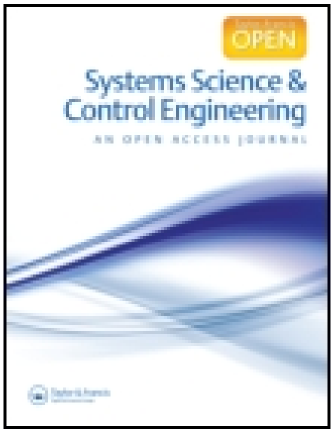
Estimation of Gaussian process regression model using probability distance measures



Click for updates

Systems Science & Control Engineering: An Open Access Journal
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tssc20

Estimation of Gaussian process regression model using probability distance measures

Xia Hong[a], Junbin Gao[b], Xinwei Jiang[c] & Chris J. Harris[d]

[a] School of Systems Engineering, University of Reading, Reading RG6 6AY, UK

[b] School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia

[c] School of Computer Science, China University of Geosciences, Wuhan 430074, China

[d] Electronics and Computer Science, University of Southampton, Southampton, UK

Published online: 31 Oct 2014.

To cite this article: Xia Hong, Junbin Gao, Xinwei Jiang & Chris J. Harris (2014) Estimation of Gaussian process regression model using probability distance measures, Systems Science & Control Engineering: An Open Access Journal, 2:1, 655-663, DOI: 10.1080/21642583.2014.970731

To link to this article: http://dx.doi.org/10.1080/21642583.2014.970731

Systems Science & Control Engineering: An Open Access Journal, 2014
Vol. 2, 655–663, http://dx.doi.org/10.1080/21642583.2014.970731

Taylor & Francis
Taylor & Francis Group

Estimation of Gaussian process regression model using probability distance measures

Xia Hong[a*], Junbin Gao[b], Xinwei Jiang[c] and Chris J. Harris[d]

[a]*School of Systems Engineering, University of Reading, Reading RG6 6AY, UK;* [b]*School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia;* [c]*School of Computer Science, China University of Geosciences, Wuhan 430074, China;* [d]*Electronics and Computer Science, University of Southampton, Southampton, UK*

(*Received 14 November 2013; final version received 25 September 2014*)

A new class of parameter estimation algorithms is introduced for Gaussian process regression (GPR) models. It is shown that the integration of the GPR model with probability distance measures of (i) the integrated square error and (ii) Kullback–Leibler (K–L) divergence are analytically tractable. An efficient coordinate descent algorithm is proposed to iteratively estimate the kernel width using golden section search which includes a fast gradient descent algorithm as an inner loop to estimate the noise variance. Numerical examples are included to demonstrate the effectiveness of the new identification approaches.

Keywords: Gaussian process; optimization; probability distance measures

1. Introduction

Regression models that define a system input/output relationship are widely used for system analysis and design in many scientific disciplines. Common regression models including linear models and nonlinear models such as neural networks are characterized by their use of a specific function to describe the system input/output relationship. The Gaussian process regression (GPR) model (Rasmussen, 2004; Rasmussen & Williams, 2006) is a nonparametric probabilistic model in which the system output is a data sample drawn from a Gaussian distribution conditional on its input. One can think of a Gaussian process as defining a distribution over functions, and inference taking place directly in the space of functions (Rasmussen, 2004). A Gaussian process is completely specified by its mean function and covariance function, and is defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. As such the functional mapping between the system input/output in GPR is assumed unknown, but a random function (an infinite dimensional vector), that is, a process with a specific covariance function of the input. The GPR is a powerful modeling tool since it predicts the output mean and variance conditional on a specific input at the same time. Clearly, this is an advantage over many other nonlinear functional-based modeling paradigms, for example, support vector regression (Schölkopf & Smola, 2002), which cannot quantify the uncertainties at the sample level. The GPR model has been successfully applied to a wide range of applications, for example, latent models for dimensionality reduction (Jiang, Gao, Wang, & Zheng, 2012; Lawrence, 2005) and modeling dynamical systems (Turner, Huber, Hanebeck, & Rasmussen, 2012).

The predictive output distribution of a GPR model is parameterized by a small number of parameters in the covariance function, which can be served by a typical kernel function, as well as the variance of additive noise, which can be regarded as one of the parameters. Typically, for a given data set the estimation of the GPR model also involves finding the most appropriate parameters, this is in general achieved by maximum log marginal likelihood or just maximum a posteriori estimation (Rasmussen & Williams, 2006). Alternatively, the GPR model estimation can be configured as a special type of probability density estimation problem concerning the conditional probability of an output variable. It is therefore a straightforward matter to construct objective functions based on the distance measure between the estimated output probability density function (pdf) for a given data set and an assumed true pdf. Well-known probability distance measures include the integrated square error (ISE) (Girolami & He, 2003; Hong et al., 2013; Silverman, 1986) which has been successfully applied in probability density estimation (Girolami & He, 2003; Silverman, 1986) and the Kullback–Leibler (K–L) divergence (Kullback & Leibler,

[*] Corresponding author. Email: x.hong@reading.ac.uk

1951) which is a widely used theoretic information metric for model selection (Burnham & Anderson, 2002).

In this paper, an efficient coordinate descent algorithm is proposed that iteratively estimates the kernel width using the golden section algorithm, which includes a gradient descent algorithm to rapidly update the noise variance. This paper is organized as follows. Section 2 introduces the GPR model. Section 3 formulates two types of probability distance measures based on ISE and K–L divergence based on a full GPR model. Section 4 introduces the proposed gradient descent algorithm for estimating the noise variance and the kernel width using ISE and K–L divergence metrics, respectively. Numerical experiments are utilized to illustrate the effectiveness of the proposed algorithm in Section 5 and our conclusions are given in Section 6.

2. Gaussian process regression

For a given data set $D_N = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where $\boldsymbol{x}_n \in \Re^m$, let $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^{\mathrm{T}}$ denote the observed input data matrix and also input space. $Y = [y_1, \ldots, y_N]^{\mathrm{T}}$ is an observed output vector and is also the output space. Define a kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ $(i,j = 1, \ldots N)$ on the input space X. In this study, the Gaussian kernel given by

$$k(\boldsymbol{u}, v) = \exp\left(-\frac{\|\boldsymbol{u} - v\|^2}{2\rho^2}\right) \tag{1}$$

is adopted, but other kernels can also be used. $\rho > 0$ is the width parameter.

Let $\mathcal{N}(\mu, \Sigma)$ denote the Gaussian distribution with mean μ and covariance Σ. In the classical GPR model, each sample y_n is generated based on

$$y = f(\boldsymbol{x}) + \epsilon, \tag{2}$$

where f is drawn from a (zero mean) Gaussian process $\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, K_{XX})$ which is dependent only on a specific covariance/kernel function $K_{XX} = \{k(\boldsymbol{x}_i, \boldsymbol{x}_j)\} \in \Re^{N\times N}$, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Denote $\boldsymbol{k}_{X\boldsymbol{x}} = [k(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x})]^{\mathrm{T}} \in \Re^N$.

The classical GPR aims to estimate the predictive distribution $p(y \mid \boldsymbol{x}^*)$ for any test data $\boldsymbol{x}^* \in X$. Consider a new test observation $\boldsymbol{x}^*$. Under the Gaussian likelihood assumption, it is easy to prove (Rasmussen & Williams, 2006) that the estimated predictive distribution conditioned on the given observation is

$$\hat{p}(y \mid \boldsymbol{x}^*, X, Y) \sim \mathcal{N}(f(\boldsymbol{x}^*), g(\boldsymbol{x}^*)), \tag{3}$$

where

$$f(\boldsymbol{x}^*) = \boldsymbol{k}_{X\boldsymbol{x}^*}^{\mathrm{T}}(K_{XX} + \sigma^2\mathbf{I})^{-1}Y, \tag{4}$$

$$g(\boldsymbol{x}^*) = \sigma^2 + k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}_{X\boldsymbol{x}^*}^{\mathrm{T}}(K_{XX} + \sigma^2\mathbf{I})^{-1}\boldsymbol{k}_{X\boldsymbol{x}^*} \tag{5}$$

with $\mathbf{I}$ denoting identity matrix with appropriate dimension. Specifically, let $\alpha = [\alpha_1, \ldots, \alpha_N]^{\mathrm{T}} = (K_{XX} + \sigma^2\mathbf{I})^{-1}Y$. The mean of Equation (3) can be written as

$$f(\boldsymbol{x}^*) = \alpha^{\mathrm{T}}\boldsymbol{k}_{X\boldsymbol{x}^*} = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}^*), \tag{6}$$

This form of the prediction exhibits the fact that a GP can be represented in terms of a number of basis functions according to the representer theorem (Schölkopf & Smola, 2002).

The marginal likelihood $p(Y \mid X)$ is the integral of the likelihood times the prior

$$p(Y \mid X) = \int p(Y \mid \boldsymbol{f}, X)p(\boldsymbol{f} \mid X)\,\mathrm{d}\boldsymbol{f} \tag{7}$$

and the log marginal likelihood is given by Rasmussen & Williams (2006) as

$$\begin{aligned} J^{\mathrm{ML}} = \log p(Y \mid X) &= -\frac{1}{2}Y^{\mathrm{T}}(K_{XX} + \sigma^2\mathbf{I})^{-1}Y \\ &\quad -\frac{1}{2}\log\det(K_{XX} + \sigma^2\mathbf{I}) - \frac{N}{2}\log(2\pi) \end{aligned} \tag{8}$$

which is the mostly used criterion for the estimation of GPR model parameters.

3. Probability distance measures for GPR

Probability distance measures are similarity metrics between two pdfs. In this paper, we propose to identify GPR model using the distance between the true output pdf and its estimator. We use data set X as prior for a GPR model, which predicts the conditional probability of the system output induced by any input vector in X. Not all pdf distance measures are tractable since the true probability is always unknown. In the following, we formulate two tractable cost functions that are related to the ISE and the K–L divergence for an output probability density estimator based on the GPR model.

3.1. The minimum integrated square error

The minimum integrated square error (MISE) between a pdf estimator and the true density is a classical goodness-of-fit criterion of probability density estimation, both for nonparametric (Girolami & He, 2003; Silverman, 1986) and for parametric models (Scott, 2001). The ISE for the GPR pdf estimator is given by

$$\begin{aligned} \mathrm{ISE} &= \int (p(y) - \hat{p}(y \mid X, Y))^2\,\mathrm{d}y \\ &= \int (p(y))^2\,\mathrm{d}y - 2\int \hat{p}(y \mid X, Y)p(y)\,\mathrm{d}y \\ &\quad + \int (\hat{p}(y \mid X, Y))^2\,\mathrm{d}y \\ &= \int (p(y))^2\,\mathrm{d}y + Q, \end{aligned} \tag{9}$$

where Q can be used as the cost function instead of ISE since $\int (p(y))^2\,\mathrm{d}y$ does not contain adjustable parameters.

Consider the problem of estimating $\hat{p}(y \mid X, Y)$ based on given data set $\{X, Y\}$, we have

$$\begin{aligned}\hat{p}(y \mid X, Y) &= \int \hat{p}(y \mid \boldsymbol{x}, X, Y) p(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \\ &= E(\hat{p}(y \mid \boldsymbol{x}, X, Y)) \\ &\approx \frac{1}{N}\sum_{j=1}^{N} \frac{1}{\sqrt{2\pi g(\boldsymbol{x}_j)}} \exp\left(-\frac{(y - f(\boldsymbol{x}_j))^2}{2\,g(\boldsymbol{x}_j)}\right).\end{aligned} \tag{10}$$

Here, we applied the well-known Bayesian rule, and then the principle of the plug-in estimator which states that sample average can be used to approximate an expected value when the true density is unknown.

By making use of Equation (10) and also the principle of plug-in estimation for $\int \hat{p}(y \mid \boldsymbol{x}_j, X, Y) p(y)\,\mathrm{d}y$ with respect to the true density $p(y)$, we have

$$\begin{aligned}Q &= \int \left(\frac{1}{N}\sum_{j=1}^{N} \frac{1}{\sqrt{2\pi g(\boldsymbol{x}_j)}} \exp\left(-\frac{(y - f(\boldsymbol{x}_j))^2}{2\,g(\boldsymbol{x}_j)}\right)\right)^2 \mathrm{d}y \\ &\quad - 2\int \hat{p}(y \mid X, Y) p(y)\,\mathrm{d}y \\ &\approx J^{\mathrm{ISE}} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{1}{2\pi\sqrt{g(\boldsymbol{x}_i) g(\boldsymbol{x}_j)}} \\ &\quad \times \int \exp\left(-\frac{(y - f(\boldsymbol{x}_i))^2}{2\,g(\boldsymbol{x}_i)} - \frac{(y - f(\boldsymbol{x}_j))^2}{2\,g(\boldsymbol{x}_j)}\right) \mathrm{d}y \\ &\quad - \frac{2}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \frac{1}{\sqrt{2\pi g(\boldsymbol{x}_j)}} \exp\left(-\frac{(y_i - f(\boldsymbol{x}_j))^2}{2\,g(\boldsymbol{x}_j)}\right) \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} (q_{i,j} - 2p_{i,j})\end{aligned} \tag{11}$$

in which

$$\begin{aligned}q_{i,j} &= \frac{1}{\sqrt{2\pi(g_i + g_j)}} \exp\left(-\frac{(f_i - f_j)^2}{2(g_i + g_j)}\right), \\ p_{i,j} &= \frac{1}{\sqrt{2\pi g_j}} \exp\left(-\frac{e_{i,j}^2}{2g_j}\right),\end{aligned} \tag{12}$$

are used for brevity and the appendix was applied to generate Equation (11). Also, $e_{i,j} = y_i - f(\boldsymbol{x}_j)$. f_i, g_i denote $f(\boldsymbol{x}_i)$ and $g(\boldsymbol{x}_i)$, respectively. Note that the plug-in estimator can be fully justified since the approximation error is asymptotically in the order of $N^{-1/2}$ for many classes of probability functions (van der Vaart, 2000).

3.2. K–L divergence

Similarly, we derive a cost function based on the Kullback–Leibler divergence (Kullback & Leibler, 1951), given by

$$\begin{aligned}\mathrm{KL} &= \int p(y) \log \frac{p(y)}{\hat{p}(y \mid X, Y)}\,\mathrm{d}y \\ &= \int p(y) \log p(y)\,\mathrm{d}y - \int \log \hat{p}(y \mid X, Y) p(y)\,\mathrm{d}y\end{aligned} \tag{13}$$

in which the second term $R = \int \log \hat{p}(y \mid X, Y) p(y)\,\mathrm{d}y \approx E(\log \hat{p}(y \mid X, Y))$ needs to be maximized. Applying Equation (10), we have

$$\begin{aligned}R &\approx J^{\mathrm{KL}} \\ &= \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{N}\sum_{j=1}^{N} \frac{1}{\sqrt{2\pi g(\boldsymbol{x}_j)}} \exp\left(-\frac{(y_i - f(\boldsymbol{x}_j))^2}{2\,g(\boldsymbol{x}_j)}\right)\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{N}\sum_{j=1}^{N} p_{i,j}\right).\end{aligned} \tag{14}$$

Remark 1 The Kullback–Leibler divergence can also be defined by swapping the order of the true density and the estimated density in Equation (13), yet it will become analytically intractable since the expectation needs to be taken with respective to the estimated probability.

Remark 2 For both the second line of Equation (13) of KL expression and the first line of Equation (11) of Q expression, their second terms are similar in that the expectation of either the estimated probability or its log needs to be estimated with respect to the true density. This feature leads to their computational tractability which is absent in other distance measures. For example, we cannot easily calculate the Hellinger distance from the data.

Remark 3 In Equation (14), since nothing is known about the true density $p(y)$, we also approximate KL using the well-known principle of plug-in estimator, similar to ISE.

Remark 4 We do not claim superiority of proposed metrics over the well established J^{KL}. Clearly, the proposed metrics are based on sample average, while the latter is based on a closed-form solution of integration on functional space, which makes it very difficult to carry out functional analysis on their differences. This issue is an open problem.

4. Parameter estimation using coordinate descent algorithms for GPR models

In the GPR model estimation, the variance of noise is usually regarded as a parameter and catenated with a small number of parameters in the kernel function, and can be jointly estimated via maximizing the log marginal likelihood J^{ML} given by Equation (8). In this work, we propose the idea of GPR parameter estimation based on either minimizing J^{ISE} or maximizing J^{KL}, which can be achieved by using a gradient descent algorithm for a local optimum. However, the computational cost for each iteration is in the order of $O(N^3)$ due to a matrix inversion in calculating gradient direction. Alternatively, it is computationally cheaper to apply coordinate descent algorithm to search for ρ and σ^2, one at a time.

Consider solving σ^2 for a fixed ρ using the following the gradient descent algorithm. With an initial σ^2_{old}, the gradient descent algorithm for minimizing J^{ISE} of Equation (11) is given as follows:

$$\begin{aligned}\sigma^2_{\text{new}} &= \max\left\{\sigma^2_{\min}, \sigma^2_{\text{old}} - \eta\cdot\text{sign}\left(\left.\frac{\partial J^{\text{ISE}}}{\partial\sigma^2}\right|_{\sigma=\sigma_{\text{old}}}\right)\right\},\\ \sigma^2_{\text{old}} &= \sigma^2_{\text{new}},\end{aligned} \tag{15}$$

where $\eta > 0$ is a very small positive learning rate. $\sigma_{\min} > 0$ is set as a very small number to improve numerical stability. Note that $\text{sign}(\partial J^{\text{ISE}}/\partial\sigma^2)$ is used in Equation (15), indicating that this is a normalized version of gradient descent algorithm and a small learning rate η will scale well with the search space of σ^2, irrespective of the actual size of $\partial J^{\text{ISE}}/\partial\sigma^2$. Equation (15) is repeated until $\text{sign}(\partial J^{\text{ISE}}/\partial\sigma^2)$ for two consecutive steps are different indicating a local minimum of J^{ISE}, or when a preset number of iterations It is reached. For example, It = 100. From Equation (11), we obtain

$$\frac{\partial J^{\text{ISE}}}{\partial\sigma^2} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial q_{i,j}}{\partial\sigma^2} - 2\frac{\partial p_{i,j}}{\partial\sigma^2}\right) \tag{16}$$

in which

$$\begin{aligned}\frac{\partial q_{i,j}}{\partial\sigma^2} &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\mu_{i,j}^2}{2\nu_{i,j}}\right)\\ &\quad\times\left\{\mu_{i,j}\nu_{i,j}^{-3/2}\left(\frac{\partial}{\partial\sigma^2}f(\boldsymbol{x}_j) - \frac{\partial}{\partial\sigma^2}f(\boldsymbol{x}_i)\right)\right.\\ &\quad+0.5(\mu_{i,j}^2\nu_{i,j}^{-5/2} - \nu_{i,j}^{-3/2})\\ &\quad\times\left.\left(\frac{\partial}{\partial\sigma^2}g(\boldsymbol{x}_i) + \frac{\partial}{\partial\sigma^2}g(\boldsymbol{x}_j)\right)\right\}\end{aligned} \tag{17}$$

with $\mu_{i,j} = f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)$, $\nu_{i,j} = g(\boldsymbol{x}_i) + g(\boldsymbol{x}_j)$, and

$$\begin{aligned}\frac{\partial p_{i,j}}{\partial\sigma^2} &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{e_{i,j}^2}{2g_j}\right)\left(e_{i,j}g_j^{-3/2}\frac{\partial}{\partial\sigma^2}f(\boldsymbol{x}_j)\right.\\ &\quad\left.+0.5(e_{i,j}^2g_j^{-5/2} - g_j^{-3/2})\frac{\partial}{\partial\sigma^2}g(\boldsymbol{x}_j)\right).\end{aligned} \tag{18}$$

The required $f(\boldsymbol{x}_j)$, $g(\boldsymbol{x}_j)$ and their gradients can be calculated efficiently as follows.

Write $K_{XX} = \sum_{n=1}^{R}s_n\boldsymbol{u}_n\boldsymbol{u}_n^{\text{T}}$, where $s_1 \geq s_2 \geq \cdots \geq s_R > 0$ are $R \leq N$ nonzero singular values of K_{XX}, and $\boldsymbol{u}_n$, $n = 1,\ldots,R$ are the first R singular vectors (R can be found by rank.m). It can be verified that for $j = 1,\ldots N$,

$$f(\boldsymbol{x}_j) = \boldsymbol{k}_{X\boldsymbol{x}_j}^{\text{T}}(K_{XX} + \sigma^2\mathbf{I})^{-1}Y = \sum_{n=1}^{R}\frac{\alpha_{j,n}\beta_n}{s_n + \sigma^2}, \tag{19}$$

$$\begin{aligned}\frac{\partial}{\partial\sigma^2}f(\boldsymbol{x}_j) &= -\boldsymbol{k}_{X\boldsymbol{x}_j}^{\text{T}}(K_{XX} + \sigma^2\mathbf{I})^{-2}Y\\ &= -\sum_{n=1}^{R}\frac{\alpha_{j,n}\beta_n}{(s_n + \sigma^2)^2},\end{aligned} \tag{20}$$

$$\begin{aligned}g(\boldsymbol{x}_j) &= 1 + \sigma^2 - \boldsymbol{k}_{X\boldsymbol{x}_j}^{\text{T}}(K_{XX} + \sigma^2\mathbf{I})^{-1}\boldsymbol{k}_{X\boldsymbol{x}_j}\\ &= 1 + \sigma^2 - \sum_{n=1}^{R}\frac{\alpha_{j,n}^2}{s_n + \sigma^2},\end{aligned} \tag{21}$$

$$\begin{aligned}\frac{\partial}{\partial\sigma^2}g(\boldsymbol{x}_j) &= 1 + \boldsymbol{k}_{X\boldsymbol{x}_j}^{\text{T}}(K_{XX} + \sigma^2\mathbf{I})^{-2}\boldsymbol{k}_{X\boldsymbol{x}_j}\\ &= 1 + \sum_{n=1}^{R}\frac{\alpha_{j,n}^2}{(s_n + \sigma^2)^2},\end{aligned} \tag{22}$$

where $\alpha_{j,n} = \boldsymbol{u}_n^{\text{T}}\boldsymbol{k}_{X\boldsymbol{x}_j}$ and $\beta_n = \boldsymbol{u}_n^{\text{T}}Y$. Note that each iteration of Equation (15) requires Equations (19)–(22) to be updated at $O(N)$ complexity except for the first iteration. This is because only σ^2 is changed in the right-hand side of the equations over each iteration. During the first iteration the singular value decomposition (SVD) of K_{XX} is made at a cost of $O(N^3)$.

Similarly for the maximization of J^{KL}, we use the iteration

$$\begin{aligned}\sigma^2_{\text{new}} &= \max\left\{\sigma^2_{\min}, \sigma^2_{\text{old}} + \eta\cdot\text{sign}\left(\left.\frac{\partial J^{\text{KL}}}{\partial\sigma^2}\right|_{\sigma=\sigma_{\text{old}}}\right)\right\},\\ \sigma^2_{\text{old}} &= \sigma^2_{\text{new}},\end{aligned} \tag{23}$$

until either a preset number of It iterations is reached or when $\text{sign}(\partial J^{\text{ISE}}/\partial\sigma^2)$ is different for two consecutive steps, where

$$\frac{\partial J^{\text{KL}}}{\partial\sigma^2} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sum_{j=1}^{N}p_{i,j}}\sum_{j=1}^{N}\frac{\partial p_{i,j}}{\partial\sigma^2} \tag{24}$$

Note that Equation (24) uses Equation (18) which in turn also utilizes Equations (19)–(22) for efficient computation as the case of minimizing J^{ISE}.

Similarly for comparative study we also consider the gradient descent algorithm based on J^{ML} for a fixed ρ. By applying the SVD of $K_{XX} = \sum_{n=1}^{R}s_n\boldsymbol{u}_n\boldsymbol{u}_n^{\text{T}}$ to Equation (8)

$$\begin{aligned}J^{\text{ML}} &= -\frac{1}{2}Y^{\text{T}}(K_{XX} + \sigma^2\mathbf{I})^{-1}Y - \frac{1}{2}\log\det(K_{XX} + \sigma^2\mathbf{I})\\ &\quad-\frac{N}{2}\log(2\pi)\end{aligned}$$

$$= -\frac{1}{2}\sum_{n=1}^{R}\frac{\beta_n^2}{s_n+\sigma^2} - \frac{1}{2}\sum_{n=1}^{N}\log(s_n+\sigma^2) - \frac{N}{2}\log(2\pi), \tag{25}$$

where $s_n = 0$ if $n > R$. Thus

$$\frac{\partial J^{\mathrm{ML}}}{\partial \sigma^2} = \frac{1}{2}\sum_{n=1}^{R}\frac{\beta_n^2}{(s_n+\sigma^2)^2} - \frac{1}{2}\sum_{n=1}^{N}\frac{1}{s_n+\sigma^2}. \tag{26}$$

For completeness the gradient decent iteration for J^{ML} is given as the iteration

$$\sigma_{\mathrm{new}}^2 = \max\left\{\sigma_{\min}^2, \sigma_{\mathrm{old}}^2 + \eta\cdot\mathrm{sign}\left(\left.\frac{\partial J^{\mathrm{ML}}}{\partial\sigma^2}\right|_{\sigma=\sigma_{\mathrm{old}}}\right)\right\},$$
$$\sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{new}}^2, \tag{27}$$

until either a preset number of It iterations is reached or when $\mathrm{sign}(\partial J^{\mathrm{ML}}/\partial\sigma^2)$ is different for two consecutive steps.

Now consider the optimization of J^{ISE} or J^{KL} with respect to ρ, which affects all elements in K_{XX}. Their gradients are not only much more computationally expensive, but also complex. However, since there is only one variable to optimize for fixed σ^2, we opt to use the golden section search (Venkataraman, 2002) as the outer loop which directly evaluates J^{ISE} or J^{KL} for fixed ρ and the resultant σ^2 obtained using the gradient descent algorithms of Equation (15) or Equation (23) as its inner loop. Similarly, maximization J^{ML} can use the same framework, so next we present the general scheme of an optimization algorithm that can deal with all the three criteria we mentioned.

Pseudocode of the golden section-based coordinate descent algorithm

```
Predetermine $\rho_{max}$, $\rho_{min}$ {Search range for $\rho$}
Initialize $\sigma_{\mathrm{old}}^2$
$\tau = 0.3897; \varepsilon = 0.1$ {Precision}
$n_{max} = -2.078\log\left(\varepsilon/(\rho_{max}-\rho_{min})\right)$ {The number of iterations}
$\rho_1 = (1-\tau)\rho_{min} + \tau\rho_{max}$
Obtain $\sigma_{\mathrm{old1}}$ in response to $\rho = \rho_1$ using gradient descent algorithm (15) or (23) or (27)
Obtain $J_1^{\mathrm{ISE}}$ in response to $\rho = \rho_1, \sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{old1}}^2$ using (11) (or $J_1^{\mathrm{KL}}$ via (14); or $J_1^{\mathrm{ML}}$ via (25) )
$\rho_2 = \tau\rho_{min} + (1-\tau)\rho_{max}$
Obtain $\sigma_{\mathrm{old2}}^2$ in response to $\rho = \rho_2$ using gradient descent algorithm (15) or (23) or (27)
Obtain $J_2^{\mathrm{ISE}}$ in response to $\rho = \rho_2, \sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{old2}}^2$ using (11) (or $J_2^{\mathrm{KL}}$ via (14); or $J_2^{\mathrm{ML}}$ via (25))
for $n = 1 \to n_{max}$ do
    if $J_2^{\mathrm{ISE}} < J_1^{\mathrm{ISE}}$ (or $J_2^{\mathrm{KL}} > J_1^{\mathrm{KL}}$ or $J_2^{\mathrm{ML}} > J_1^{\mathrm{ML}}$) then
        $\rho_{min} \leftarrow \rho_1$; $\rho_1 \leftarrow \rho_2$ ; $J_1^{\mathrm{ISE}} \leftarrow J_2^{\mathrm{ISE}}$; (or $J_1^{\mathrm{KL}} \leftarrow J_2^{\mathrm{KL}}$ or $J_1^{\mathrm{ML}} \leftarrow J_2^{\mathrm{ML}}$ )
        $\rho_2 = \tau\rho_{min} + (1-\tau)\rho_{max}$
        Initialize $\sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{old2}}^2$
        Obtain $\sigma_{\mathrm{old2}}^2$ in response to $\rho = \rho_2$ using gradient descent algorithm (15) or (23) or (27)
        Obtain $J_2^{\mathrm{ISE}}$ in response to $\rho = \rho_2, \sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{old2}}^2$ using (11) (or $J_2^{\mathrm{KL}}$ via (14); or $J_2^{\mathrm{ML}}$ via (25) )
    end if
    if $J_1^{\mathrm{ISE}} < J_2^{\mathrm{ISE}}$ ( or $J_1^{\mathrm{KL}} > J_2^{\mathrm{KL}}$ or $J_1^{\mathrm{ML}} > J_2^{\mathrm{ML}}$) then
        $\rho_{max} \leftarrow \rho_2$; $\rho_2 \leftarrow \rho_1$ ; $J_2^{\mathrm{ISE}} \leftarrow J_1^{\mathrm{ISE}}$; (or $J_2^{\mathrm{KL}} \leftarrow J_1^{\mathrm{KL}}$; or $J_2^{\mathrm{ML}} \leftarrow J_1^{\mathrm{ML}}$ )
        $\rho_1 = (1-\tau)\rho_{min} + \tau\rho_{max}$
        Initialize $\sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{old1}}^2$
        Obtain $\sigma_{\mathrm{old1}}$ in response to $\rho = \rho_1$ using gradient descent algorithm (15) or (23) or (27)
        Obtain $J_1^{\mathrm{ISE}}$ in response to $\rho = \rho_1, \sigma_{\mathrm{old}}^2 = \sigma_{\mathrm{old1}}^2$ using (11) (or $J_1^{\mathrm{KL}}$ via (14); or $J_2^{\mathrm{ML}}$ via (25) )
    end if
    $n \leftarrow n + 1$
end for
Obtain $\rho_{optimal} = (\rho_1 + \rho_2)/2$
Obtain the final $\sigma^2$ in response to $\rho = \rho_{optimal}$ using gradient descent algorithm (15) or (23) or (27)
```

Remark 5 The gradient descent algorithm using only first-order derivatives has slow convergence. In this application, the second-order derivatives with respect to the parameters in the kernel function are very involved. The golden section algorithm is a derivative free optimization technique but only suitable for one-dimensional variable search. If the kernel functions have more parameters, other random search algorithms such as particle swarm optimization algorithms (Kennedy & Eberhart, 2001) are recommended which can obtain a suboptimal solution without finding derivatives for solving the parameters in the kernel function.

Remark 6 The proposed algorithm converges to a local minimum. This is because the cost functions are highly nonlinear functions.

5. Simulation study

Example 1 (1D Scalar Function) Consider using the GP model to approximate an unknown scalar function

$$f(x) = \frac{\sin(x)}{x}. \tag{28}$$

A data set of 200 points was generated from $y = f(x) + \xi$, where the input x was uniformly distributed in $[-10, 10]$ and the noise $\xi \sim N(0, 0.04)$. The data were very noisy. The Gaussian kernel of Equation (1) was used. The coordinate descent algorithms based on either J^{ISE} or J^{KL} were

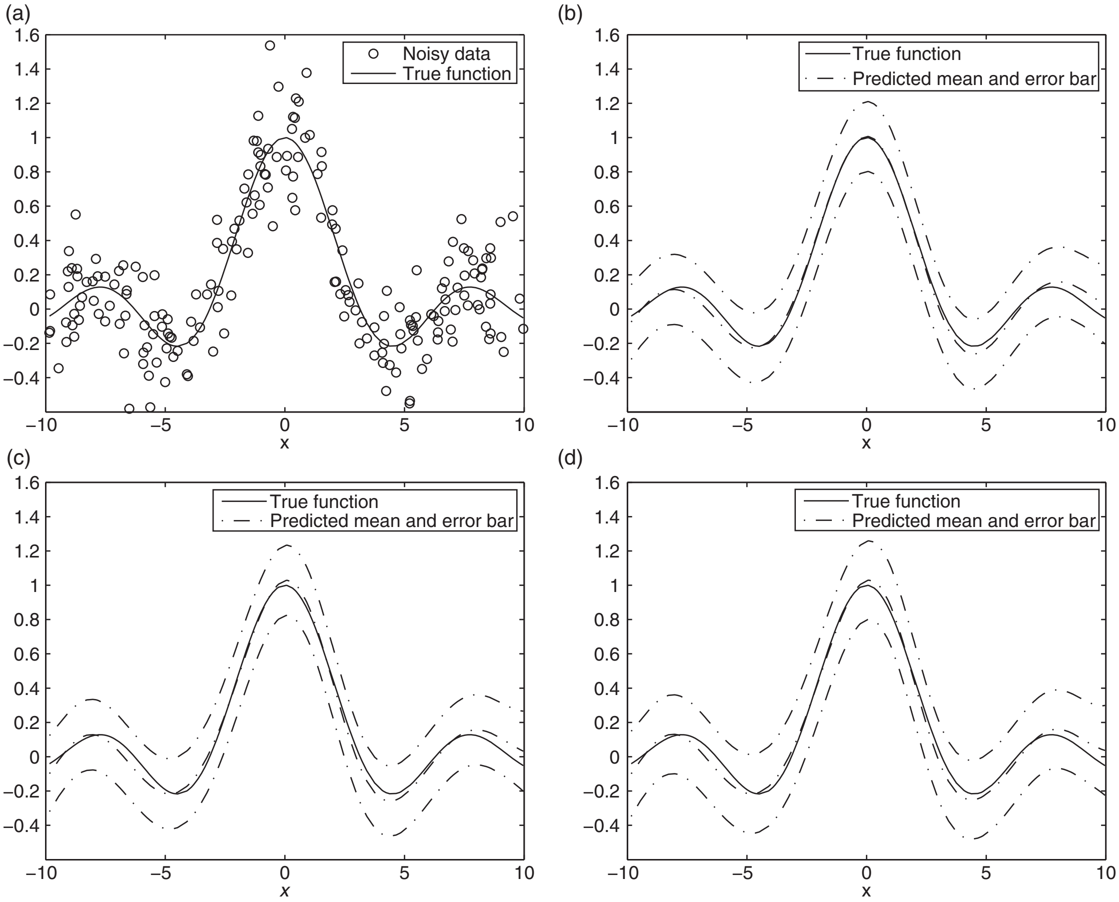


Figure 1. The modeling results of 1-D scalar function problem; (a) True function and noisy data; (b) GPR model prediction based on J^{ISE} for $f(x) \pm \sqrt{g(x)}$; (c) GPR model prediction based on J^{KL} for $f(x) \pm \sqrt{g(x)}$; and (d) GPR model prediction based on J^{ML} for $f(x) \pm \sqrt{g(x)}$.

applied to jointly estimate the kernel parameter ρ as well as the noise variance σ^2. For comparison, the well-known maximization of log marginal likelihood J^{ML} criterion is also experimented. The same parameter settings were used for three algorithms. The search range is set $[\rho_{\min}, \rho_{\max}] = [2, 5]$, and the variance σ^2_{old} was initialized as 0.5. The learning rate was set as $\eta = 0.01$, and the maximum iteration in the gradient algorithm $\mathrm{It} = 100$. A maximum of five iterations was set for golden section search. The search range of ρ was determined empirically for this example. Because the cost functions are multimodal the solutions are only locally optimum. The estimated mean function for $f(x)$ are plotted in Figure 1(b)–(d), respectively, for three resultant GPR models. The modeling results were given in Table 1 demonstrating that the obtained GPR models are comparable with an excellent capability to approximate the underlying true function.

Table 1. Comparison of modeling performance for the scalar function; (a) Example 1; and (b) Example 2.

	MSE over noisy outputs	MSE over true function
(a)		
GPR via J^{ISE}	0.0563	0.0013
GPR via J^{KL}	0.0558	0.0015
GPR via J^{ML}	0.0559	0.0016
(b)		
GPR via J^{ISE}	0.0092	6.41×10^{-4}
GPR via J^{KL}	0.0092	6.46×10^{-4}
GPR via J^{ML}	0.0088	6.22×10^{-4}

Example 2 (2D Scalar Function) The Matlab logo was generated by the first eigenfunction of the L-shaped membrane. A 31×31 meshed data set $f(x_1, x_2)$ was generated by using Matlab command membrane.*m*, which is defined over a unit square input region $x_1 \in [0, 1]$ and $x_2 \in [0, 1]$. The $N = 961$ sized data set $y(x_1, x_2) = f(x_1, x_2) + e(x_1, x_2)$ was then generated by adding a noise term $e(x_1, x_2) \sim N(0, 0.01)$. The noisy data and true function are plotted in Figure 2(a) and 2(b), respectively. By using the Gaussian kernel of Equation (1) over $x_1 \in [0, 1]$ and $x_2 \in [0, 1]$, the coordinate descent algorithms based on J^{ISE}, J^{KL} and J^{ML} were applied to jointly estimate the

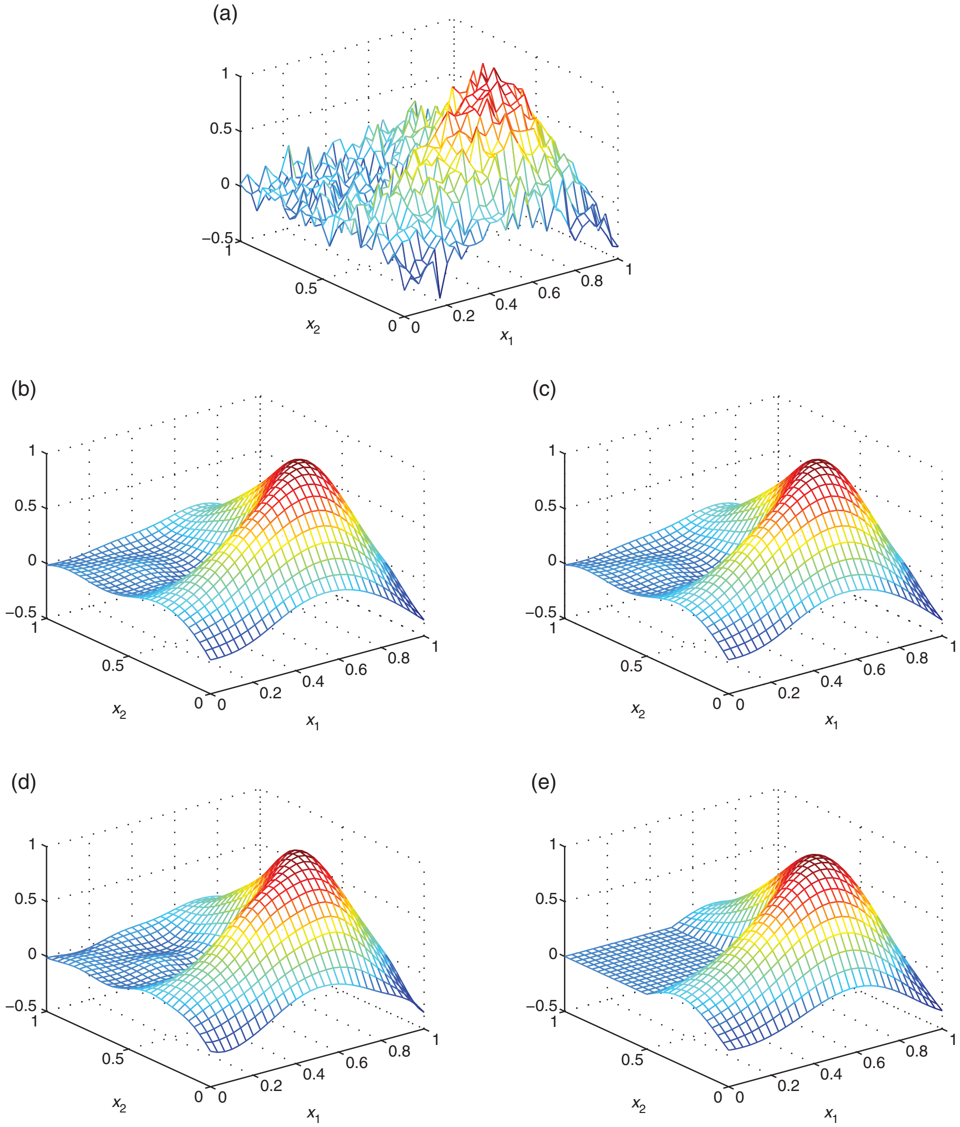


Figure 2. The modeling results of 2-D scalar function problem; (a) Noisy data; (b) True function; (c) GPR model prediction for $f(x_1, x_2)$ based on J^{ISE}; (d) GPR model prediction for $f(x_1, x_2)$ based on J^{KL}; and (e) GPR model prediction for $f(x_1, x_2)$ based on J^{ML}.

kernel parameter ρ as well as the noise variance σ^2. For all three algorithms, we preset the search range $[\rho_{\min}, \rho_{\max}] = [0.2, 0.4]$ and initialized the variance σ^2_{old} as 0.1, the learning rate was set as $\eta = 0.005$, and the maximum iteration in the gradient algorithm It = 100. A maximum of five iterations was set for golden section search. The estimated mean function for $f(x_1, x_2)$ are plotted in Figure 2(c)–(e), respectively, for three resultant GPR models. The modeling results were given in Table 1 demonstrating that they have comparable performance and can provide an excellent approximation to the true underlying function.

Example 3 (Boston Housing Data) This is a classic regression benchmark data set, available at the University of California, Irvine (UCI) repository (Frank & Asuncion, 2010). The data set comprises 506 data points with 14 variables. We perform the task of predicting the median house value from the remaining 13 attributes. The GPR model parameters were estimated based on the whole 506 data samples, in which the 13 attributes were normalized so that each attribute has zero mean, and standard deviation of one. The Gaussian kernel of Equation (1) based on the normalized 13 features was used. Similar to the previous examples, the coordinate descent algorithms based on J^{ISE}, J^{KL} , and J^{ML} were applied over the search range $[\rho_{\min}, \rho_{\max}] = [2, 5]$. σ^2_{old} was intialized as 0.1. The maximum iteration number in gradient descent algorithm was set as It = 100. A maximum of five iterations was set for golden section search. For both J^{ISE} and J^{KL} based algorithms, the learning rate was set as $\eta = 0.001$, but for J^{KL} the result of $\eta = 0.0001$ was used since it gives better performance. For 100 realizations, we randomly selected 456

Table 2. Comparison of modeling performance for Boston House Data.

	MSE over training data set	MSE over test data set	Model size
ε-SVM (Chen et al., 2009)	6.80 ± 0.44	23.18 ± 9.05	243 ± 5.3
LROLS-LOO (Chen et al., 2009)	12.97 ± 2.67	17.42 ± 4.67	58.6 ± 11.3
OFS-LOO (Chen et al., 2009)	10.10 ± 3.40	14.07 ± 3.62	34.6 ± 8.4
GPR via J^{ISE}	4.07 ± 0.28	9.46 ± 5.54	456 ± 0
GPR via J^{KL}	4.01 ± 0.28	9.62 ± 5.72	456 ± 0
GPR via J^{ML}	4.75 ± 6.93	11.31 ± 6.93	456 ± 0

Notes: The results of were averaged over 100 realizations and given as mean ± standard deviation. The results of ε-SVM, LROLS-LOO, and OFS-LOO were quoted from (Chen et al., 2009).

data points from the data set, from which the 13 features are normalized so that each attribute has zero mean, and a standard deviation of one, which is used to construct a covariance matrix based on the learnt parameters. The remaining 50 data points forms the test set. Average results were given over 100 realizations, and given in Table 2 and are compared with a few known nonlinear regression methods of the ε-SVM (Gun, 1998), the LROLS-LOO (Chen, Hong, Harris, & Sharkey, 2004) and the OFS-LOO (Chen, Hong, & Harris, 2009). The details of the experimental settings of these algorithms can be found in Chen et al. (2009). The large sized GPR models have better performance in general, with the two GPR models based on J^{ISE} and J^{KL}, respectively, are shown to have superior performance for this particular example.

6. Conclusions

New parameter estimation algorithms have been introduced for GPR models. Our original contribution here is to integrate the probability distance measures of the ISE and K–L divergence with GPR model as new cost functions for GPR parameter estimation. By using a kernel width as the single parameter in the covariance function, we iteratively estimate the kernel width using golden section search which has an inner loop of fast gradient descent algorithm to estimate the noise variance. Numerical examples have been utilized to demonstrate the effectiveness of the new identification approaches in comparison with the well-known maximal log marginal likelihood cost criterion, and it is shown that ISE and K–L divergence are effective alternative cost functions for GPR models.

Acknowledgements

Junbin Gao and Xia Hong acknowledge the support of ARC under Grant DP130100364.

References

Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. New York: Springer Science.

Chen, S., Hong, X., & Harris, C. J. (2009). Construction of tunable radial basis function networks using orthogonal forward selection. *IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics*, *39*(2), 457–466.

Chen, S., Hong, X., Harris, C. J., & Sharkey, P. M. (2004). Sparse modelling using orthogonal forward regression with PRESS statistic and regularization. *IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics*, *34*(2), 898–911.

Frank, A., & Asuncion, A. (2010). *UCI machine learning repository*. Retrieved from http://archive.ics.uci.edu/ml.

Girolami, M., & He, C. (2003). Probability density estimation from optimally condensed data samples. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *25*(10), 1253–1264.

Gun, S. R. (1998). *Support vector machines for classification and regression*. Southampton: ISIS Research Group, Department Electronics Computer Science, University of Southampton.

Hong, X., Chen, S., Qatawneh, A., Daqrouq, K., Sheikh, M., & Morfeq, A. (2013). Sparse probability density function estimation using the minimum integrated square error. *Neurocomputing*, *114*, 122–129.

Kennedy, J., & Eberhart, R. C. (2001). *Swarm intelligence*. San Francisco, CA: Morgan Kaufmann.

Jiang, X., Gao, J., Wang, T., & Zheng, L. (2012). Supervised latent linear Gaussian process latent variable model for dimensionality reduction. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, *42*(6), 1620–1632.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, *6*, 1783–1816.

Rasmussen, C. E. (2004). Gaussian processes in machine learning. In *Advanced lectures on machine learning.* Lecture notes in computer science (Vol. 3176, pp. 63–71). New York: Springer.

Rasmussen, C. C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press.

Schölkopf, B., & Smola, A. (2002). *Learning with Kernels*. Cambridge, MA: The MIT Press.

Scott, S. W. (2001). Parametric statistical modeling by minimum integrated square error. *Technometrics*, *43*(3), 274–285.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis.* London: Chapman and Hall.

Turner, R. D., Huber, M. F., Hanebeck, U. D., & Rasmussen, C. E. (2012). Robust filtering and smoothing with gaussian processes. *IEEE Transactions of Automatic Control*, *57*(7), 1865–1871.

van der Vaart, A. W. (2000). *Asymptotic statistics*. Cambridge: Cambridge University Press.

Venkataraman, P. (2002). *Applied optimization with MATLAB programming*. New York, NY: Wiley-Interscience.

Appendix. Integrating $\int \exp(-(y-f(x_i))^2/2g(x_i) - (y-f(x_j))^2/2g(x_j))\,\mathrm{d}y$ in Equation (11)

For brevity, let f_i, g_i denote $f(x_i)$ and $g(x_i)$, respectively.

$$\int \exp\left(-\frac{(y-f_i)^2}{2g_i} - \frac{(y-f_j)^2}{2g_j}\right)\mathrm{d}y$$

$$= \int \exp\left(-\frac{\begin{array}{c}(g_i+g_j)y^2 - 2(f_ig_j+f_jg_i)y\\ +f_i{}^2g_j+f_j{}^2g_i\end{array}}{2g_ig_i}\right)\mathrm{d}y$$

$$= \exp\left(-\frac{\begin{array}{c}(f_j{}^2g_i+f_i{}^2g_j)/(g_i+g_j)\\ -((f_ig_j+f_jg_i)/(g_i+g_j))^2\end{array}}{2g_ig_j/(g_i+g_j)}\right)$$

$$\times \int \exp\left(-\frac{[y-(f_ig_j+f_jg_i)/(g_i+g_j)]^2}{2g_ig_j/(g_i+g_j)}\right)\mathrm{d}y \tag{A1}$$

By making use of $\int (1/\sqrt{2\pi\Sigma})\exp(-(y-\mu)^2/2\Sigma)\,\mathrm{d}y = 1$, i.e. Gaussian density integrates to one, we have

$$\int \exp\left(-\frac{(y-f_i)^2}{2g_i} - \frac{(y-f_j)^2}{2g_j}\right)\mathrm{d}y$$

$$= \sqrt{\frac{2\pi g_ig_j}{(g_i+g_j)}}\exp\left(-\frac{\begin{array}{c}(f_j{}^2g_i+f_i{}^2g_j)/(g_i+g_j)\\ -((f_ig_j+f_jg_i)/(g_i+g_j))^2\end{array}}{2g_ig_j/(g_i+g_j)}\right)$$

$$= \sqrt{\frac{2\pi g_ig_j}{(g_i+g_j)}}\exp\left(-\frac{(f_i-f_j)^2}{2(g_i+g_j)}\right) \tag{A2}$$