A Probabilistic Framework for SVM Regression and Error Bar Estimation

J.B. GAO*   jg@ecs.soton.ac.uk
S.R. GUNN   srg@ecs.soton.ac.uk
C.J. HARRIS   cjh@ecs.soton.ac.uk
Image, Speech and Intelligent System Research Group, Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

M. BROWN   m1brown@uk.ibm.com
Data Exploitation Group, D0170, HS&T, IBM Hursley Laboratory, Winchester SO21 2JN, UK

Editor: Nello Cristianini

Abstract. In this paper, we elaborate on the well-known relationship between Gaussian Processes (GP) and Support Vector Machines (SVM) under some convex assumptions for the loss functions. This paper concentrates on the derivation of the evidence and error bar approximation for regression problems. An error bar formula is derived based on the ϵ-insensitive loss function.

Keywords: support vector machine (SVM), Gaussian process, ϵ-loss function, error bar estimation

1. Introduction

The foundation of Support Vector Machines (SVM) has been developed by Vapnik (1995) and has gained popularity due to its many attractive, analytic and computational features, and promising empirical performance. The formulation embodies the Structural Risk Minimization (SRM) principle, which has been shown (Gunn, Brown, & Bossley, 1997) to be superior to the Empirical Risk Minimization (ERM) principle employed by many conventional neural networks. SVMs were developed to solve the classification problem, in which it is shown that the generalization error is bounded by the sum of the training set error and a term depending on the VC (Vapnik-Chervonenkis) dimension of the model. Recently they have been extended to the domain of regression problems (Vapnik, 1995; Vapnik, 1998; Smola, 1998).

In the literature the terminology for SVMs can be slightly confusing. As proposed in Gunn (1998) and here, we use the term SVM to refer to both classification and regression methods, and the terms Support Vector Classification (SVC) and Support Vector Regression (SVR) to the specific problems of classification and regression respectively.

---

*This author is on leave from Department of Mathematics, Huazhong University of Science and Technology (HUST), Wuhan 430074, People's Republic of China.

SVCs are motivated by the geometric interpretation of maximizing the margin of discrimination, and are characterized by the use of a kernel function. It has been shown that SVM methodology can be cast as a variational/regularization problem in terms of a reproducing kernel Hilbert space (RKHS) (Wahba, 1990, 1999; Girosi, 1998; Poggio & Girosi, 1998), and hence SVMs and penalty methods, as used in the statistical theory of nonparametric regression, have a strong interrelationship (Evgeniou, Pontil, & Poggio, 1999).

However, like most penalty methods, some parameters in the SVM have to be "tuned" to suit a specific problem. One of the most important parameters is the regularisation parameter which is often chosen, a priori. A more disciplined approach uses a validation dataset or cross-validation, but this can be very computationally expensive.

Recently it has been shown (Sollich, 1999c) that SVC can be interpreted as a maximum a posterior solution to inference problems with Gaussian priors and an appropriate likelihood function based on a probabilistic interpretation. This interpretation enables Bayesian methods to be employed to determine the regularisation parameters in the SVM framework. In the last decade neural networks have been used to tackle regression and classification problems, with some notable successes. It has also been widely recognized that they form a part of a wide variety of nonlinear statistical techniques that can be used for these tasks. In these methods, Bayesian models which are based on Gaussian priors, both on parameter spaces and function spaces are becoming increasingly popular in the neural computation community (see MacKay, 1997; Neal, 1996; Williams, 1997, 1998). These ideas provide the possibility of a probabilistic interpretation of SVR.

In this paper, we introduce the probabilistic SVR model with an ϵ-insensitive loss function in Section 2. Section 3 proposes an approach using an evidence computation. Then in order to determine the regularisation parameter we show how MacKay's evidence framework (MacKay, 1992) can be used in the case of Gaussian SVR. We conclude the paper by deriving an error bar formula for the Gaussian SVR prediction. Finally some comparisons are made between different loss functions and the difficulties in dealing with the Huber's loss function are discussed.

2. Gaussian SVRs and ϵ-insensitive loss function

In regression estimation we try to estimate a functional dependency $a(\mathbf{x})$ between a set of sampled points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\ell\}$ taken from $\mathbb{R}^n$, and target values $Y = \{y_1, y_2, \ldots, y_\ell\}$ with $y_i \in \mathbb{R}$. Let us assume that these samples have been drawn independently from an unknown probability distribution $P(\mathbf{x}, y)$ and that there exists a Hilbert space of real-valued functions

$$\mathcal{F} = \{a(\mathbf{x}) | a : \mathbb{R}^n \longrightarrow \mathbb{R}\},$$

then the basic problem is to find a function $a(\mathbf{x}) \in \mathcal{F}$ that minimizes a risk functional

$$R[a] = \int l(y - a(\mathbf{x})) dP(\mathbf{x}, y)$$

where l is a loss function used to measure the deviations between the estimate and the target value. Specializing $l(y - a(\mathbf{x})) = (y - a(\mathbf{x}))^2$ leads to the definition of the usual least mean square error risk.

As the probability density function $P(\mathbf{x}, y)$ is unknown, we cannot evaluate (and hence minimize) $R[a]$ directly. Instead we only can try to approximate the minimum of $R[a]$ by some function $\hat{f}$, using the given datasets of X and Y. In practice, this requires consideration of the empirical risk functional $R_{\text{emp}}[a]$, which is obtained by replacing the integrals over the probability density function $P(\mathbf{x}, y)$ with summations over the empirical data:

$$R_{\text{emp}}[a] = \frac{1}{N} \sum_{\mathbf{x}_i \in X} l(y_i - a(\mathbf{x}_i)).$$

In general, this is an ill-posed problem in Tikhonov's sense (Tikhonov & Arsenin, 1977) resulting in poor generalization. Therefore it is not advisable to minimize the empirical risk without any means of structural control or regularization. Hence the SRM principle is preferable in practice. The standard regularization term in the spirit of Tikhonov and Arsenin (1977) is a positive semidefinite operator $\hat{P}$ mapping $\mathcal{F}$ into a dot-product (feature) space F by which a regularized risk functional

$$R_{\text{reg}}[a] = R_{\text{emp}}[a] + \frac{\lambda}{2} \|\hat{P}a\|^2$$

can be used with a regularization parameter $\lambda \geq 0$. This additional term effectively reduces the model space and thereby controls the complexity of the solution. In this paper we will consider an equivalent form defined as follows

$$R_{\text{reg}}[a] = C \sum_{\mathbf{x}_i \in X} l(y_i - a(\mathbf{x}_i)) + \frac{1}{2} \|\hat{P}a\|^2 \tag{2.1}$$

In the above techniques one of the open questions remaining is how to determine the best regularization parameter λ (or equivalently the parameter C in (2.1)). One method can use the model selection criteria such as VC-theory (Vapnik, 1995), Bayesian methods (MacKay, 1991), AIC (Akaike, 1974) and NIC (Murata, Yoshizawa, & Amari, 1994) etc.

Recently the so-called Gaussian Process have been introduced into classification and regression problems by setting the regulariser $\hat{P}$ in (2.1) equal to a Gaussian Process (GP) see (Williams, 1998) and therein. In fact, if (2.1) is regarded as a negative posterior probability (see the following discussion), then the second term corresponds automatically to a GP with kernel $K = (\hat{P}^T \hat{P})^{-1}$. In general, let us define a vector of function values $\mathbf{a}(X)$ as:

$$\mathbf{a}(X) = [a(\mathbf{x}_1), a(\mathbf{x}_2), \ldots, a(\mathbf{x}_\ell)]^T$$

Then the conditional probability of $\mathbf{a}(X)$ given a training dataset $\mathcal{D} = \{X, Y\}$ is denoted by $P[\mathbf{a}(X) \mid \mathcal{D}]$, and the conditional probability (or likelihood) of $\mathcal{D}$ given $\mathbf{a}(X)$ is then denoted by $P[\mathcal{D} \mid \mathbf{a}(X)]$. If the function underlying the data is $a(\mathbf{x})$, then $P[\mathcal{D} \mid \mathbf{a}(X)]$ is the likelihood

probability that, by random sampling the function $a(\mathbf{x})$ at the input X, the measurement Y is obtained, and can therefore be considered as a noise distribution for the additive model. $P[\mathbf{a}(X)]$ is the *a priori* probability of the unknown function $a(\mathbf{x})$ at X. This embodies the *a priori* knowledge of the function, and can be used to impose constraints on the model, assigning significant probability only to those functions which satisfy these constraints.

Assuming that the probability distributions $P[\mathcal{D} \mid \mathbf{a}(X)]$ and $P[\mathbf{a}(X)]$ are known, the *a posteriori* distribution $P[\mathbf{a}(X) \mid \mathcal{D}]$ can now be computed by applying the Bayesian Rule as:

$$P[\mathbf{a}(X) \mid \mathcal{D}] = \frac{P[\mathcal{D} \mid \mathbf{a}(X)] P[\mathbf{a}(X)]}{P[\mathcal{D}]} \tag{2.2}$$

where $P[\mathcal{D}]$ is called the evidence.

We now make the assumption that the data, $\mathcal{D}$, have been generated i.i.d. according to the following model

$$P(\mathbf{x}, y) = P(y \mid \mathbf{x}) P(\mathbf{x}) = P(y - a(\mathbf{x})) P(\mathbf{x})$$

with $P(y - a(\mathbf{x})) \propto \exp\{-Cl(y - a(\mathbf{x}))\}$. Therefore the probabilistic distribution $P[\mathcal{D} \mid \mathbf{a}(X)]$ can be written as:

$$P[\mathcal{D} \mid \mathbf{a}(X)] = K(C) \exp\left\{-C \sum_{\mathbf{x}_i \in X} l(y_i - a(\mathbf{x}_i))\right\}$$

where the loss function l is used to measure and penalize noise and $K(C)$ is a corresponding normalization constant.
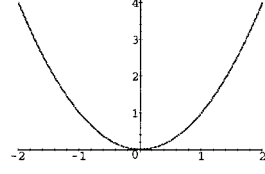
There exist a large number of loss functions which could be utilized. Figure 1 illustrates four possible loss functions. The loss function in figure 1(a) corresponds to the conventional least squares error criterion which is optimal for a Gaussian noise density model. The loss function in figure 1(b) is the Laplacian loss function that is less sensitive to outliers than the quadratic loss function. Huber proposed the loss function in figure 1(c) as a robust loss function that has optimal properties when the underlying distribution of the data is unknown. These three loss functions produce little sparseness in the support vectors. To address this issue Vapnik proposed the famous ϵ-insensitive loss function shown in figure 1(d) as an approximation to Huber's loss function that enables a sparse set of support vectors to be obtained. In the following we concentrate on the ϵ-insensitive loss function, given by

$$L_\epsilon(u) = \begin{cases} 0 & \text{for } |u| < \epsilon, \\ |u| - \epsilon & \text{otherwise.} \end{cases}$$
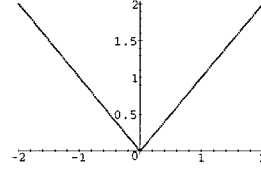
The model for the prior probability distribution $P[\mathbf{a}(X)]$ corresponding to the regularisation model (2.1) is chosen as

$$P[\mathbf{a}(X)] \propto \exp\left\{-\frac{1}{2} \|\hat{P}a\|^2\right\}.$$
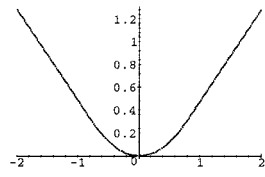
This form of probability distribution gives high probability only to those functions for which the regularisation term $\hat{P}a$ is small.

(a) Square Loss Function
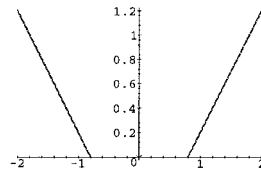
(b) 1-Norm Loss Function

(c) Huber's Loss Function

(d) ϵ-Insensitive Loss Function

Figure 1. Four typical loss functions.

The probabilistic interpretation of SVRs can be regarded as defining the loss function and in the case of the ϵ-insensitive loss function resulting in a likelihood through,

$$P[\mathcal{D} \mid \mathbf{a}(X)] = \left[\frac{1}{2} \frac{C}{\epsilon C + 1} \right]^N \exp\left\{ -C \sum_{\mathbf{x}_i \in X} L_\epsilon(y_i - a(\mathbf{x}_i)) \right\} \qquad (2.3)$$

with the prior probability distribution $P[\mathbf{a}(X)]$ as a functional Gaussian process in the above framework. A Gaussian process is defined as a stochastic process specified by giving only the mean vector and covariance matrix for any finite subset of points. We specify the prior probability distribution $P[\mathbf{a}(X)]$ as a Gaussian process with a zero mean and a covariance function $K(\mathbf{x}, \mathbf{x}')$, i.e.,

$$P[\mathbf{a}(X)] = \frac{1}{\sqrt{\det 2\pi K_{X,X}}} \exp\left\{ -\frac{1}{2} \mathbf{a}(X)^T K_{X,X}^{-1} \mathbf{a}(X) \right\}. \qquad (2.4)$$

where $K_{X,X} = [K(\mathbf{x}_i, \mathbf{x}_j)]$ is the covariance matrix at the points X.

Following the Bayesian Rule (2.2) the *a posteriori* probability of $\mathbf{a}(X)$ is written as

$$P[\mathbf{a}(X) \mid \mathcal{D}] = \frac{[G(C, \epsilon)]^N}{\sqrt{\det 2\pi K_{X,X}} P[\mathcal{D}]} \exp\left\{ -C \sum_{\mathbf{x}_i \in X} L_\epsilon(y_i - a(\mathbf{x}_i)) \right.$$
$$\left. -\frac{1}{2} \mathbf{a}(X)^T K_{X,X}^{-1} \mathbf{a}(X) \right\} \qquad (2.5)$$

where $G(C, \epsilon) = \frac{1}{2} \frac{C}{\epsilon C + 1}$.

A simple estimate of the function $a(\mathbf{x})$ from (2.5) is the so-called Maximum A Posterior (MAP) estimate which is the function that maximizes the *a posteriori* probability $P[\mathbf{a}(X)|\mathcal{D}]$, or minimizes the exponent in the Eq. (2.5). Thus the MAP solution of $a(\mathbf{x})$ is the minimizer of the following risk functional:

$$R_{\text{GSVM}}[a] = C \sum_{\mathbf{x}_i \in X} L_{\epsilon}(y_i - a(\mathbf{x}_i)) + \frac{1}{2}\mathbf{a}(X)^T K_{X,X}^{-1} \mathbf{a}(X) \tag{2.6}$$

We call the minimisation of (2.6) a Gaussian SVM problem for regression. It is easy to show that the Gaussian SVM is equivalent to the standard SVM problem with the penalty term $\frac{1}{2}\|\mathbf{w}\|^2$ where $\mathbf{w}$ is the weight vector in the feature space defined by kernel $K(\mathbf{x}, \mathbf{y})$ see eg. (Smola, 1998), and hence, $\frac{1}{2}\mathbf{a}(X)^T K_{X,X}^{-1} \mathbf{a}(X) = \frac{1}{2}\|\mathbf{w}\|^2$. The standard SVM algorithm finds the minimum $a^*(\mathbf{x})$ of $R_{\text{GSVM}}[a]$. Following the discussion in Girosi (1998) and Sollich (1999c) we can write $a^*(\mathbf{x})$ in the form

$$a^*(\mathbf{x}) = \sum_{\mathbf{x}_i \in X} \beta_i K(\mathbf{x}_i, \mathbf{x}) \tag{2.7}$$

where $\beta_i = \alpha_i^+ - \alpha_i^-$, and both α_i^+ and α_i^- can be determined by a QP-problem using the Wolfe's dual of original minimisation (Vapnik, 1995). The training dataset X can be divided into four parts with respect to the SVM solution $a^*(\mathbf{x})$,

$$X_0 = \{\mathbf{x}_i \mid |y_i - a^*(\mathbf{x}_i)| < \epsilon \quad \text{with } \alpha_i^+ = \alpha_i^- = 0\} \tag{2.8}$$
$$X_C = \{\mathbf{x}_i \mid |y_i - a^*(\mathbf{x}_i)| > \epsilon \quad \text{with } \alpha_i^+ = C, \alpha_i^- = 0$$
$$\text{or } \alpha_i^- = C, \alpha_i^+ = 0\} \tag{2.9}$$
$$X_{M^-} = \{\mathbf{x}_i \mid a^*(\mathbf{x}_i) - y_i - \epsilon = 0 \quad \text{with } 0 < \alpha_i^- < C\} \tag{2.10}$$
$$X_{M^+} = \{\mathbf{x}_i \mid y_i - a^*(\mathbf{x}_i) - \epsilon = 0 \quad \text{with } 0 < \alpha_i^+ < C\} \tag{2.11}$$

All points $\mathbf{x}_i$ in X_{M^+} or X_{M^-} are collected as X_M and are called the marginal vectors, i.e., the marginal vectors X_M are given by $X_M = X_{M^-} \cup X_{M^+}$. The support vectors, X_{SV}, are given by $X_{SV} = X_M \cup X_C$. Denote $\bar{X}_M = X \backslash X_M$. Also we should note that $\alpha_i^+ \alpha_i^- = 0$, i.e., α_i^+ and α_i^- cannot be simultaneously different from zero.

3. Evidence for SVR

3.1. Calculation for the evidence

The evidence $P[\mathcal{D}]$ in (2.2) is simply the likelihood of the data for a given model, obtained by integration over the model parameter space $\{a(\mathbf{x})\}$, i.e.,

$$P[\mathcal{D}] = \int P[\mathbf{a}(X)]P[\mathcal{D} \mid \mathbf{a}(X)] \, d\mathbf{a}(X) \tag{3.1}$$

Inserting (2.3) and (2.4) leads to an integral which is analytically intractable. In order to overcome this problem we use an approximation via a second order Taylor's expansion around the obtained SVR solution $a^*(\mathbf{x})$. First denote $\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))$ the (left/right) derivative of L_ϵ at the $y_i - a^*(\mathbf{x}_i)$. Thus for the support vector $\mathbf{x}_i \in X_{M^-}$, $y_i - a^*(\mathbf{x}_i) = -\epsilon$, then

$$\delta a(\mathbf{x}_i) = a(\mathbf{x}_i) - a^*(\mathbf{x}_i) = y_i - a^*(\mathbf{x}_i) - (y_i - a(\mathbf{x}_i)) = -\epsilon - (y_i - a(\mathbf{x}_i))$$

Thus

$$\delta a(\mathbf{x}_i) > 0 \iff -\epsilon > y_i - a(\mathbf{x}_i) \implies \partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i)) = -1$$
$$\delta a(\mathbf{x}_i) < 0 \iff -\epsilon < y_i - a(\mathbf{x}_i) \implies \partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i)) = 0$$

And for the support vector $\mathbf{x}_i \in X_{M^+}$, $y_i - a^*(\mathbf{x}_i) = \epsilon$, then

$$\delta a(\mathbf{x}_i) = a(\mathbf{x}_i) - a^*(\mathbf{x}_i) = y_i - a^*(\mathbf{x}_i) - (y_i - a(\mathbf{x}_i)) = \epsilon - (y_i - a(\mathbf{x}_i)).$$

Hence

$$\delta a(\mathbf{x}_i) > 0 \iff \epsilon > y_i - a(\mathbf{x}_i) \implies \partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i)) = 0$$
$$\delta a(\mathbf{x}_i) < 0 \iff \epsilon < y_i - a(\mathbf{x}_i) \implies \partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i)) = 1$$

Let

$$F(\mathbf{a}(X)) = \ln P[\mathbf{a}(X)] P[\mathcal{D} \mid \mathbf{a}(X)], \tag{3.2}$$

and define $\delta \mathbf{a}(X) = \mathbf{a}(X) - \mathbf{a}^*(X)$, then

$$
\begin{aligned}
F(\mathbf{a}(X)) =\ & F(\mathbf{a}^*(X)) + \partial_1 F(\mathbf{a}^*(X))^T \delta \mathbf{a}(X) + \frac{1}{2} \delta \mathbf{a}(X)^T \partial_2 F(\mathbf{a}^*(X)) \delta \mathbf{a}(X) \\
=\ & F(\mathbf{a}^*(X)) - \mathbf{a}^*(X)^T K_{X,X}^{-1} \delta \mathbf{a}(X) \\
& + C \sum_{\mathbf{x}_i \in X} \partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i)) \delta a(\mathbf{x}_i) - \frac{1}{2} \delta \mathbf{a}(X)^T K_{X,X}^{-1} \delta \mathbf{a}(X)
\end{aligned}
$$

From Eq. (2.7) and the MAP property that the linear terms in $\delta a(\mathbf{x}_i)$ are zero for all input points $\mathbf{x}_i \notin X_M$, we obtain,

$$
\begin{aligned}
F(\mathbf{a}(X)) =\ & F(\mathbf{a}^*(X)) - \sum_{\mathbf{x}_i \in X_M} [\beta_i - C \partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))] \delta a(\mathbf{x}_i) \\
& - \frac{1}{2} \delta \mathbf{a}(X)^T K_{X,X}^{-1} \delta \mathbf{a}(X) \tag{3.3}
\end{aligned}
$$

Inserting (3.2) and (3.3) into (3.1) results in

$$P[\mathcal{D}] = \int \exp\Bigg\{ F(\mathbf{a}^*(X)) - \sum_{\mathbf{x}_i \in X_M} [\beta_i - C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))]\delta a(\mathbf{x}_i)$$
$$- \frac{1}{2}\delta\mathbf{a}(X)^T K_{X,X}^{-1} \delta\mathbf{a}(X) \Bigg\} d\delta\mathbf{a}(X)$$

By marginalizing over the variables $\bar{X}_M$, the above integral can be computed as:

$$P[\mathcal{D}] = \exp\{F(\mathbf{a}^*(X))\} \int \exp\Bigg\{ - \sum_{\mathbf{x}_i \in X_M} [\beta_i - C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))]\delta a(\mathbf{x}_i)$$
$$- \frac{1}{2}\delta\mathbf{a}(X_M)^T \big[K_{X,X}^{-1}\big]\Big|_{X_M} \delta\mathbf{a}(X_M) \Bigg\} d\delta\mathbf{a}(X_M) \Bigg/ \sqrt{\det\big(2\pi\big[K_{X,X}^{-1}\big]\big|_{X_M}^{-1}\big)} \quad (3.4)$$

where $[K_{X,X}^{-1}]|_{X_M}$ is the sub-block matrix of $K_{X,X}^{-1}$ with respect to X_M. Discarding the second order terms and considering the integrals with respect to the linear term δa: For $\mathbf{x}_i \in X_{M^-}$, we have

$$\int_{-\infty}^{\infty} \exp\{-\beta_i \delta a(\mathbf{x}_i) + C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))\delta a(\mathbf{x}_i)\} d\delta a(\mathbf{x}_i)$$
$$= \int_{-\infty}^{0} \exp\{-\beta_i \delta a(\mathbf{x}_i)\} d\delta a(\mathbf{x}_i) + \int_{0}^{\infty} \exp\{-\beta_i \delta a(\mathbf{x}_i) - C\delta a(\mathbf{x}_i)\} d\delta a(\mathbf{x}_i)$$
$$= \int_{-\infty}^{0} \exp\{\alpha_i^- \delta a(\mathbf{x}_i)\} d\delta a(\mathbf{x}_i) + \int_{0}^{\infty} \exp\{-(C - \alpha_i^-)\delta a(\mathbf{x}_i)\} d\delta a(\mathbf{x}_i)$$
$$= \frac{1}{\alpha_i^-} + \frac{1}{C - \alpha_i^-} = \frac{C}{\alpha_i^-(C - \alpha_i^-)}$$

Similarly, for $\mathbf{x}_i \in X_{M^+}$ we have,

$$\int_{-\infty}^{\infty} \exp\{-\beta_i \delta a(\mathbf{x}_i) + C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))\delta a(\mathbf{x}_i)\} d\delta a(\mathbf{x}_i) = -\frac{C}{\alpha_i^+(C - \alpha_i^+)}$$

Then the final evidence is

$$\ln P(\mathcal{D}) \approx -\frac{1}{2}(\beta)^T\big|_{X_{SV}} K_{X_{SV}, X_{SV}} \beta\big|_{X_{SV}} - \frac{1}{2}\ln\det\big(2\pi K_{X_M}\big) + \ell \ln G(C, \epsilon)$$
$$- C \sum_{\mathbf{x}_i \in X_C} L_\epsilon(y_i - a^*(\mathbf{x}_i)) + \sum_{\mathbf{x}_i \in X_M} \ln \frac{C}{|\beta_i|(C - |\beta_i|)} \quad (3.5)$$

It is also important to note that Sollich has presented this kind of argument for the case of SVC in Sollich (1999a, 1999b, 2000). From (3.4) a more accurate approximation can be

given

$$P[\mathcal{D}] = \exp\{F(\mathbf{a}^*(X))\} \int \exp\bigg\{ - \sum_{\mathbf{x}_i \in X_M} [\beta_i - C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))]\delta a(\mathbf{x}_i)$$

$$- \frac{1}{2} \sum_{\mathbf{x}_i \in X_M} \delta a(\mathbf{x}_i)[K_{X,X}^{-1}]\big|_{X_M}(i,i)\delta a(\mathbf{x}_i)\bigg\} d\delta\mathbf{a}(X_M) \bigg/ \sqrt{\det\big(2\pi[K_{X,X}^{-1}]\big|_{X_M}^{-1}\big)}$$

and each integral can be carried out by noting that the integral of the normal distribution is an error function.

3.2. Hyperparameter estimation

So far we have assumed that the values of the regularisation parameter C, or the hyperparameter in the model, are known. However, for most applications, we will have little idea of a suitable value for C. In general, the normal Bayesian treatment for hyperparameters such as C, whose value is unknown, is to integrate them out of any predictions. However this treatment will result in an intractable integral problem on the parameter C. An alternative approach, known as the evidence approximation, has been discussed by MacKay (1992) and has recently been utilized within a Gaussian process technique (see Williams, 1998) and therein. The core of this approach is to find the maximum C^* of the evidence $P[\mathcal{D}]$ or equivalently maximize the log evidence in (3.5).

By differentiating (3.5) with respect to C and setting to zero we get

$$C^{\text{new}} = \frac{[\ell + |X_M|]}{\sum\limits_{\mathbf{x}_i \in X_C} L_\epsilon(y_i - a^*(\mathbf{x}_i)) + \sum\limits_{\mathbf{x}_i \in X_M} \frac{1}{C - |\beta_i|} + \frac{\ell\epsilon}{\epsilon C + 1}} \tag{3.6}$$

The current estimate of C is used to evaluate the quantities on the righthand side of (3.6), and the procedure is started by making some initial guess for the value of C. Each new estimate C^{new} is then used to train the SVM algorithm to determine all of the α's.

Let us take a simple example to demonstrate the application of Eq. (3.6). A given data set was generated by the simple model, $y = \sin x$, with the target corrupted by additive Gaussian noise with variance 0.5. The standard SVM algorithm is implemented with the RBF kernel function with width $\sigma^2 = 0.8$, the ϵ-loss function of $\epsilon = 0.4$. We take an initial value of $C = 10$ in (3.6) and then start the iterative procedure. In order to stop the iterative procedure we employ the criteria that the two successive updated values of C satisfy $|C_1 - C_2| < 0.05$. Table 1 lists the iterative values of C.

Table 1. The value of hyperparameter C.

Step	1	2	3	4	5	6	7	8	9	10	11	12	⋯ ⋯
C	10	3.4276	2.2356	2.0308	2.2081	2.1078	2.1951	2.1252	2.1883	2.1339	2.1828	2.1405	⋯ ⋯

From Table 1 we can choose $C = 2.1$ as an approximate value to the true value of C. However, the iterative update formula (3.6) is not globally stable, i.e., if the initial value of C is far from its "true" value, then the iterative procedure may not converge. In general, (3.6) will not find a stationary point of (3.5) with respect to C due to the SVM solution changing with C. In order to implement the evidence framework, a gradient-based search can be employed to search the maximum of (3.5) based on the derivative of the log evidence with respect to C. For simplicity, we take an auxiliary algorithm to estimate a better initial value for the hyperparameter C based on Eq. (3.5). Like the demonstration method employed by Sollich in Sollich (1999b), the log-evidence, $\ln P(\mathcal{D})$, can be evaluated as a function of the hyperparameter C. Thus the curve of this function versus C can be plotted by taking different values of C. However, it should be pointed out that the approximation (3.5) has logarithmic singularities when, as C is varied, the β_i for one of the marginal inputs approaches either 0 or C. That means that (3.5) is not actually a smooth function of C. In order to get a smooth curve we take an average over 50 different dataset of the same size as the above example where these singularities are smoothed out. The curve in figure 2 is plotted based on those different datasets. This curve just reflects the approximated position of the "true" value of C. From figure 2 we can find an initial guess for the value of C near the "true" value, e.g., $C_0 = 2.1$, and then use the update formula (3.6) for a more precise value of C. It is worthy to mention that the evidence curve can be used as an estimation method for C (Sollich, 1999b).

In order to confirm that the estimated value $C = 2.1$ is reliable, let us investigate the relationship between the hyperparameter C and the corresponding average generalisation error,

$$E(y(x) - a^*(x)) = \int L_\epsilon(y(x) - a^*(x))P(x)\,dx$$

where $P(x)$ is the density distribution of input x, assumed to be uniform. It is obvious that E is a function of C. The "true" value C_{true} of C should be the minima of the function E.
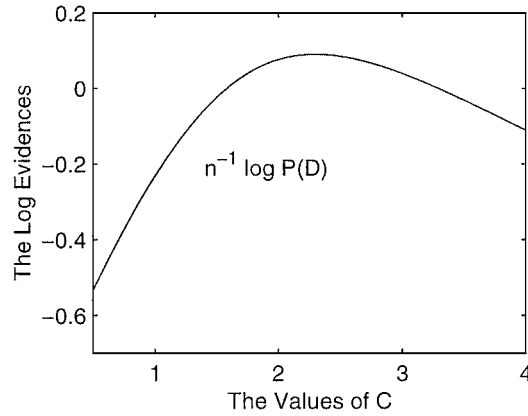

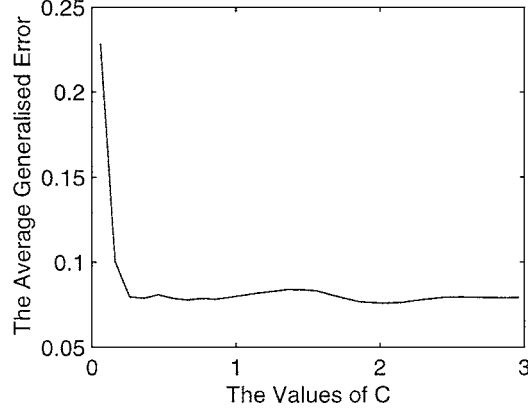*Figure 2.* The relationship between C and the evidence: $C = 2.1$?

Figure 3. The relationship between *C*'s and the average generalised errors.

For the above example, the curve of this function is plotted in figure 3 from which it can be found that the possible "true" value of *C* is near to 2.

4. An approximated formula for error bar of SVR

Next we consider the problem of estimating the prediction error for the Gaussian SVR model. When given a prediction, it is also very useful to be given some idea of the error bars associated with that prediction. Error bars arise naturally in a Bayesian treatment of learning machines and are made up of two terms, one due to the *a posterior* uncertainty (the uncertainty of function $a(\mathbf{z})$), and the other due to the intrinsic noise in the data.

4.1. The variance due to the function uncertainty

First of all, let us focus on the computation of the variance due to the function uncertainty.

Assume that $\mathbf{z}$ is a test example. Then first calculate the predictive distribution $a(\mathbf{z})$ corresponding to $\mathbf{z}$. This can be obtained by using Bayes rule:

$$P(a(\mathbf{z}) \mid \mathcal{D}) = \frac{1}{P(\mathcal{D})} \int P(\mathcal{D} \mid \mathbf{a}(X)) P(\mathbf{a}(X), a(\mathbf{z})) d\mathbf{a}(X) \tag{4.1}$$

Denote the covariance matrix with respect to the training data set $X = X_M \cup \bar{X}_M$ as follows

$$K_{X,X} = \begin{pmatrix} K_{X_M,X_M} & K_{X_M,\bar{X}_M} \\ K_{X_M,\bar{X}_M}^T & K_{\bar{X}_M,\bar{X}_M} \end{pmatrix}$$

and the covariance matrix between X and a test point $\mathbf{z}$

$$K_{[X,\mathbf{z}],[X,\mathbf{z}]} = \begin{pmatrix} K_{X,X} & K_{X,\mathbf{z}} \\ K_{X,\mathbf{z}}^T & K_{\mathbf{z},\mathbf{z}} \end{pmatrix}$$

where $K_{X,\mathbf{z}} = (K_{X_M,\mathbf{z}}^T, K_{\bar{X}_M,\mathbf{z}}^T)^T$ is a column vector and $K_{\mathbf{z},\mathbf{z}} = K(\mathbf{z},\mathbf{z})$ is a scalar, being the value of the covariance function K at $\mathbf{z}$. Denote the inverse of $K_{[X,\mathbf{z}],[X,\mathbf{z}]}$ as follows:

$$K_{[X,\mathbf{z}],[X,\mathbf{z}]}^{-1} = \begin{pmatrix} G_{X,X} & G_{X,\mathbf{z}} \\ G_{X,\mathbf{z}}^T & G_{\mathbf{z},\mathbf{z}} \end{pmatrix} \quad \text{and} \quad G_{X,X} = \begin{pmatrix} G_{X_M,X_M} & G_{X_M,\bar{X}_M} \\ G_{X_M,\bar{X}_M}^T & G_{\bar{X}_M,\bar{X}_M} \end{pmatrix} \tag{4.2}$$

Considering the formula for the predictive distribution (4.1), and using the above notation, the log integrand can be expressed as,

$$F(\mathbf{a}(X), a(\mathbf{z})) = -\frac{1}{2}\mathbf{a}(X)^T G_{X,X}\mathbf{a}(X) - a(\mathbf{z})G_{X,\mathbf{z}}^T\mathbf{a}(X) - \frac{1}{2}a(\mathbf{z})G_{\mathbf{z},\mathbf{z}}a(\mathbf{z})$$

$$- C\sum_{\mathbf{x}_i \in X} L_\epsilon(y_i - a(\mathbf{x}_i)) - \frac{1}{2}\ln\det\left(2\pi K_{[X,\mathbf{z}],[X,\mathbf{z}]}\right) + \ell\ln G(C,\epsilon).$$

Denoting $a^*(\mathbf{z}) = \sum_{i=1}^N \beta_i K(\mathbf{x}_i, \mathbf{z})$, then

$$a(\mathbf{z})G_{X,\mathbf{z}}^T\mathbf{a}^*(X) = -a(\mathbf{z})\left(K_{\mathbf{z},\mathbf{z}} - K_{X,\mathbf{z}}^T K_{X,X}^{-1} K_{X,\mathbf{z}}\right)^{-1} a^*(\mathbf{z}) = -a(\mathbf{z})G_{\mathbf{z},\mathbf{z}}a^*(\mathbf{z}).$$

Taking $a(\mathbf{z})$ to be fixed and expanding $F(\mathbf{a}(X), a(\mathbf{z}))$ with respect to $\mathbf{a}(X)$ at the optimal SVM solution, $\mathbf{a}^*(X)$, using a second order Taylor series expansion gives

$$F(\mathbf{a}(X), a(\mathbf{z})) \approx \tilde{F}(\mathbf{a}^*(X), a^*(\mathbf{z})) - \sum_{\mathbf{x}_i \in X_M} \{\beta_i - C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i))\}\delta a(\mathbf{x}_i)$$

$$-\frac{1}{2}\delta\mathbf{a}(X)^T G_{X,X}\delta\mathbf{a}(X) - \delta a(\mathbf{z})G_{X,\mathbf{z}}^T\delta\mathbf{a}(X)$$

$$-\frac{1}{2}\delta a(\mathbf{z})G_{\mathbf{z},\mathbf{z}}\delta a(\mathbf{z}). \tag{4.3}$$

Now let us denote $\delta\mathbf{a}(\bar{X}_M) = [\delta a(\mathbf{x}_i)]_{\mathbf{x}_i \in \bar{X}_M}^T$ and $G_{X,\mathbf{z}}^T = (G_{X_M,\mathbf{z}}^T, G_{\bar{X}_M,\mathbf{z}}^T)$, then, similar to the computation for the evidence, we have by taking the integration over $\delta\mathbf{a}(x)$,

$$P(a(\mathbf{z}) \,|\, \mathcal{D}) \propto \exp(\tilde{F}(\mathbf{a}^*(X)))\left[\prod_{\mathbf{x}_i \in X_{M^-}} \frac{C}{\alpha_i^-(C - \alpha_i^-)}\right]\left[\prod_{\mathbf{x}_i \in X_{M^+}} \frac{C}{\alpha_i^+(C - \alpha_i^+)}\right]$$

$$\times \sqrt{\det\left(2\pi G_{\bar{X}_M,\bar{X}_M}\right)}\exp\left[-a(\mathbf{z})G_{\mathbf{z},\mathbf{z}}\mathbf{a}^*(\mathbf{z}) - \frac{1}{2}a(\mathbf{z})G_{\mathbf{z},\mathbf{z}}a(\mathbf{z})\right]$$

$$\times \exp\left[+\frac{1}{2}\delta a(\mathbf{z})G_{\bar{X}_M,\mathbf{z}}^T\left(G_{\bar{X}_M,\bar{X}_M}\right)^{-1}G_{\bar{X}_M,\mathbf{z}}\delta a(\mathbf{z})\right].$$

It is easy to check that

$$G_{\mathbf{z},\mathbf{z}} - G_{\bar{X}_M,\mathbf{z}}^T \left(G_{\bar{X}_M,\bar{X}_M}\right)^{-1} G_{\bar{X}_M,\mathbf{z}} = \left(K_{\mathbf{z},\mathbf{z}} - K_{X_M,\mathbf{z}}^T K_{X_M,X_M}^{-1} K_{X_M,\mathbf{z}}\right)^{-1}.$$

Thus

$$P(a(\mathbf{z}) \mid \mathcal{D}) \propto \exp\left\{ -\frac{1}{2}(a(\mathbf{z}) - a^*(\mathbf{z}))\left(K_{\mathbf{z},\mathbf{z}} - K_{X_M,\mathbf{z}}^T K_{X_M,X_M}^{-1} K_{X_M,\mathbf{z}}\right)^{-1} \right.$$
$$\left. \times \, (a(\mathbf{z}) - a^*(\mathbf{z})) \right\} \tag{4.4}$$

The Eq. (4.4) means that $P(a(\mathbf{z}) \mid \mathcal{D})$ can be approximated by a Gaussian with mean $a^*(\mathbf{z})$ and variance

$$\sigma_{K_M}^2(\mathbf{z}) = K_{\mathbf{z},\mathbf{z}} - K_{X_M,\mathbf{z}}^T K_{X_M,X_M}^{-1} K_{X_M,\mathbf{z}} \tag{4.5}$$

In order to compare the estimate (4.5) with the true one an approximating bound for the variance can be derived. First by the property of marginal points $\mathbf{x}_i \in X_M$, see subsection 3.1, it is easy to prove that

$$\beta_i - C\partial_1 L_\epsilon(y_i - a^*(\mathbf{x}_i)) \begin{cases} > 0 & \text{when } \delta a(\mathbf{x}_i) > 0, \\ < 0 & \text{when } \delta a(\mathbf{x}_i) < 0. \end{cases}$$

Then the exponential of (4.3) can be upper bounded by

$$\exp\left\{ \tilde{F}(\mathbf{a}^*(X), a^*(\mathbf{z})) - \frac{1}{2}\delta\mathbf{a}(X)^T G_{X,X}\delta\mathbf{a}(X) - \delta a(\mathbf{z})G_{X,\mathbf{z}}^T \delta\mathbf{a}(X) \right.$$
$$\left. -\frac{1}{2}\delta a(\mathbf{z})G_{\mathbf{z},\mathbf{z}}\delta a(\mathbf{z}) \right\}$$

Denoting by $\sigma_K^2(\mathbf{z}) = K_{\mathbf{z},\mathbf{z}} - K_{X,\mathbf{z}}^T K^{-1} K_{X,\mathbf{z}}$ and integrating out $\delta\mathbf{a}(X)$ results in

$$P(a(\mathbf{z})|\mathcal{D}) \leq \frac{1}{\sqrt{2\pi\sigma_K^2(\mathbf{z})}} \exp\left\{ -\frac{(a(\mathbf{z}) - a^*(\mathbf{z}))^2}{2\sigma_K^2(\mathbf{z})} \right\}.$$

Thus the variance of $a(\mathbf{z})$ can be upper bounded by $\sigma_K^2(\mathbf{z})$, i.e.,

$$\text{var}(a(\mathbf{z})) \leq \sigma_K^2(\mathbf{z}). \tag{4.6}$$

4.2. *The variance for the prediction*

Consider the prediction model for a new test data point $\mathbf{z}$

$$t = a(\mathbf{z}) + e(\mathbf{z}) \tag{4.7}$$

where $e(\mathbf{z})$ is random noise independent of $a(\mathbf{z})$. We want to find the variance for the target on a test data point $\mathbf{z}$. Let us first note that the likelihood of SVR case is described by $\exp\{-L_\epsilon(t - a(\mathbf{z}))\}$ as shown in (2.3) for the training dataset. Recently Evgeniou, Pontil, and Poggio (1999) proved that

$$\exp(-L_\epsilon(t - a(\mathbf{z})))$$
$$= \frac{2(\epsilon + 1)}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \int_0^{+\infty} \lambda(u)\mu(\beta)\sqrt{\beta} \exp(-\beta(t - a(\mathbf{z}) - u)^2) du d\beta, \qquad (4.8)$$

with

$$\lambda(u) = \frac{1}{2(\epsilon + 1)} \left(\chi_{[-\epsilon,\epsilon]}(u) + \delta(u - \epsilon) + \delta(u + \epsilon) \right), \qquad (4.9)$$

$$\mu(\beta) = \frac{1}{4}\beta^{-2} \exp\left(-\frac{1}{4\beta}\right). \qquad (4.10)$$

where $\chi_{[-\epsilon,\epsilon]}(u)$ is 1 for $u \in [\epsilon, \epsilon]$, 0 otherwise.

Equation (4.8) means that the underlying noise model $e(\mathbf{z})$ described by the ϵ-insensitive loss function can be represented by a superposition of Gaussian processes with differing variances, $1/\beta$, with distribution (4.9) and differing means, u, with distribution (4.10). The variables u and β can be viewed as hidden variables with respect to the priors (4.9) and (4.10), respectively. This viewpoint has been used to derive a variational approach for the SVR problem in our recent paper (Gao et al., 2000). The advantage of (4.8) is that one can convert the uncertainty analysis, such as the error bar problem of SVR, into a mixture of Gaussian distributions.

Based on the prediction model (4.7) the posterior distribution for the target prediction t takes the following form

$$P(t \mid \mathcal{D}) \propto \int G(C) \exp\{-CL_\epsilon(t - a(\mathbf{z}))\} P(a(\mathbf{z}) \mid \mathcal{D}) \, da(\mathbf{z})$$

By (4.4) the posterior prediction function $a(\mathbf{z})$ at an input point $\mathbf{z}$ is distributed by

$$P(a(\mathbf{z}) \mid \mathcal{D}) = \frac{1}{\sqrt{2\pi\sigma_K^2(\mathbf{z})}} \exp\left\{ -\frac{1}{2\sigma_K^2(\mathbf{z})} (a(\mathbf{z}) - a^*(\mathbf{z}))^2 \right\}$$

Then the target prediction distribution can be expressed as

$$P(t \mid \mathcal{D}) = \frac{G(C)}{\sqrt{2\pi\sigma_{K_M}^2}} \int_{-\infty}^{+\infty} \exp\{-CL_\epsilon(t - a^*(\mathbf{z}) - \delta a(\mathbf{z}))\}$$
$$\times \exp\left\{ -\frac{\delta a(\mathbf{z})^2}{2\sigma_{K_M}^2(\mathbf{z})} \right\} d\delta a(\mathbf{z})$$

where $\delta a(\mathbf{z}) = a(\mathbf{z}) - a^*(\mathbf{z})$ and $G(C)$ is a normalization constant. In the following, the constant $G(C)$ may be distinct in the different equalities. Thus in terms of (4.8) we obtain:

$$P(t \mid \mathcal{D}) = \frac{G(C)}{\sqrt{2\pi\sigma_K^2(\mathbf{z})}} \int_{-\infty}^{+\infty} \exp\{-L_{\epsilon C}(Ct - Ca^*(\mathbf{z}) - C\delta a(\mathbf{z}))\}$$

$$\times \exp\left\{-\frac{\delta a(\mathbf{z})^2}{2\sigma_{K_M}^2(\mathbf{z})}\right\} d\delta a(\mathbf{z})$$

$$= \frac{G(C)\sqrt{\pi}}{C} \int_{-\infty}^{+\infty} \int_0^{+\infty} \lambda_C(u)\mu(\beta) \frac{\exp\left\{-\frac{(t-a^*(\mathbf{z})-\frac{u}{C})^2}{2\left(\sigma_{K_M}^2(\mathbf{z})+\frac{1}{2\beta C^2}\right)}\right\}}{\sqrt{2\pi}\sqrt{\sigma_{K_M}^2(\mathbf{z})+\frac{1}{2\beta C^2}}} du\,d\beta$$

where $\lambda_C(u)$ is the C scaled version of $\lambda(u)$. To simplify denote $\sigma_t^2(\mathbf{z}, \beta) = \sigma_{K_M}^2(\mathbf{z}) + \frac{1}{2\beta C^2}$, then the normalized $P(t \mid \mathcal{D})$ is given by

$$P(t \mid \mathcal{D}) = \int_{-\infty}^{+\infty} \int_0^{+\infty} \lambda_C(u)\mu(\beta) \frac{\exp\left\{-\frac{(t-a^*(\mathbf{z})-\frac{u}{C})^2}{2\sigma_t^2(\mathbf{z},\beta)}\right\}}{\sqrt{2\pi}\sigma_t^2(\mathbf{z}, \beta)} du\,d\beta \qquad (4.11)$$

Thus all of uncertainty analysis for the target, t, can be carried out based on the above distribution. In the following, we will propose a simple derivation of error bar formula.

From (4.7), it follows directly that the mean and variance of t are the sum of the means and variance of $a(\mathbf{z})$ and $e(\mathbf{z})$ because $a(\mathbf{z})$ and $e(\mathbf{z})$ can be considered as independent random variables. Now the probability distribution of $e(\mathbf{z})$ is $P(e(\mathbf{z})) \propto \exp\{-CL_\epsilon(e(\mathbf{z}))\}$, so by the symmetry of $E(e(\mathbf{z})) = 0$, we can obtain

$$E[t \mid \mathcal{D}] = a^*(\mathbf{z}).$$

On the other hand, it is easy to prove that

$$\mathrm{var}(e(\mathbf{z})) = E[e^2(\mathbf{z})] = \frac{2}{C^2} + \frac{\epsilon^2(\epsilon C + 3)}{3(\epsilon C + 1)}. \qquad (4.12)$$

Then by Eqs. (4.5) and (4.12) we have

$$\begin{aligned}\sigma_t^2(\mathbf{z}) &= \sigma_{K_M}^2(\mathbf{z}) + \mathrm{var}(e(\mathbf{z})) \\ &= \sigma_{K_M}^2(\mathbf{z}) + \frac{2}{C^2} + \frac{\epsilon^2(3 + \epsilon C)}{3(\epsilon C + 1)} = \sigma_{K_M}^2(\mathbf{z}) + \sigma_{C(\epsilon)}^2\end{aligned} \qquad (4.13)$$

Thus Eq. (4.11) provides an approximation to the target predictive distribution with the mean $m = a^*(\mathbf{z})$ and the variance (4.13). Then it is easy to obtain an estimate of the uncertainty (or "error bar") about the predicted mean $a^*(\mathbf{z})$. This error bar has two components, see Eq. (4.13). The first σ_K^2 is an estimate of the width of the posterior over the function $a(\mathbf{z})$

and reflects the uncertainty induced in the function given the finite amount of data available. The second term can be viewed as the measure for the uncertainty induced in the target value t determined by the hyperparameters C and ϵ. By (4.6) the variance of target t can be upper bounded as

$$\sigma_t^2(\mathbf{z}) \leq \sigma_K^2(\mathbf{z}) + \sigma_{C(\epsilon)}^2 \qquad (4.14)$$

4.3. Simulation

Figure 4 shows a simple application of the error bar estimation (4.13). Artificial data is generated by a simple function

$$y = \sin x$$

which is corrupted by Gaussian noise with variance 0.5. The standard SVR algorithm is implemented with an RBF kernel function $K(x, y) = \exp\{-\frac{1}{2\sigma^2}(x - y)^2\}$ with width $\sigma^2 = 0.8$, the ϵ-loss function of $\epsilon = 0.4$ and $C = 2.1$. The $C = 2.1$ is a possible true value of ϵ-insensitive noise parameter which can be obtained by the maximum evidence method as shown in Section 3. Figure 4(a) shows the result of the SVR algorithm based on the dataset of size 100 on the interval $[0, 4\pi]$, in which the points marked with plus-circle are the support vectors (51%) and the solid line is the estimation of $y = \sin x$ by the SVR algorithm. Figure 4(b) shows the error bars given by Eq. (4.13), in which the dotted lines are the error bar curves for the estimated curve (the solid line) and the dash-dotted line is the bound given by (4.14).

We should note that, when test point $\mathbf{z}$ is far away from the training data set, the covariance between $\mathbf{z}$ and X_M, $K_{X_M,\mathbf{z}}$, will approach zero due to the RBF kernel being used. Thus the error bar will be constant, just depending on $K_{\mathbf{z},\mathbf{z}} = 1$, C and ϵ (see (4.13)). The error bar will be dominated by $2/C^2$ when $\epsilon \to 0$. In this L_1-loss function case, the noise variance can be measured by $2/C^2$. When $C \to 0$ one will get a poor error bar estimation, and when $C \to \infty$, the noiseless case, the error bar will be controlled by the deadzone parameter ϵ.

(a) The Result of SVR Algorithm   (b) The Error Bar Curves
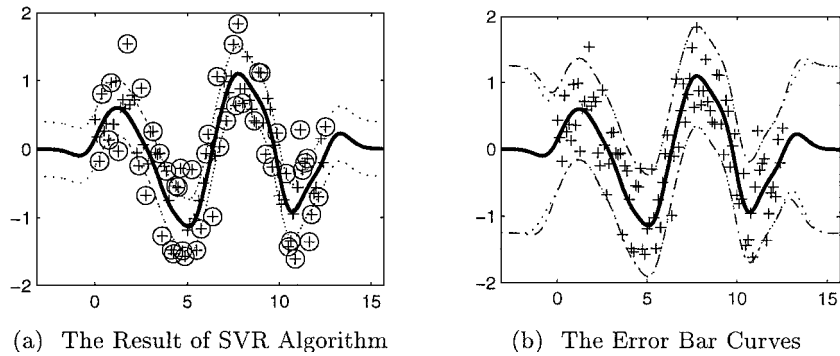
Figure 4.   The simulation results for the standard SVR algorithm and the error bar curves given by (4.13).

5. Comparison for error bars based on different loss functions

In the following discussion the loss function $l(u)$ is required to be convex and to be at least continuous everywhere on $u \in \mathbb{R}$. Loss functions can be categorized into three classes: (1) The loss function is C^1 except for at a finite number of points; for example, the ϵ-insensitive loss function $L_\epsilon(u)$ belongs to this class due to its non-differentiable points $u = \pm \epsilon$; (2) The loss function is C^2 except for at a finite number of points; for example, the Huber loss function; (3) the loss function is at least C^2 everywhere, for instance, the quadratic loss function.

For each loss function *a posteriori* probability of the unknown function $a(\mathbf{x})$ can be defined as:

$$P[\mathbf{a}(X) \,|\, \mathcal{D}] \propto \frac{\exp\left\{-C \sum_{\mathbf{x}_i \in X} l(y_i - a(\mathbf{x}_i)) - \frac{1}{2}\mathbf{a}(X)^T K_{X,X}^{-1} \mathbf{a}(X)\right\}}{\sqrt{\det 2\pi K_{X,X}} P[\mathcal{D}]} \tag{5.1}$$

and the resulting predictive distribution is given by:

$$P(a(\mathbf{z}) \,|\, \mathcal{D}) = \frac{1}{P(\mathcal{D})} \int P(\mathcal{D} \,|\, \mathbf{a}(X)) P(\mathbf{a}(X), a(\mathbf{z})) d\mathbf{a}(X).$$

For the loss function in the third class, one can easily tackle the problem of computing the evidence and the error bars of the model prediction based on a Taylor expansion at the MAP solution $a^*(x)$. The basic result, for the error bars, resembles the one for the quadratic loss function (Williams, 1998). Some examples can be found in Kwok (1998) for the soft-loss function used in classification problems.

A difficulty arises in dealing with the loss functions of the second class. Consider Huber's loss function as an example. In order to compute the evidence $P[\mathcal{D}]$ we need to compute the following integral:

$$\int \exp\left\{-\frac{1}{2}\delta\mathbf{a}(X)^T \left[K_{X,X}^{-1} + C\Lambda(\mathbf{a}^*(X))\right]\delta\mathbf{a}(X)\right\} d\delta\mathbf{a}(X), \tag{5.2}$$

where $\Lambda(\mathbf{a}^*(X)) = \text{diag}[\lambda(a^*(\mathbf{x}_i))]$ is a diagonal matrix whose components are a function of $\delta a(\mathbf{x}_i)$ defined as follows:

– For $y_i - a^*(\mathbf{x}_i) < -\epsilon$, $\lambda(a^*(\mathbf{x}_i)) = 0$;
– For $y_i - a^*(\mathbf{x}_i) > \epsilon$, $\lambda(a^*(\mathbf{x}_i)) = 0$;
– For $\epsilon < y_i - a^*(\mathbf{x}_i) < \epsilon$, $\lambda(a^*(\mathbf{x}_i)) = 1$;
– For $y_i - a^*(\mathbf{x}_i) = -\epsilon$,

$$\lambda(a^*(\mathbf{x}_i)) = \begin{cases} 0 & \delta a(\mathbf{x}_i) < 0 \\ 1 & \delta a(\mathbf{x}_i) > 0 \end{cases}$$

– For $y_i - a^*(\mathbf{x}_i) = \epsilon$,

$$\lambda(a^*(\mathbf{x}_i)) = \begin{cases} 1 & \delta a(\mathbf{x}_i) < 0 \\ 0 & \delta a(\mathbf{x}_i) > 0 \end{cases}$$

The results are analytically intractable, and an approximate method needs to be developed for its evaluation.

6. Conclusions

In summary, this paper describes a probabilistic framework for Gaussian SVM regression model. This approach allows an evidence to be defined and computed by the MAP approximation method, enabling an optimal value of the regularisation parameter C to be determined by Bayesian methods such as MacKay's evidence technique. Additionally, the corresponding error bars for prediction can be derived from the Bayesian Rule. Future work will focus on a more comprehensive test of the evidence and error bar formula, and investigate the comparison between the Gaussian SVM and other standard Gaussian Process methods.

Acknowledgments

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19:6*, 716–723.

Evgeniou, T., Pontil, M., & Poggio, T. (1999). *A unified framework for regularization networks and support vector machines*. A.I. Memo 1654, AI Lab, MIT, Massachusetts.

Gao, J., Gunn, S., Harris, C., & Brown, M. (2000). *A variational approach for support vector regression based on probabilistic framework*. Reaserch Report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machines. *Neural Computation, 10:6*, 1455–1480.

Gunn, S. (1998). *Support vector machines for classification and regression*. Technical Report, ISIS, Department of Electronics and Computer Science, University of Southampton.

Gunn, S., Brown, M., & Bossley, K. (1997). Network performance assessment for neurofuzzy data modelling. In *Lecture notes in computer science* (vol. 1280, pp. 313–323). Boston: Academic Press.

Kwok, J. T.-Y. (1999). Moderating the outputs of support vector machine classifiers. *IEEE-NN, 10:5*, 1018.

MacKay, D. (1991). *Bayesian modelling and neural networks*. Ph.D. Thesis, California Institute of Technology, Pasadena, CA.

MacKay, D. (1992). Bayesian interpolation. *Neural Computation, 4:3*, 415–447.

MacKay, D. (1997). Gaussian processes, a replacement for neural networks. NIPS tutorial 1997, Cambridge University.

Murata, N., Yoshizawa, S., & Amari, S. (1994). Network information criterion–determining the number of hidden units for artificial neural network models. *IEEE Transactions on Neural Networks*, *5*, 865–872.

Neal, R. (1996). *Bayesian learning for neural networks*, Lecture Notes in Statistics. New York: Springer.

Poggio, T. & Girosi, F. (1998). A sparse representation for function approximation. *Neural Computation*, *10*, 1445–1454.

Smola, A. (1998). *Learning with kernels*. Ph.D. Thesis, Technischen Universität Berlin, Berlin, Germany.

Sollich, P. (1999a). Approximate learning curves for Gaussian processes. In *ICANN99: Ninth International Conference on Artificial Neural Networks* (pp. 437–442). London.

Sollich, P. (1999b). *Bayesian methods for support vector machines: Evidence and error bars*. Technical Report, King's College London, London, UK. Accepted by *Machine Learning*.

Sollich, P. (1999c). *Probabilistic interpretations and Bayesian methods for support vector machines*. Technical Report, King's College London, London, UK.

Sollich, P. (2000). Probabilistic methods for support vector machines. In S. Solla, T. Leen, & K. Möller (Eds.), *Advnaces in neural information processing systems* (pp. 349–355). Cambridge, MA: MIT Press.

Tikhonov, A. & Arsenin, V. (1977). *Solution of ill-posed problems*. Washington, D.C.: W.H. Winston.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

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

Wahba, G. (1990). *Splines models for observational data*, vol. 59 of Series in Applied Mathematics. Philadelphia: SIAM Press.

Wahba, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods—support vector learning* (pp. 68–88). Cambridge, MA: MIT Press.

Williams, C. (1997). Computing with infinite networks. In M. Mozer, M. Jordan, & T. Petsche (Eds.), *Neural information processing systems* (vol. 9, pp. 295–301). Cambridge, MA: MIT Press.

Williams, C. (1998). Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. Jordan (Ed.), *Learning in graphical models* (pp. 599–621). Cambridge, MA: MIT Press.