ARTICLE IN PRESS

ELSEVIER

Neurocomputing ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Mean field method for the support vector machine regression

J.B. Gao[a,*], S.R. Gunn[b], C.J. Harris[b]

[a] *School of Mathematical and Computer Sciences, University of New England, Armidale, NSW 2351, Australia*

[b] *Image, Speech and Intelligent System Research Group, Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*

Received 3 April 2001; accepted 17 February 2002

Abstract

This paper deals with two subjects. First, we will show how support vector machine (SVM) regression problem can be solved as the maximum a posteriori prediction in the Bayesian framework. The second part describes an approximation technique that is useful in performing calculations for SVMs based on the mean field algorithm which was originally proposed in Statistical Physics of disordered systems. One advantage is that it handle posterior averages for Gaussian process which are not analytically tractable.

Keywords: Support vector machine; Mean field method; Regression; Gaussian process

1. Introduction

Recently, there has been a great deal of interest in non-parametric Bayesian approaches to regression and classification problems which are based on the concept of Gaussian processes (GP) [10,26,12], etc. It is well known that support vector machines (SVM) can be interpreted as the maximum a posteriori (MAP) prediction with a Gaussian prior, i.e., GP, under the Bayesian framework so that some statistical quantities such as error bars can be determined from this probabilistic interpretation, (see [4,19,18,25], etc.). The underlying idea is conceptually very simple. Instead of defining prior distributions over parameters of a learning machine, one directly defines a Gaussian prior distribution over the function space on which the machine computes. These

* Corresponding author.

E-mail addresses: jbgao@mcs.une.edu.au (J.B. Gao), srg@ecs.soton.ac.uk (S.R. Gunn), cjh@ecs.soton.ac.uk (C.J. Harris).

0925-2312/02/$ - see front matter
PII: S0925-2312(02)00573-8

ideas provide a probabilistic interpretation for regression and classification problems. The SVM method is a non-parametric technique by which infinitely many parameters can be tuned. However, only a finite number of them are active, the number increasing with the amount of data. The standard SVM solution can be obtained by a quadratic optimization algorithm that maximizes the posterior distribution through a dual optimization problem. This is closely related to other kernel-based methods [24,17].

Bayesian methods have a number of virtues, particularly their uniform treatment of uncertainty at all levels of the modelling process. The formalism also allows ready incorporation of prior knowledge and the seamless combination of such knowledge with observed data [6]. Another virtue of Bayesian framework is that it gives prediction statistics so that one can easily obtain error bars, etc. The Bayesian method has been successfully applied to the L_2 network regularization and some classification cases with a Gaussian prior [27,29,11,28]. The main difficulty in adopting a probabilistic framework for SVMs is due to the SVM likelihood (loss) function, i.e., the non-normalized likelihood in SVM classification and the likelihood defined by Vapnik's ε-loss function which will result in an intractable high-dimension integral. One advantage of such loss functions is that they enable spares solutions to be obtained, enabling fast implementations. In the L_2 network, the likelihood on the training data set is a Gaussian, so that the posterior distribution of the given data is also a Gaussian when a Gaussian prior distribution is given. However, for some statistical models like the ones used for classification or SVMs, the high-dimensional integrals which occur in performing a posteriori averages can only be treated by approximative methods. An approximation to these integrations can be based on Markov Chain Monte Carlo sampling [11] which, for large data sets, may be time consuming, for example, Sollich's technique for SVM classification. There are other possible approaches which can be used to approximate the posterior distribution or its statistical average, such as variational field algorithm [7,8] or Laplace's methods (the approximation of the posterior by a multivariate Gaussian at the most probable solution, i.e., an MAP). The Laplace approximation has been used for both classification problems [1,28], SVR problems [4] and SVC problems [18,19]. In an SVM the MAP solution has to be determined by a quadratic programming (QP) problem which is very time consuming when dealing with a large training data set and then the posterior distribution is simply approximated by a Gaussian distribution centered at the MAP solution based on the first-order expansion. We should notice here that such approximation is only needed when we try to put the SVM in a Bayesian framework. Also the variational method has recently been applied to the SVM regression in [3] and an adaptive method was developed to estimate the control factor C from the data set. In this method a Gaussian approximation to the posterior distribution is learnt step by step from the best local Gaussian approximation.

This paper deals with a different approach which has its origin in the Statistical Physics of disordered systems, called mean field theory. This method has been recently applied to the classification problem corresponding *probit model* [25,13]. This paper is organized as follows: Section 2 gives the basic definition for SVR with a Gaussian prior on the function space. Section 3 is dedicated to constructing the framework of the mean field method for the SVM regression and producing a system of mean field equations. Based on the mean field equations an iterative algorithm is designed to

approximate the SVR solution in Section 4. Section 5 is dedicated to deriving an error bar formula for the SVR solution. In Section 6, a numerical example is used to demonstrate the performance of the proposed approach and to compare it with the standard SVM regression algorithm.

2. SVR with Gaussian prior

Consider the supervised learning problem: A training set, $\mathscr{D}=\{(\boldsymbol{x}_i,t_i)|i=1,2,\ldots,N\}$ of input vectors $\boldsymbol{x}_i$ and associated targets t_i is given and the goal is to infer the output t for a new input $\boldsymbol{x}$. Here, we consider the special case of the SVR problem with Vapnik's ε-loss function defined as

$$L_\varepsilon(t-y(\boldsymbol{x}))=L(t,y(\boldsymbol{x}))=\begin{cases}0 & |t-y(\boldsymbol{x})|\leqslant\varepsilon,\\ |t-y(\boldsymbol{x})|-\varepsilon & |t-y(\boldsymbol{x})|>\varepsilon,\end{cases} \tag{2.1}$$

where $\varepsilon\geqslant 0$ is a prespecified constant controlling the noise tolerances. There is no penalty at $\boldsymbol{x}_i$ when $|t_i-y(\boldsymbol{x}_i)|\leqslant\varepsilon$.

In order to construct a Bayesian framework under Vapnik's ε-insensitive loss function L_ε, we employ the probabilistic model in which the probability of the output t, at a given point $\boldsymbol{x}$, the likelihood $P[t|y(\boldsymbol{x})]$ is assumed, through a hidden function variable $y(\boldsymbol{x})$, by the following relationship:

$$P[t|y(\boldsymbol{x})]=\frac{C}{2(\varepsilon C+1)}\exp\{-CL_\varepsilon(t-y(\boldsymbol{x}))\}. \tag{2.2}$$

Thus, Eq. (2.2) can be interpreted as an additive noise model of the target t. Recently, [16] have investigated this noise model and proposed a probabilistic interpretation, showing that the standard SVR framework is a special case of four regularization network [2] with this particular noise model.

The probabilistic interpretation of SVRs can be regarded as the following likelihood defined by the ε-insensitive loss function

$$P[\mathscr{D}|\boldsymbol{y}(\boldsymbol{X})]=\left[\frac{1}{2}\,\frac{C}{\varepsilon C+1}\right]^N\exp\left\{-C\sum_{i=1}^{N}L_\varepsilon(t_i-y(\boldsymbol{x}_i))\right\}, \tag{2.3}$$

where $\boldsymbol{y}(\boldsymbol{X})=[y(\boldsymbol{x}_1),y(\boldsymbol{x}_2),\ldots,y(\boldsymbol{x}_N)]$. We take the prior probability distribution $P[y(\boldsymbol{x})]$ as a functional GP. A functional GP is defined as a stochastic process specified by giving only the mean vector and covariance matrix for any finite subset of points. For any finite point set $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N]$ the prior probability distribution $P[\boldsymbol{y}(\boldsymbol{x})]$ is specified as a GP with a zero mean and a covariance function $K(\boldsymbol{x},\boldsymbol{x}')$,

$$P[\boldsymbol{y}(\boldsymbol{X})]=\frac{1}{\sqrt{\det 2\pi K_N}}\exp\left\{-\frac{1}{2}\,\boldsymbol{y}(\boldsymbol{X})^{\mathrm{T}}K_N^{-1}\boldsymbol{y}(\boldsymbol{X})\right\}, \tag{2.4}$$

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

Combining (2.3) and (2.4) and applying Bayes' rule results in the following posterior about the process function $\boldsymbol{y}(\boldsymbol{X})$:

$$P[\boldsymbol{y}(\boldsymbol{X})|\mathscr{D}] = \frac{K(C,\varepsilon)^N}{\sqrt{\det(2\pi K_N^{-1})}} \times \exp\left\{-C\sum_{i=1}^{N} L_\varepsilon(t_i - y(\boldsymbol{x}_i)) - \frac{1}{2}\boldsymbol{y}(\boldsymbol{X})^{\mathrm{T}} K_N^{-1} \boldsymbol{y}(\boldsymbol{X})\right\} \Big/ P[\mathscr{D}], \tag{2.5}$$

where $K(C,\varepsilon) = C/2(\varepsilon C + 1)$ and the normalization constant $P[\mathscr{D}]$ is given by

$$P[\mathscr{D}] = \frac{K(C,\varepsilon)^N}{\sqrt{\det(2\pi K_N^{-1})}} \int \exp\left\{ -C\sum_{i=1}^{N} L_\varepsilon(t_i - y(\boldsymbol{x}_i)) - \frac{1}{2}\boldsymbol{y}(\boldsymbol{X})^{\mathrm{T}} K_N^{-1} \boldsymbol{y}(\boldsymbol{X}) \right\} \mathrm{d}\boldsymbol{y}(\boldsymbol{X}). \tag{2.6}$$

It is obvious that the MAP estimate of the posterior distribution $P[\boldsymbol{y}(\boldsymbol{X})|\mathscr{D}]$ is the minimizer of

$$\min_{\boldsymbol{y}(\boldsymbol{X})} C\sum_{i=1}^{N} L_\varepsilon(t_i - y(\boldsymbol{x}_i)) + \frac{1}{2}\boldsymbol{y}(\boldsymbol{X})^{\mathrm{T}} K_N^{-1} \boldsymbol{y}(\boldsymbol{X}). \tag{2.7}$$

The original SVM setting, see (2.7), can be converted into a QP problem (see [23,17,5], etc.) by introducing some slack variables and dual variables. Due to the size of the optimization problems arising from SVM one has to pay special attention as to how these problems can be solved efficiently. Several algorithms can be used to solve the QP problem arising in SVR. Most of them can be shown to share some common strategy that can be understood well in the view of duality theory. These algorithms include the interior point algorithm [21], the subset selection algorithms [22,14,9], the sequential minimal optimization (SMO) [15]. The interior point algorithm is relatively fast and achieves a high-precision solution in the case of moderate size problems (up to approximately 3000 samples). The subset selection algorithms requires the initial problem to be broken up into sub-problems which are then in turn solved separately, so that the solution approaches the global optimum. The SMO technique is a very robust algorithm which is the limiting case of Osuna's method having a working set size of two. The optimization sub-problem can be solved analytically without explicitly invoking a quadratic optimizer. However, all of these algorithms are limited by the scale of the problem and also require a priori control factor C and precision size ε.

3. Mean field theory of SVR

The calculation posterior average needed to derive Bayes algorithm is typically intractable and approximation techniques are required. Recently, [13] have introduced an

advanced mean field theory approach based on ideas of statistical mechanics to cope with the Gaussian classification problem. This approach is equivalent to the so-called TAP mean field theory first developed by [20].

In this section we will follow the discussion in [13]. From the posterior distribution defined in (2.5) the prediction on a new test input $\boldsymbol{x}$ is given by

$$\langle y(\boldsymbol{x})\rangle = \int y(\boldsymbol{x}) P[y(\boldsymbol{x})|\mathscr{D}]\,\mathrm{d}y(\boldsymbol{x}) = \int y(\boldsymbol{x}) P[y(\boldsymbol{x}),\boldsymbol{y}(\boldsymbol{X})|\mathscr{D}]\,\mathrm{d}y(\boldsymbol{x})\,\mathrm{d}\boldsymbol{y}(\boldsymbol{X})$$

$$= \frac{K(C,\varepsilon)^N}{\sqrt{\det(2\pi K_{N+1}^{-1})}} \times \int y(\boldsymbol{x}) \frac{\exp\{-C\sum_{i=1}^N L_\varepsilon(t_i - y(\boldsymbol{x}_i)) - \frac{1}{2}\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x})^{\mathrm{T}} K_{N+1}^{-1}\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x})\}}{P[\mathscr{D}]}\,\mathrm{d}y(\boldsymbol{x})\,\mathrm{d}\boldsymbol{y}(\boldsymbol{X}), \tag{3.1}$$

where $\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x}) = [y(\boldsymbol{x}_1),\ldots,y(\boldsymbol{x}_N),y(\boldsymbol{x})]^{\mathrm{T}}$ and

$$K_{N+1} = \begin{pmatrix} K_N & k_N(\boldsymbol{X})^{\mathrm{T}} \\ k_N(\boldsymbol{X}) & K(\boldsymbol{x},\boldsymbol{x}) \end{pmatrix}$$

with $k_N(\boldsymbol{X})$ defined by $k_N(\boldsymbol{X}) = [K(\boldsymbol{x}_1,\boldsymbol{x}),K(\boldsymbol{x}_2,\boldsymbol{x}),\ldots,K(\boldsymbol{x}_N,\boldsymbol{x})]$. Note that

$$y(\boldsymbol{x})\exp\left\{-\frac{1}{2}\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x})^{\mathrm{T}} K_{N+1}^{-1}\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x})\right\} = \sum_{i=1}^{N+1} K(\boldsymbol{x},\boldsymbol{x}_i)\frac{\partial}{\partial y(\boldsymbol{x}_i)} \times\exp\left\{-\frac{1}{2}\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x})^{\mathrm{T}} K_{N+1}^{-1}\boldsymbol{y}(\boldsymbol{X},\boldsymbol{x})\right\},$$

where $\boldsymbol{x}$ denotes $\boldsymbol{x}_{N+1}$, then by substituting the above relation into (3.1) and applying integration by parts

$$\langle y(\boldsymbol{x})\rangle = \frac{K(C,\varepsilon)^N}{P[\mathscr{D}]}\sum_{i=1}^N K(\boldsymbol{x},\boldsymbol{x}_i)\int N(\boldsymbol{y}(\boldsymbol{X})|\boldsymbol{0},K_N)\frac{\partial}{\partial y(\boldsymbol{x}_i)} \times\exp\left\{-C\sum_{j=1}^N L_\varepsilon(t_j - y(\boldsymbol{x}_j))\right\}\mathrm{d}\boldsymbol{y}(\boldsymbol{X})$$

$$= \sum_{i=1}^N K(\boldsymbol{x},\boldsymbol{x}_i)w_i, \tag{3.2}$$

where w_i is a constant defined as

$$w_i = \frac{K(C,\varepsilon)^N}{P[\mathscr{D}]}\int N(\boldsymbol{y}(\boldsymbol{X})|\boldsymbol{0},K_N)\frac{\partial}{\partial y(\boldsymbol{x}_i)} \times\exp\left\{-C\sum_{j=1}^N L_\varepsilon(t_j - y(\boldsymbol{x}_j))\right\}\mathrm{d}\boldsymbol{y}(\boldsymbol{X}). \tag{3.3}$$

Let us define a new distribution for each i as follows:

$$P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i] = \frac{\int N(\boldsymbol{y}(\boldsymbol{X})|\boldsymbol{0}, K_N)\exp\{-C\sum_{j\neq i} L_\varepsilon(t_j - y(\boldsymbol{x}_j))\}\,\mathrm{d}\boldsymbol{y}(\bar{\boldsymbol{X}}_i)}{\int N(\boldsymbol{y}(\boldsymbol{X})|\boldsymbol{0}, K_N)\exp\{-C\sum_{j\neq i} L_\varepsilon(t_j - y(\boldsymbol{x}_j))\}\,\mathrm{d}\boldsymbol{y}(\boldsymbol{X})}, \tag{3.4}$$

where $\bar{\mathscr{D}}_i$ and $\bar{\boldsymbol{X}}_i$ are obtained by removing the data pattern $(\boldsymbol{x}_i, t_i)$ from $\mathscr{D}$. In fact, $P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i]$ is the predictive distribution at the "test" point $\boldsymbol{x}_i$ given the data set $\bar{\mathscr{D}}_i$. Denoting an average with respect to this predictive distribution by

$$\langle\cdots\rangle_i = \int \cdots P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i]\,\mathrm{d}\,y(\boldsymbol{x}_i)$$

we can rewrite the coefficient in (3.3) as, see Eq. (2.6),

$$w_i = \frac{\langle K(C,\varepsilon)(\partial/\partial y(\boldsymbol{x}_i))\exp\{-CL_\varepsilon(t_i - y(\boldsymbol{x}_i))\}\rangle_i}{\langle K(C,\varepsilon)\exp\{-CL_\varepsilon(t_i - y(\boldsymbol{x}_i))\}\rangle_i}. \tag{3.5}$$

The magnitude of w_i can be interpreted as the normalized variant rate of the likelihood. Thus, the weight coefficients in the SVM solution (3.2) can be determined by the likelihood variant rates with respect to the local predictive distribution $P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i]$. In order to calculate such weights, a simple and direct method is to apply some Gaussian approximation to the local predictive distribution as follows:

$$P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i] \approx \frac{1}{\sqrt{2\pi\sigma_i^2}}\exp\left\{-\frac{(y(\boldsymbol{x}_i) - \langle y(\boldsymbol{x}_i)\rangle_i)^2}{2\sigma_i^2}\right\} \tag{3.6}$$

with the variance defined as $\sigma_i^2 = \langle y(\boldsymbol{x}_i)^2\rangle_i - \langle y(\boldsymbol{x}_i)\rangle_i^2$. Inserting (3.6) into (3.5) we derive the following expression:

$$w_i \approx \frac{F(\langle y(\boldsymbol{x}_i)\rangle_i, \sigma_i^2)}{G(\langle y(\boldsymbol{x}_i)\rangle_i, \sigma_i^2)}, \tag{3.7}$$

where F and G are computed, respectively, by the following explicit formula:

$$\begin{aligned} F(\langle y(\boldsymbol{x}_i)\rangle_i, \sigma_i^2) = {} & \frac{C}{2}\exp\left\{\frac{C}{2}(2\langle y(\boldsymbol{x}_i)\rangle_i - 2t_i + 2\varepsilon + C\sigma_i^2)\right\} \\ & \times\left[1 - \mathrm{erf}\left[\frac{\langle y(\boldsymbol{x}_i)\rangle_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ & - \frac{C}{2}\exp\left\{\frac{C}{2}(-2\langle y(\boldsymbol{x}_i)\rangle_i + 2t_i + 2\varepsilon + C\sigma_i^2)\right\} \\ & \times\left[1 - \mathrm{erf}\left[\frac{-\langle y(\boldsymbol{x}_i)\rangle_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2}\sigma_i^2}\right]\right], \end{aligned} \tag{3.8}$$

$$G(\langle y(\boldsymbol{x}_i)\rangle_i, \sigma_i^2) = \frac{1}{2}\,\mathrm{erf}\left[\frac{t_i - \langle y(\boldsymbol{x}_i)\rangle_i + \varepsilon}{\sqrt{2\sigma_i^2}}\right] - \frac{1}{2}\,\mathrm{erf}\left[\frac{t_i - \langle y(\boldsymbol{x}_i)\rangle_i - \varepsilon}{\sqrt{2\sigma_i^2}}\right]$$
$$+ \frac{1}{2}\exp\left\{\frac{C}{2}\,(2\langle y(\boldsymbol{x}_i)\rangle_i - 2t_i + 2\varepsilon + C\sigma_i^2)\right\}$$
$$\times\left[1 - \mathrm{erf}\left[\frac{\langle y(\boldsymbol{x}_i)\rangle_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right]$$
$$+ \frac{1}{2}\exp\left\{\frac{C}{2}\,(-2\langle y(\boldsymbol{x}_i)\rangle_i + 2t_i + 2\varepsilon + C\sigma_i^2)\right\}$$
$$\times\left[1 - \mathrm{erf}\left[\frac{-\langle y(\boldsymbol{x}_i)\rangle_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right]. \tag{3.9}$$

Eqs. (3.7)–(3.9) are called the mean field equations corresponding to the weight parameters. In order to work out the weight coefficients, one has to determine the local predictive average $\langle y(\boldsymbol{x}_i)\rangle_i$ and variance σ_i^2 in the approximated Gaussian (3.6). Recently, [13] have derived an effective mean field equation for both $\langle y(\boldsymbol{x}_i)\rangle_i$ and σ_i^2 by the TAP linear respondent approach. Denote by $\langle y(\boldsymbol{x}_i)\rangle$ the posterior average at $\boldsymbol{x}_i$ which is given by (3.2) as

$$\langle y(\boldsymbol{x}_i)\rangle = \sum_{j=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j) w_j,$$

then the formulas for $\langle y(\boldsymbol{x}_i)\rangle_i$ and σ_i^2 can be explicitly represented as

$$\langle y(\boldsymbol{x}_i)\rangle_i \approx \langle y(\boldsymbol{x}_i)\rangle - \sigma_i^2 w_i, \tag{3.10}$$

$$\sigma_i^2 \approx \frac{1}{[(\Sigma + K)^{-1}]_{ii}} - \Sigma_i, \tag{3.11}$$

with $\Sigma = \mathrm{diag}(\Sigma_1, \Sigma_2, \ldots, \Sigma_N)$ and

$$\Sigma_i = -\sigma_i^2 - \left(\frac{\partial w_i}{\partial\langle y(\boldsymbol{x}_i)\rangle_i}\right)^{-1}.$$

An explicit expression for $\partial w_i/\partial\langle y(\boldsymbol{x}_i)\rangle_i$ can be obtained from (3.7) as

$$\frac{\partial w_i}{\partial\langle y(\boldsymbol{x}_i)\rangle_i} \approx C^2 - w_i^2 - \frac{w_i\langle y(\boldsymbol{x}_i)\rangle_i + \sigma_i^2 C^2 \int_{t_i-\varepsilon}^{t_i+\varepsilon} P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i]\,\mathrm{d}\,y(\boldsymbol{x}_i)}{\sigma_i^2 G(\langle y(\boldsymbol{x}_i)\rangle_i, \sigma_i^2)}. \tag{3.12}$$

Since $P[y(\boldsymbol{x}_i)|\bar{\mathscr{D}}_i]$ is assumed to be Gaussian, the integral in (3.12) can easily be computed using the error function.

4. Algorithm for the SVR mean field equations

The required information needed for prediction at a test input $\boldsymbol{x}$ in (3.2) is w_i and the local predictive posterior average $\langle y(\boldsymbol{x}_i)\rangle_i$ and the local predictive posterior variance σ_i^2. These variables satisfy the non-linear mean field equations (3.7), (3.10) and (3.11), etc. The mean field equation can be solved by an iteration method:

1. initialization: Set the learning rate η, e.g., $\eta = 0.05$, and randomly set w_i,
2. calculate the Kernel matrix K and let $\sigma_i^2 = K_{ii}$,
3. iterate steps 4–6 until the change in w_i is below a given tolerance,
4. for $i = 1, \ldots, N$ do

$$\langle y(\boldsymbol{x}_i)\rangle := \sum_{j=1}^{N} K(\boldsymbol{x}_i, \boldsymbol{x}_j) w_j,$$

$$\langle y(\boldsymbol{x}_i)\rangle_i := \langle y(\boldsymbol{x}_i)\rangle - \sigma_i^2 w_i,$$

$$\begin{aligned} F_i := {} & \frac{C}{2} \exp\left\{\frac{C}{2}\left(2\langle y(\boldsymbol{x}_i)\rangle_i - 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\} \\ & \times \left[1 - \operatorname{erf}\left[\frac{\langle y(\boldsymbol{x}_i)\rangle_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ & - \frac{C}{2} \exp\left\{\frac{C}{2}\left(-2\langle y(\boldsymbol{x}_i)\rangle_i + 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\} \\ & \times \left[1 - \operatorname{erf}\left[\frac{-\langle y(\boldsymbol{x}_i)\rangle_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right], \end{aligned}$$

$$\begin{aligned} G_i := {} & \frac{1}{2} \operatorname{erf}\left[\frac{t_i - \langle y(\boldsymbol{x}_i)\rangle_i + \varepsilon}{\sqrt{2\sigma_i^2}}\right] - \frac{1}{2} \operatorname{erf}\left[\frac{t_i - \langle y(\boldsymbol{x}_i)\rangle_i - \varepsilon}{\sqrt{2\sigma_i^2}}\right] \\ & + \frac{1}{2} \exp\left\{\frac{C}{2}\left(2\langle y(\boldsymbol{x}_i)\rangle_i - 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\} \\ & \times \left[1 - \operatorname{erf}\left[\frac{\langle y(\boldsymbol{x}_i)\rangle_i - t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right] \\ & + \frac{1}{2} \exp\left\{\frac{C}{2}\left(-2\langle y(\boldsymbol{x}_i)\rangle_i + 2t_i + 2\varepsilon + C\sigma_i^2\right)\right\} \\ & \times \left[1 - \operatorname{erf}\left[\frac{-\langle y(\boldsymbol{x}_i)\rangle_i + t_i + \varepsilon + C\sigma_i^2}{\sqrt{2\sigma_i^2}}\right]\right], \end{aligned}$$

5. update w_i by

$$w_i := w_i + \eta(F_i/G_i - w_i),$$

6. for every M iterations of w_i, update

$$IG_i = \frac{1}{2}\,\mathrm{erf}\left[\frac{t_i - \langle y(\boldsymbol{x}_i)\rangle_i + \varepsilon}{\sqrt{2\sigma_i^2}}\right] - \frac{1}{2}\,\mathrm{erf}\left[\frac{t_i - \langle y(\boldsymbol{x}_i)\rangle_i - \varepsilon}{\sqrt{2\sigma_i^2}}\right],$$

$$Dw_i := C^2 - w_i^2 - \frac{w_i\langle y(\boldsymbol{x}_i)\rangle_i + \sigma_i^2 C^2 IG_i}{\sigma_i^2 G(\langle y(\boldsymbol{x}_i)\rangle_i, \sigma_i^2)},$$

$$\Sigma_i := -\sigma_i^2 - 1/Dw_i,$$

$$\sigma_i^2 := \frac{1}{[(\Sigma + K)^{-1}]_{ii}} - \Sigma_i.$$

In the above iteration, steps 4 and 5 are called the inner iteration and step 6 the outer iteration. It is obvious that the most expensive step in the above mean field algorithm is the inversion of the matrix $K + \Sigma$ in the outer iteration cycle and in general the iterative procedure is somewhat insensitive to the precise value of the weights w_i, so we choose to make a less iterations for the outer iteration than for the inner iteration. For example, after $M = 30$ inner iteration update Σ_i and σ_i^2 in the outer iteration. The most important thing is to make the algorithm converge to the required accuracy. There are several things that one can do to improve convergence, for example: (1) decrease the learning rate η; (2) make η adaptable to the data; and (3) increase noise in the diagonal of kernel matrix K, etc.

5. Error bar estimation

Here, we consider the problem of estimating the prediction error for the SVR problem through the mean field method. When given a prediction, it is also very useful to have some estimates of the error bars associated with that model prediction. Error bars arise naturally in a Bayesian treatment of learning machines and are made up of two terms, one due to a posteriori uncertainty (the uncertainty of parameter $\boldsymbol{w}$), and the other due to the intrinsic target noise in the data.

In the mean field method the posterior distribution was characterized by the first two moments of the posterior distribution, the mean $\langle y(\boldsymbol{x})\rangle$ and the variance $\sigma_{\boldsymbol{x}}^2$ defined as

$$\sigma_{\boldsymbol{x}}^2 = \langle y(\boldsymbol{x})^2\rangle - \langle y(\boldsymbol{x})\rangle^2.$$

The implicit assumption of this approach is that the posterior distribution $P[y(\boldsymbol{x})|\mathscr{D}]$ can be approximated by a Gaussian distribution with the mean $\langle y(\boldsymbol{x})\rangle$ and variance $\sigma_{\boldsymbol{x}}^2$. The posterior mean $\langle y(\boldsymbol{x})\rangle$ has been calculated by the mean field algorithm. An approximation formula for the variance of the posterior distribution, $\sigma_{\boldsymbol{x}}^2$, has been derived in [13] by applying a linear response argument, such that

$$\sigma_{\boldsymbol{x}}^2 \approx K(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{K}(\boldsymbol{x},\boldsymbol{X})^{\mathrm{T}}[K + \Sigma]^{-1}\boldsymbol{K}(\boldsymbol{x},\boldsymbol{X}), \tag{5.1}$$

where $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{X}) = [K(\boldsymbol{x},\boldsymbol{x}_1), K(\boldsymbol{x},\boldsymbol{x}_2), \ldots, K(\boldsymbol{x},\boldsymbol{x}_N)]^{\mathrm{T}}$ and $\Sigma = \mathrm{diag}(\Sigma_1, \Sigma_2, \ldots, \Sigma_N)$.

Thus, in the mean field method the posterior distribution is approximated by

$$P[y(\boldsymbol{x})|\mathscr{D}] \approx N(y(\boldsymbol{x})|\langle y(\boldsymbol{x})\rangle, \sigma_{\boldsymbol{x}}^2) = \frac{1}{\sqrt{2\pi\sigma_{\boldsymbol{x}}^2}} \exp\left\{-\frac{(y(\boldsymbol{x}) - \langle y(\boldsymbol{x})\rangle)^2}{2\sigma_{\boldsymbol{x}}^2}\right\}.$$

Consider the prediction for a new test data point $\boldsymbol{x}$. The prediction distribution for the target t given the data set $\mathscr{D}$ can be generated from the ε-insensitive noise model (2.2) and the approximated posterior $N(\langle y(\boldsymbol{x})\rangle, \sigma_{\boldsymbol{x}}^2)$ as follows

$$P[t|\mathscr{D}] = \int K(C,\varepsilon) \exp\{-CL_\varepsilon(t - y(\boldsymbol{x}))\} N(y(\boldsymbol{x})|\langle y(\boldsymbol{x})\rangle, \sigma_{\boldsymbol{x}}^2)\, \mathrm{d}y(\boldsymbol{x}). \tag{5.2}$$

Let us first note that the likelihood of SVM case is described by $K(C,\varepsilon)\exp\{-CL_\varepsilon (t - y(\boldsymbol{x}))\}$. This distribution function with respect to t can be represented as a superposition of GP, (see [2]). By using the technique in [3] the posterior mean (first moment) of the target is

$$\langle t|\mathscr{D}\rangle = \int tP[t|\mathscr{D}]\, \mathrm{d}t = \sum_{i=1}^{N} K(\boldsymbol{x}, \boldsymbol{x}_i) w_i$$

and the second moment of the target is

$$\langle t^2|\mathscr{D}\rangle = \sigma_{\boldsymbol{x}}^2 + \frac{2}{C^2} + \frac{\varepsilon^3 + 3\varepsilon^2}{2C^2(\varepsilon+1)} + \langle t|\mathscr{D}\rangle^2.$$

Therefore, the estimates of the variance, or the error bar of the SVR based on the mean field algorithm, is then given by

$$\sigma_t^2 = \sigma_{\boldsymbol{x}}^2 + \frac{2}{C^2} + \frac{\varepsilon^3 + 3\varepsilon^2}{2C^2(\varepsilon+1)}. \tag{5.3}$$

This error bar has two components, see Eq. (5.3). The first $\sigma_{\boldsymbol{x}}^2$ is an estimate of the width of the posterior over the hidden function $y(\boldsymbol{x})$, i.e., the function uncertainty. The second term can be viewed as the measure for the uncertainty induced in the target noise determined by the control factor C and ε.

A similar error bar formula for the SVR problem has been given in our previous paper [4] based on the Laplacian approximation to the posterior distribution at the SVM solution. The approximation approach taken there is achieved by a first-order Taylor expansion of the ε-insensitive loss function which is just (left and right) differentiable. In this paper, the posterior distribution has also been approximated by a Gaussian distribution but with the moments determined by the mean field. Theoretically, the mean field method may approximate the posterior average with arbitrary accuracy with enough computation time. In this case, the error bar given by the mean field method will give more confidence for the uncertainty estimates.

6. Simulation

In this section, the performance of the mean field method for the SVR problem is studied. As an illustration of this algorithm, we consider a simple illustrative problem

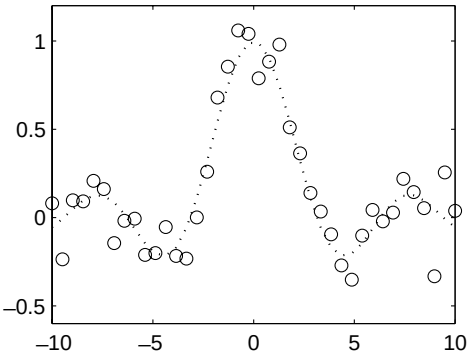

Fig. 1. The true function $y(x)$ and the uniformly spaced samples with the noises of level $C = 5$.

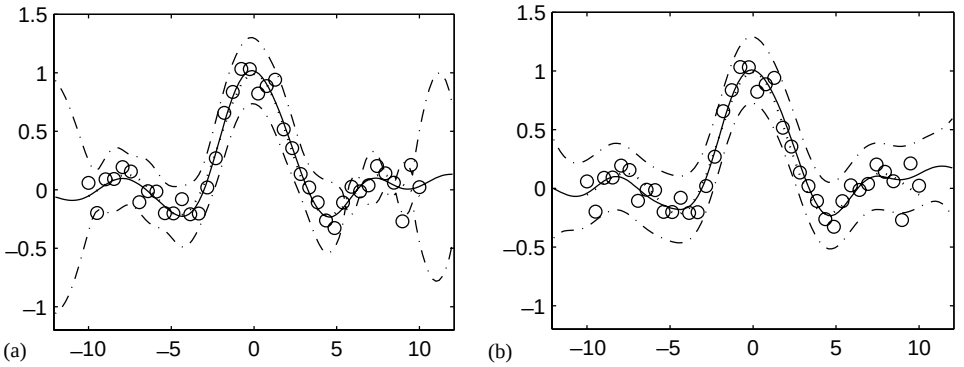

Fig. 2. The mean field approximation and the standard QP approximation of SVR to $\sin c(x)$ with values of $C = 5$: the true function (dotted curve) and the approximated function (solid curve) and the error bar curves (dot–dashed curves).

involving one input and one output in which the mean target function $y(x)$ is known for a target with additive noise. The $y(x)$ is the so-called $\sin c$ function, i.e., $y(x) = \sin c(x) = x^{-1}\sin(x)$. We take a data set of $N = 40$ training points in which the input data point x is picked uniformly from the interval $[-10, 10]$ and the target, t, is generated by an additive noise process, $t_i = y(x_i) + \xi_i$, where ξ_i is additive noise with zero mean and the standard derivation $\sigma = 0.1$. For the experiment a Gaussian RBF kernel $K(x, x_i) = \exp\{-|x - x_i|^2/\tau^2\}$ with width $\tau = 2$ was used. The function and the training data set are illustrated in Fig. 1.

We approximate the true function $y(x)$ by the standard SVM regression and the mean field algorithm. The cost function is chosen to be $L_\varepsilon(\cdot)$ with the precision parameter ε being the standard derivation of the target noise, $\varepsilon = 0.1$, in this implementation. The control parameter C was chosen to be 5 as suggested in [3]. Fig. 2(a) illustrates the

result approximated by the mean field algorithm. In Fig. 2, the circled points are the training data points and the true function is plotted as a dotted line and the approximated function is drawn as a solid line. The dot–dashed lines represent the error bar given by formula (5.3). Under the same parameter the standard SVR algorithm is implemented and the result is shown in Fig. 2(b).

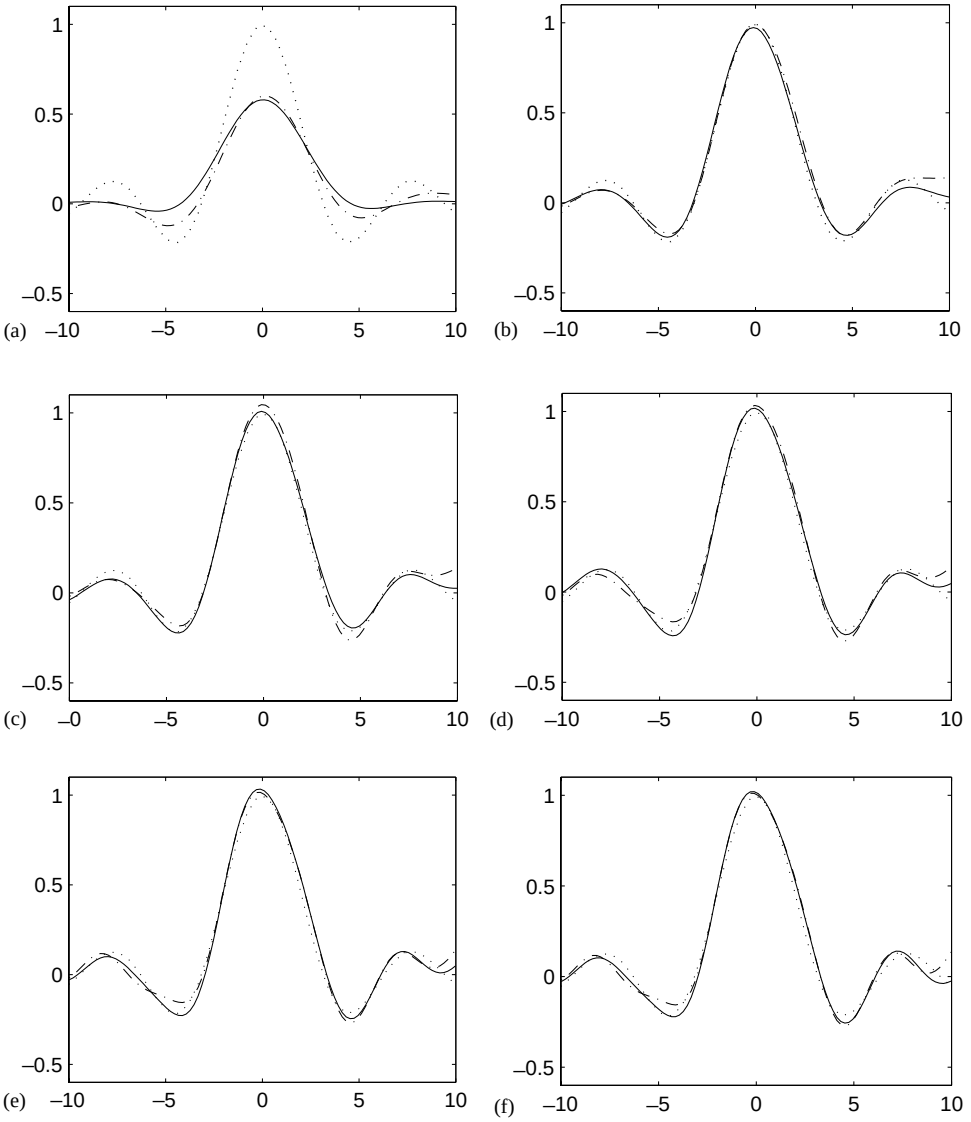

Fig. 3. SVM approximation to $\sin c(x)$ with different values of C: the true function (dotted curve) and the approximated function by mean field method (solid curve), and the solution generated by the standard SVM algorithm with QP (dashed curve), where: (a) $C = 0.1$; (b) $C = 1$; (c) $C = 2$; (d) $C = 4$; (e) $C = 8$; and (f) $C = 10$.

From Fig. 2 we can see that the error bar given by (5.3) is more sensitive to the data set than the error bar given by the Laplace approximation. The ratio of the error bar width outside of data set to inside of data set region for the error bar (5.3) is larger than that of the error bar of the Laplace approximation.

In order to further compare the performance of the mean field algorithm for the SVR problem and the standard QP algorithm, we have also implemented these two algorithms for the different values of the control factor C. These results are shown in the Fig. 3. We can conclude that the approximation performance of the mean field algorithm is similar to that of the SVR through QP algorithm.

7. Conclusions

We have shown that the mean field approach can be used in the SVR problem as the same as in the classification problem. Based on the mean field equation for a Gaussian process an efficient iterative implementation algorithm has been derived in this paper. Another point to note is that the mean field SVR method is moderately easy to implement and use. Like the standard SVR algorithm, the implementation of mean field SVR requires a great deal of computation for large matrix inverses. The control factor C and precision parameter ε should be prespecified in the current form of algorithm. The better algorithm is able to adapt these parameters to the training data set. The standard SVM algorithm gives a deterministic solution but does not provide any statistical information, thus no confidence estimate, such as the error bars are available. Although we can interpret the SVM regression method under the probabilistic framework [4], the error bar estimation is calculated from the whole Gaussian approximation at the MAP solution based on the support vectors. However, under the mean field framework here a Gaussian distribution with covariance $K + \Sigma$ is used to approximate the posterior distribution with much more possible accuracy.

Acknowledgements

This research is sponsored by EPSRC, UK. Partial support was also provided by the Natural Science Foundation of China (Grant No.: 19871032). The first author wishes to thank Ole Winther and Tommi S. Jaakkola for their helpful discussions as well as the Reviewers for their encouraging suggestions.

References

[1] D. Barber, C. Williams, Gaussian processes for Bayesian classification via hybrid Monte Carlo, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Neural Information Processing Systems, Vol. 9, MIT Press, Cambridge, MA, 1997, pp. 340–346.

[2] T. Evgeniou, M. Pontil, T. Poggio, A unified framework for regularization networks and support vector machines, A.I. Memo 1654, AI Lab, MIT, MA, 1999.

[3] J. Gao, S. Gunn, C. Harris, M. Brown, SVM regression through variational methods and its online implementation, Technical Report, IEEE Trans. Neural Networks, 2000, submitted for publication.

[4] J. Gao, S. Gunn, C. Harris, M. Brown, A probabilistic framework for SVM regression and error bar estimation, Mach. Learn. 22 (1–2) (2002).

[5] S. Gunn, Support vector machines for classification and regression, Technical Report, ISIS, Department of Electronics and Computer Science, University of Southampton, 1998.

[6] D. Heckerman, D. Geiger, D. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data Mach. Learn. 20 (3) (1995) 197–201.

[7] T. Jaakkola, M. Jordan, Variational probabilistic inference and the qmr-dt database, J. Artif. Intell. Res. 10 (1999) 291–322.

[8] T. Jaakkola, M. Jordan, Bayesian parameter estimation through variational methods, Statist. Comput. 10 (2000) 25–37.

[9] T. Joachims, Making large-scale SVM learning practical, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 169–184.

[10] D. MacKay, Gaussian processes, A replacement for neural networks, NIPS Tutorial 1997, Cambridge University, Cambridge, 1997.

[11] R. Neal, Monte Carlo implementation of Gaussian process models for Bayesian regression and classification, Technical Report CRG-TR-97-2, Department of Computer Science, University of Toronto, 1997.

[12] M. Opper, O. Winther, Gaussian processes for classification, Research Report, Neural Computing Research Group, Aston University, Birmingham, UK, 1999, Neural Computon, submitted for publication.

[13] M. Opper, O. Winther, Gaussian processes for classification: mean field algorithms Neural Comput. 12 (2000) 2655–2684.

[14] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: J. Principe, L. Gile, N. Morgan, E. Wilson (Eds.), Neural Networks for Signal Processing VII—Proceedings of the 1997 IEEE Workshop, IEEE, New York, 1997, pp. 276–285.

[15] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 185–208.

[16] M. Pontil, S. Mukherjee, F. Girosi, On the noise model of support vector machine regression, A.I. Memo 1651, AI Laboratory, MIT, MA, 1998.

[17] A. Smola, Learning with Kernels, Ph.D. Thesis, Technischen Universität Berlin, Germany, 1998.

[18] P. Sollich, Approximate learning curves for Gaussian processes, in: ICANN99: Ninth International Conference on Artificial Neural Networks, IEE, London, 1999a, pp. 437–442.

[19] P. Sollich, Probabilistic interpretations and Bayesian methods for support vector machines, in: CANN99—Ninth International Conference on Artificial Neural Networks, IEE, London, 1999b, pp. 91–96.

[20] D. Thouless, P. Anderson, R. Palmer, Solution of a solvable model of a spin glass, Philos. Mag. 35 (1977) 593.

[21] R. Vanderbei, LOGO: An Interior point code for quadratic programming, TR SOR-94-15, Pn: Statistics and Operations Research, Princeton University, NJ, 1994.

[22] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[23] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[24] G. Wahba, Splines Models for Observational Data, Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, PA, 1990.

[25] G. Wahba, 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, MIT Press, Cambridge, MA, 1999, pp. 68–88.

[26] C. Williams, Computing with infinite networks, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Neural Information Processing Systems, Vol. 9, MIT Press, Cambridge, MA, 1997, pp. 295–301.

[27] C. Williams, Prediction with Gaussian processes: from linear regression to linear prediction and beyond, in: M. Jordan (Ed.), Learning in Graphical Models, MIT Press, Cambridge, MA, 1998, pp. 599–621.

[28] C. Williams, D. Barber, Bayesian classification with Gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 1342–1351.

[29] C. Williams, C. Rasmuseen, Gaussian processes for regression, in: D. Touretzky, M. Mozer, M. Hasselmo (Eds.), Neural Information Processing Systems, Vol. 8, MIT Press, Cambridge, MA, 1997, pp. 514–520.

Junbin Gao graduated from Huazhong University of Science and Technology (HUST), China, in 1982 with B.Sc. degree in Computational Mathematics and obtained Ph.D. from the Institute of Mathematical Sciences at Dalian University of Technology, China. In November of 2001, he joined the School of Mathematical and Computer Sciences at University of New England as a lecturer in Computer Science. From 1982 to 2001 he was an associate lecturer, lecturer, associate professor and professor in Department of Mathematics at HUST. His main interests include machine learning, neural networks, signal processing and numerical analysis. Dr. Gao has published over 50 papers and one book on both data-based modelling, signal processing and numerical mathematics, etc.

Steve Gunn received his Beng. and Ph.D. (1996) from the University of Southampton. He is currently a lecturer in the Image, Speech and Intelligent Systems Research Group at the University of Southampton. His research interests include computer vision, active contours, kernel-based learning methods and their application to fields such as functional medical imaging, materials science, and character recognition.

Chris. J. Harris has degrees from the Universities of Leicester, Oxford and Southampton, currently he is the Head of Department of Electronics and Computer Science at Southampton University and a member of the ISIS research group. Prof. Harris has published over 300 papers and seven research books on Non-linear Systems and Control, his current interests are in neurofuzzy and data-based modelling, estimation, and data fusion for use in advanced transportation. He received the IEE Senior Achievement medal 1998 and the IEE Faraday medal in 2001 for this work. Prof. Harris is a Fellow of the Royal Academy of Engineering.