int L(D|\boldsymbol{w}_k)\pi_k(\boldsymbol{w}_k|\lambda_k)d\boldsymbol{w}_k \approx \frac{(2\pi)^{q_k/2}}{n^{q_k/2}|H(\hat{\boldsymbol{w}}_k; \lambda_k)|^{1/2}} \exp\left\{n\ell(D, \hat{\boldsymbol{w}}_k, \lambda_k)\right\}, \tag{10}$$

where

$$\begin{aligned} \ell(D, \boldsymbol{w}_k, \lambda_k) &:= \{\log L(D|\boldsymbol{w}_k) + \log \pi_k(\boldsymbol{w}_k|\lambda_k)\}/n, \\ H(\boldsymbol{w}_k; \lambda_k) &= -\partial^2\{\ell(D, \boldsymbol{w}_k, \lambda_k)\}/\partial\boldsymbol{w}_k\partial\boldsymbol{w}_k^T, \end{aligned}$$

and $\hat{\boldsymbol{w}}_k$ is the mode of $\ell(D, \boldsymbol{w}_k, \lambda_k)$ and is corresponds to the minimizer of the penalized negative log likelihood function (5).

Substituting the Laplace approximation in equation (9) and taking the logarithm of the resulting formula, we have a Bayesian information criterion

$$\begin{aligned} \text{BIC} &= -2\log P(M_k|D, \lambda_k) \\ &\approx -2n\ell(D, \boldsymbol{w}_k, \lambda_k) - q_k\log(2\pi/n) + \log|H(\hat{\boldsymbol{w}}_k; \lambda_k)| - 2\log P(M_k). \end{aligned} \tag{11}$$

Choosing the model with the largest posterior probability among a set of candidate models for given values of λ_k is equivalent to choosing the model that minimizes the criterion (11).

Suppose that the two-class classification model (8) is constructed by minimizing the penalized log likelihood function (5), the regularization term corresponds to a singular multivariate normal prior density,

$$\pi(\boldsymbol{w}|\lambda) = (2\pi)^{-m/2}(n\lambda)^{m/2}|R|_+^{1/2}\exp\left\{-\frac{n\lambda}{2}\boldsymbol{w}^T R\boldsymbol{w}\right\}, \tag{12}$$

in which R is defined by (6) and $|R|_+$ is the product of m nonzero eigenvalues of R.

Substituting the densities of models (8) and priors (12) into the equation (11), we have a Bayesian information criterion for evaluating the models estimated by minimizing the penalized negative log likelihood function:

$$\begin{aligned}\mathrm{BIC}(S,\lambda,\sigma) &= 2\sum_{\alpha=1}^{n}\left(\log\left[1+\exp\left\{\hat{\boldsymbol{w}}^T\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\right\}\right]-y_\alpha\hat{\boldsymbol{w}}^T\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\right)+n\lambda\hat{\boldsymbol{w}}^T R\hat{\boldsymbol{w}}\\ &\quad -\log(2\pi/n)+\log|H(\hat{\boldsymbol{w}};\lambda)|-\log|R|_+-m\log\lambda-2\log P(M_k),\end{aligned} \tag{13}$$

where $H(\hat{\boldsymbol{w}};\lambda) = \Phi^T\Gamma\Phi/n + \lambda R$ and Γ is an n-dimensional diagonal matrix with α th element $\Gamma_{\alpha\alpha} = \exp\{\hat{\boldsymbol{w}}^T\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\}/[1+\exp\{\hat{\boldsymbol{w}}^T\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\}]^2$. The optimal model is chosen such that the criterion BIC in (13) is minimal. Ando *et al.* (2002) approximated the full model (2) by using a radial basis function networks and proposed generalized Bayesian information criteria for evaluating radial basis function network multi-class classification models estimated by regularization.

The regularization parameter λ, the width parameter σ and a subset of training data S are determined as the minimizer of the Bayesian information criterion $\mathrm{BIC}(S,\lambda,\sigma)$. However, it is nearly impossible to perform an exhaustive search for choosing λ, σ and S, because the search space is very large. To solve this problem, we use a genetic algorithm to obtain the optimal model.

3.3. Genetic algorithm

Genetic algorithms (GAs; Holland (1975), Goldberg (1989)) are search methods guided by the principles of evolution and natural genetics. It is well known that GAs are robust optimization techniques and have the ability to efficiently explore large search spaces.

In general, GAs contain a constant size population of potential solutions over the search space. Each potential solution is represented by a finite string of symbols encoding a possible solution. The initial population can be created randomly or based on the prior knowledge. In each step, called a generation, a new population is created based on a preceding one through the following three steps: *evaluation*, *selection* and *mutation*. In evaluation step, each individual of the current population is evaluated using a fitness function and given a value to represent its goodness, and then, individuals with better fitness are selected in selection step to generate next population. In mutation step, genetic operators are applied to the selected individuals to produce new individuals for the next generation. These three steps are iterated for many generations until a satisfactory solution is found or a terminated criterion is satisfied.

To incorporate GAs into our model selection problem, we need to specify the following four components:

(*a*) *Solution representation*

The search space is all possible subset S, which comprises all possible solutions. In our utilization, an individual solution in a population is an n-dimensional binary coded vector $\boldsymbol{\delta} = (\delta_1, ..., \delta_n)^T$, where $\delta_\alpha = 1$ if corresponding α-th observation is included in S, and 0 otherwise. We create a population of individuals and evaluate each individual, and then select the better individuals to construct the next population. Finally, the n-dimensional vector with the highest fitness value is selected.

(*b*) *Evaluation function*

For each individual $\boldsymbol{\delta} = (\delta_1, ..., \delta_n)^T$, we set

$$h(\boldsymbol{x}) = \sum_{\{j;\delta_j=1\}} \gamma_j \phi(\boldsymbol{x}_j, \boldsymbol{x}) + b. \tag{14}$$

The unknown parameters are estimated by minimizing the negative penalized log likelihood function (5). Then the BIC is used as an evaluation function for evaluating the fitness of $\boldsymbol{\delta}$.

(*c*) *Selection*

To form a new population, the individuals are selected from the population according to their fitness. We use the Holland's (1975) fitness-proportionate technique, where individuals are selected with a probability

$$P(\boldsymbol{\delta}) = \frac{\exp\{-\mathrm{BIC}(\boldsymbol{\delta})/2\}}{\sum_{\boldsymbol{\delta}'} \exp\{-\mathrm{BIC}(\boldsymbol{\delta}')/2\}}. \tag{15}$$

Here the denominator sums up BIC for all individuals in the population. Note that the defined probability $P(\boldsymbol{\delta})$ corresponds to the posterior probability of the model among a set of candidate models.

(*d*) *Genetic operators*

Genetic operators alter genetic composition of children during reproduction. We use two primary genetic operators, *crossover* and *mutation*, that are standard in GAs. The crossover operator randomly pairs individuals, and swaps parts of their genetic information to produce new individuals. We use the one-point crossover operator that selects two parents and produces two children. A randomly selected point is generated and used to cut each parent into two parts. Two children are formed by swapping the parts of parents demarcated by the crossover point. The mutation operator creates a new individual by inverting one or more genes of an individual to increase the variability of the population. We use a random mutation method that replaces a gene with a random gene from the corresponding solution domain.

Learning algorithm

For fixed value of λ^* and σ^*, the search space considered in GAs is all possible subset S, which comprises all possible solutions. We choose the best model as minimizer of $\mathrm{BIC}(S, \lambda^*, \sigma^*)$ over a set of competing models. This procedure is carried out for various combinations of λ and σ. Finally, we choose the best one that minimizes $\mathrm{BIC}(S, \lambda, \sigma)$ among the candidates. We can summarize the algorithm for finding the optimal model as follows:

Step 1. Fix λ^*, σ^*, the population size Z and maximum generation number, and initialize the population $G^{(0)} = \{\boldsymbol{\delta}_1^{(0)}, ..., \boldsymbol{\delta}_Z^{(0)}\}$.

Step 2. For each string of symbol $\boldsymbol{\delta}_j^{(l)} = (\delta_{j1}^{(l)}, ..., \delta_{jn}^{(l)})^T$ $(j = 1, ..., Z)$, in l th generation $G^{(l)}$, measure the fitness by using BIC in (13).

Step 3. Select the individuals according to the probability (15) proportional to their relative fitness.

Step 4. Generate a new population using genetic operators.

Step 5. Repeat Step 2 $\sim$ 4 until the maximum generation number is achieved.

Step 6. Choose the best model as minimizer of $\mathrm{BIC}(S, \lambda^*, \sigma^*)$ over a set of competing models.

Step 7. Repeat Step 1 $\sim$ 6 for various combinations of λ and σ and choose the final one that minimizes $\mathrm{BIC}(S, \lambda, \sigma)$.

3.4. Multi-class classification

The adaptive learning machines can naturally be generalized to the multi-class case, especially L classes. Suppose we have n set of data $\{(\boldsymbol{x}_\alpha, \boldsymbol{y}_\alpha); \boldsymbol{x}_\alpha \in R^p, \alpha = 1, .., n\}$, where the L dimensional binary variable vector $Y = (Y_1, ..., Y_L)'$ is coded as either 0 or 1 to indicate the group membership of a sample: it means $y_k = 1$ if the observation belongs to the kth class, or $y_k = 0$ otherwise. A natural extension of the two-class classification model to the multi-class classification model is to use the multiple log-odds in the form

$$\log\left\{\frac{\Pr(y_k = 1|\boldsymbol{x})}{\Pr(y_L = 1|\boldsymbol{x})}\right\} = h_k(\boldsymbol{x}), \qquad k = 1, ..., L-1,$$

where $h_k(\boldsymbol{x})$ belongs to a reproducing kernel Hilbert space $\mathcal{H}_k$. Then the penalized negative log likelihood function is

$$\ell_{\lambda_1,...,\lambda_{L-1}} = \frac{1}{n}\sum_{\alpha=1}^{n}\left[\log\left(1 + \sum_{j=1}^{L-1}\exp\{h_j(\boldsymbol{x}_\alpha)\}\right) - \sum_{k=1}^{L-1} y_{k\alpha} h_k(\boldsymbol{x}_\alpha)\right] + \sum_{j=1}^{L-1}\frac{\lambda_j}{2}||h_j||^2_{\mathcal{H}_k}. \quad (16)$$

Using the representer theorem, we derive the unknown function $h_k(\boldsymbol{x})$ that minimizes the penalized negative log likelihood function (16). It is a linear combination of the kernel functions. Especially, we shall use the following functional form

$$h_k(\boldsymbol{x}) = \sum_{j=1}^{m}\gamma_{kj}\phi(\boldsymbol{x}_j, \boldsymbol{x}) + b_k = \boldsymbol{w}_k^T\boldsymbol{\phi}(\boldsymbol{x}), \qquad k = 1, ..., L-1,$$

where $\boldsymbol{w}_k = (b_k, \gamma_{k1}, ..., \gamma_{km})^T$ and $\boldsymbol{\phi}(\boldsymbol{x}) = (1, \phi(\boldsymbol{x}_1, \boldsymbol{x}), ..., \phi(\boldsymbol{x}_m, \boldsymbol{x}))^T$, respectively.

Substituting this expression into (16), we have

$$\ell_{\lambda_1,...,\lambda_{L-1}}(\boldsymbol{w}_1, ..., \boldsymbol{w}_{L-1}) = \frac{1}{n}\sum_{\alpha=1}^{n}\left[\log\left(1 + \sum_{j=1}^{L-1}\exp\{\boldsymbol{w}_j^T\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\}\right) - y_{k\alpha}\boldsymbol{w}_k^T\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\right] + \sum_{j=1}^{L-1}\frac{\lambda_j}{2}\boldsymbol{w}_j^T R\boldsymbol{w}_j, \quad (17)$$

where R is an $(m+1)$-dimensional matrix defined by (6). By using the first and second derivatives

$$\begin{aligned}\frac{\partial \ell_{\lambda_1,...,\lambda_{L-1}}}{\partial \boldsymbol{w}_k} &= \frac{1}{n}\sum_{\alpha=1}^{n}\{\pi_k(\boldsymbol{x}_\alpha) - y_{k\alpha}\}\,\boldsymbol{\phi}(\boldsymbol{x}_\alpha) + \lambda_k R\boldsymbol{w}_k, \qquad k = 1, ..., L-1,\\ \frac{\partial \ell_{\lambda_1,...,\lambda_{L-1}}}{\partial \boldsymbol{w}_m \partial \boldsymbol{w}_l^T} &= \frac{1}{n}\sum_{\alpha=1}^{n}\pi_m(\boldsymbol{x}_\alpha)\{\delta(m,l) - \pi_l(\boldsymbol{x}_\alpha)\}\boldsymbol{\phi}(\boldsymbol{x}_\alpha)\boldsymbol{\phi}^T(\boldsymbol{x}_\alpha) + \lambda_l\delta(m,l)R,\end{aligned}$$

the unknown parameter vectors $\boldsymbol{w}_1, ..., \boldsymbol{w}_{L-1}$ are estimated by using the Newton-Raphson algorithm. Here $\delta(m,l)$ is the delta function and $\pi_k(\boldsymbol{x}) := P(Y_k = 1|\boldsymbol{x})$ are the conditional probabilities of a sample $\boldsymbol{x}$ being in class k given by

$$\begin{aligned}\pi_k(\boldsymbol{x}) &= \frac{\exp\left\{\boldsymbol{w}_k^T\boldsymbol{\phi}(\boldsymbol{x})\right\}}{1 + \sum_{j=1}^{L-1}\exp\left\{\boldsymbol{w}_j^T\boldsymbol{\phi}(\boldsymbol{x})\right\}}, \quad \text{for} \quad k = 1, ..., L-1,\\ \pi_L(\boldsymbol{x}) &= \frac{1}{1 + \sum_{k=1}^{L-1}\exp\left\{\boldsymbol{w}_k^T\boldsymbol{\phi}(\boldsymbol{x})\right\}}.\end{aligned} \tag{18}$$

Substituting the sample estimates $\hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_{L-1}$ into $h_k(\boldsymbol{x})$, we then have the multiclass classification model as the multinomial density

$$f(\boldsymbol{y}|\boldsymbol{x}; \hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_{L-1}) = \prod_{k=1}^{L}\hat{\pi}_k(\boldsymbol{x})^{y_k}, \tag{19}$$

where $\hat{\pi}_k(\boldsymbol{x})$ are estimated conditional probabilities obtained by replacing the unknown parameters in (18) by their sample estimates. Given the estimates $\hat{\pi}_k(\boldsymbol{x})$, a future observation $\boldsymbol{x}$ may be assigned by using the discriminant rule:

$$\text{assign } \boldsymbol{x} \text{ to class } j \quad \text{if} \quad \hat{\pi}_j(\boldsymbol{x}) = \max_k \hat{\pi}_k(\boldsymbol{x}).$$

As in the two-class case, an essential problem is how to choose the regularization parameters $\lambda_1, ..., \lambda_{L-1}$, width parameter σ and a subset of the training data S which construct the function $h_k(\boldsymbol{x})$. We therefore give a Bayesian information criterion for determining them

$$\begin{aligned}\mathrm{BIC}(S, \lambda_1, ..., \lambda_{L-1}, \sigma) = &-2\sum_{\alpha=1}^{n}\sum_{k=1}^{L} y_{k\alpha}\log\{\hat{\pi}_k(\boldsymbol{x}_\alpha)\} + n\sum_{j=1}^{L-1}\lambda_j\hat{\boldsymbol{w}}_j^T R\hat{\boldsymbol{w}}_j - 2\log P(M_k)\\ &+\log|H(\hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_{L-1}; \lambda_1, ..., \lambda_{L-1})| - (L-1)\log|R|_+\\ &+(L-1)m\log(n/2\pi) - m\sum_{j=1}^{L-1}\log\lambda_j,\end{aligned} \tag{20}$$

where $H(\hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_{L-1}; \lambda_1, ..., \lambda_{L-1})$ is $(L-1)(m+1)$-dimensional matrix

$$H(\hat{\boldsymbol{w}}_1, ..., \hat{\boldsymbol{w}}_{L-1}; \lambda_1, ..., \lambda_{L-1}) = -\frac{1}{n}\left(B \otimes A\right)^T\left(B \otimes A\right) + \frac{1}{n}C + \mathrm{diag}\{\lambda_1 R, ..., \lambda_{L-1}R\},$$

with $A = (\Phi, ..., \Phi)$, $B = (\hat{\boldsymbol{\pi}}_{(1)}\mathbf{1}_{m+1}^T, ..., \hat{\boldsymbol{\pi}}_{(L-1)}\mathbf{1}_{m+1}^T)$, $C = \mathrm{diag}\{C_1, ..., C_{L-1}\}$, $C_j = \Phi^T\mathrm{diag}\{\hat{\boldsymbol{\pi}}_{(j)}\}\Phi$, $\boldsymbol{y}_{(k)} = (y_{k1}, ..., y_{kn})^T$, $\hat{\boldsymbol{\pi}}_{(k)} = (\hat{\pi}_k(\boldsymbol{x}_1), ..., \hat{\pi}_k(\boldsymbol{x}_n))^T$ and $\mathbf{1}_m = (1, ..., 1)^T$.

Here the operator $\otimes$ means the element wise product, (suppose that the arbitrary matrices $A = (a_{ij})$, $B = (b_{ij})$ are given, then $A \otimes B = (a_{ij} \times b_{ij})$.

Using the genetic algorithm, the regularization parameters $\lambda_1, ..., \lambda_{L-1}$, the width parameter in Gaussian kernel functions σ and a subset of training data S are determined as the minimizer of the Bayesian information criterion BIC (20).

4. Numerical results

In this section we illustrate the efficiency of our proposed adaptive learning machine through the real data analysis and Monte Carlo simulations. In GAs, a large-sized population will obtain better solutions because the population contains more representative solutions over the search space. However, more computational times are needed in a large sized population. We therefore set the population size to be $Z = 100$. We also set the generation number to be 100, and used the probabilities for crossover and mutation such that 0.5 and 0.25, respectively. The size of mutation was set to be 1, which replaces a gene with a random gene from the corresponding solution domain. We used equal prior probabilities for candidate models in Bayesian information criterion.

4.1. Monte Carlo experiments

We illustrate our proposed methods on two popular simulated examples, waveform data taken from Breiman *et al.* (1984) and synthetic data taken from Ripley (1994).

(*a*) *Waveform data*

The waveform data contain three classes with 21 variables and are generated by

$$x_k = \begin{cases} uH_1(k) + (1-u)H_2(k) + \varepsilon_k & if \quad \boldsymbol{x} \in \text{class } 1 \\ uH_1(k) + (1-u)H_3(k) + \varepsilon_k & if \quad \boldsymbol{x} \in \text{class } 2 \\ uH_2(k) + (1-u)H_3(k) + \varepsilon_k & if \quad \boldsymbol{x} \in \text{class } 3 \end{cases} \quad k = 1, ..., 21, \tag{21}$$

where u depends uniformly on $(0, 1)$, ε_k depends on the standard normal distribution and $H_i(k)$ are the shifted triangular waveforms: $H_1(k) = \max(6 - |k - 11|, 0)$, $H_2(k) = H_1(k-4)$ and $H_3(k) = H_1(k+4)$. We generated 300 training and 500 test samples with equal prior for each class by using the probability system (21).

(*b*) *Synthetic data*

The synthetic data consist of two dimensional feature vector and two-class class distribution $\{(x_{1\alpha}, x_{2\alpha}, y_\alpha) : y_\alpha \in \{0, 1\}, \alpha = 1, ..., n\}$. The synthetic data have 250 training and 1000 test samples and are generated form equal mixtures of normal distributions with centers $(-0.7, 0.3)$ and $(0.3, 0.3)$ in class $Y = 0$ and $(-0.3, 0.7)$ and $(0.4, 0.7)$ in class $Y = 1$, with variances 0.03.

For each dataset, we construct the adaptive learning machines by using training data and evaluate the prediction accuracy based on the test data. To show the efficiency of proposed machine learning algorithm, we compare the test errors of support vector machines with Gaussian kernel. For support vector machines, the regularization parameter and the width parameter of the Gaussian kernel were selected using five-fold cross validation on the training set. Since the waveform data is a three class classification problem, we train a set of two-class classifiers with Gaussian kernel each trained to

separate one class from the rest. The observation is then classified to the class, given the maximum output value.

Table 1 shows the test errors for various techniques. Both the benchmark examples indicate that our proposed adaptive learning machine performs very well; it gives the smallest value of the test error rate. The learning results for a synthetic data are given in Figure 1. The test error for the adaptive learning machines (9.3%) is slightly superior to the support vector machines (9.4%), but the remarkable feature of contrast is the complexity of the classifiers. The support vector machine utilizes 96 Gaussian kernel functions compared to just 7 for the adaptive learning machines. Similar results are also obtained for the waveform data.

Techniques	Waveform	Synthetic
Adaptive learning machines	**15.6**	**9.3**
Support vector machines	15.8	9.4
Linear discrimination	19.1	10.8
Quadratic discrimination	20.5	10.9
Classification tree	28.9	10.1
Flexible discriminant analysis (MARS)	19.1	9.6

Table 1: Comparison of the test errors (%). The results except for adaptive learning machines and support vector machines are due to Hastie *et al.* (1994). The test errors for the waveform data show the mean values with ten runs. MARS: multivariate adaptive regression splines.

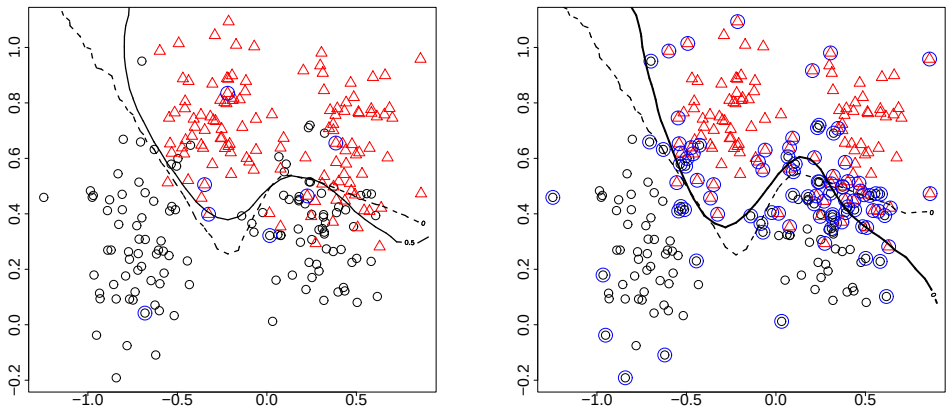

Figure 1: Adaptive learning machine (left) and support vector machine (right) classifiers on synthetic data. Samples are marked by open circles ∘ and open triangles Δ. An optimal Bayes decision boundary and the constructed decision boundaries are shown as the dashed lines (- - -) and the solid lines (—), respectively. A set S of imported points and the support vectors are shown circled ◯ to emphasize the dramatic reduction in complexity of the adaptive learning machines.

4.2. Real data analysis

We next apply our proposed methods to real datasets, the vowel recognition data (Ripley (1994)) and the Pima Indians diabetes data (Ripley (1996)).

(*a*) *Vowel recognition data*

The vowel recognition data (Ripley (1994)) is a popular benchmark for neural network algorithms, and consists of training and test data with 10 measurements and 11 classes. An ascii approximation to the International Phonetic Association symbol and the word in which the eleven vowel sounds were recorded. The word was uttered once by each of the fifteen speakers and for each utterance, ten floating-point input values were measured.

We construct the adaptive learning machine by using 528 training data from eight speakers (4 male and 4 female). The dynamics of the Bayesian information criterion and the size of important data points S in a genetic algorithm are shown in Figure 2. In this figure, the dynamics of the best individual in each population are traced. As shown in left side of Figure 2, GAs upgrade the adaptive learning machines since the BIC scores are reduced in each generation.

We then test the prediction capability of the constructed adaptive learning machine on 462 data from seven speakers (4 male and 3 female). The results of several classification procedures are shown in Table 2. As shown in Table 2, the proposed adaptive learning machine gives the smallest error rate.

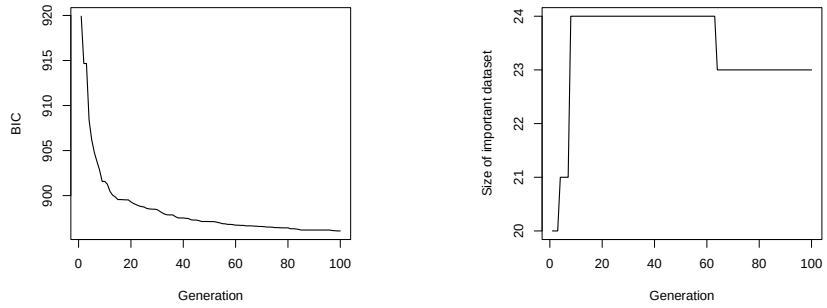

Figure 2: Evolution of the Bayesian information criterion of adaptive learning machines (left) and of the size of important dataset S (right) during a 100 generation.

Technique	Test error (%)
Adaptive learning machines	**41**
Linear discriminant analysis	56
Quadratic discriminant analysis	67
Classification tree	56
Flexible discriminant analysis (BRUTO)	44
Flexible discriminant analysis (MARS)	42
Single layer perceptron	67

Table 2: Vowel recognition. The results except for adaptive learning machines are due to Hastie *et al.* (1994). MARS: multivariate adaptive regression splines. BRUTO: spline additive models with BRUTO algorithm.

(*b*) *Pima Indians diabetes data*

Ripley (1996) provides an analysis of a dataset on diabetes in Pima Indian women. The main goal is to predict the presence of diabetes using seven covariates: the number of pregnancies, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, body mass index, diabetes pedigree, and age. There are 532 complete records, of which 200 are used as the training data and the other 332 are used as the test data. Figure 3 shows the relationship between the test error and the value of BIC. We can see from this figure that the smaller values of BIC give the smaller values of test error. This implies that the best model will be determined as the minimizer of the model selection criterion, BIC.

Table 3 summarizes the test error rate for the other classification methods. As indicated in table, the classification accuracy of our proposed adaptive learning machine is superior to all of the other classification methods.

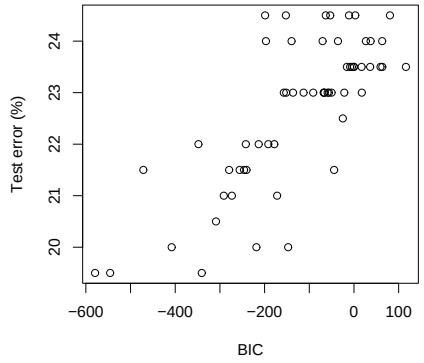

Figure 3: The relationship between test error rates and values of BIC.

Technique	Test error (%)
Adaptive learning machines	**19.8**
Linear discriminant analysis	24.7
Quadratic discriminant analysis	22.8
Logistic discrimination	19.9
Classification tree	24.4
Flexible discriminant analysis (MARS)	22.6
Projection pursuit regression	22.6
Multilayer perceptron network	22.6

Table 3: Pima Indians diabetes data. The results except for adaptive learning machines are due to Hastie *et al.* (1994). MARS: multivariate adaptive regression splines.

5. Conclusion

The regularization theory has been widely used in machine learning algorithms for ensuring the generalization ability of the trained models. In this article we concentrate

on the probabilistic multi-class classification of high-dimensional data in the context of Bayesian framework. The proposed adaptive learning machine typically utilizes dramatically fewer kernel functions than a comparable support vector machine while offering a number of additional advantages. These include the benefits of a resolution of computational cost, probabilistic predictions, a natural treatment of multi-class classification and an automatic model selection scheme.

As demonstrated by the some illustrative examples of its application along with some comparative benchmarks, the proposed adaptive learning machine yields stable prediction results. We use the genetic algorithm for searching the optimal model. However, this algorithm does not ensure that the determined model is the best model in model candidate space. Hence, the development of a better algorithm is one of the important problems and we would like to discuss it in a future paper.

Acknowledgement

The authors would like to thank the anonymous reviewer for their helpful comments and suggestions.

References

Ando, T., Simauchi, J. and Konishi, S. (2002). Nonlinear pattern recognition using radial basis function networks (in Japanese). *Jap. J. Appl. Statist.*, **31**, 123-139.

Breiman, L., Friedman, J. H., Olshen, R. and Stone, C. J. (1984). *Classification and regression trees.* Wadsworth, Belmont, California.

Cortes, V. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, **20**, 273-297.

Davison, A. C. (1986). Approximate predictive likelihood. *Biometrika*, **73** , 323-332.

Goldberg, D. E. (1989). *Genetic algorithm in search, optimization and machine learning.* Addison - Wesley.

Hastie, T. and Tibshirani, R. (1990). *Generalized additive models.* Chapman & Hall/CRC, London.

Hastie, T., Tibshirani, R. and Buja. A. (1994). Flexible discriminant analysis by optimal scoring. *J. Am. Statist. Assoc.*, **89**, 1255-1270.

Holland, J. H. (1975). *Adaptation in neural and artificial systems.* University of Michigan Press, Ann Arbor.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, **33**, 82-95.

Konishi, S., Ando, T. and Imoto, S. (2004). Bayesian information criteria and smoothing parameter selection in radial basis function networks. *Biometrika*, **91**, 27-43.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R. and Klein, B. (2000). Smoothing spline ANOVA models for large data sets with Bernoulli observations and randomized GACV. *Ann. Statist.*, **28**, 1570-1600.

Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proceeding of the IEEE*, **78**, 263-266.

Ripley, B. D. (1994). Neural networks and related methods for classification. *J. R. Statist. Soc.* B **56**, 409-456.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist*, **6**, 461-464.

Smola, A. and Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. *Proceedings of the 17th International Conference on Machine Learning.* Morgan Kaufmann Publishers.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *J. Am. Statist. Assoc.*, **81**, 82-86.

Vapnik, V. (1998). *Statistical learning theory*. New York; Wiley.

Wahba, G., Lin, Y. and Zhang, H. (2000). Generalized approximate cross validation for support vector machines, or, another way to look at margin-like quantities. in *Advances in Large Margin Classifiers*, (Ed, Smola, A., Bartlett, P., Schölkopf, B. and Schurmans, C), MIT Press, 297-309.

Zhu, J. and Hastie, T. (2001). Kernel logistic regression and the import vector machine. *Proceedings of Neural Information Processing Systems 2001*, Vancouver.

Received October 11, 2003
Revised April 8, 2004