Statistics and Computing **15**: 31–41, 2005

Hierarchical Gaussian process mixtures for regression

J.Q. SHI*, R. MURRAY-SMITH†,‡ and D.M. TITTERINGTON**

**School of Mathematics and Statistics, University of Newcastle, UK*
j.q.shi@ncl.ac.uk
†*Department of Computing Science, University of Glasgow, Glasgow, UK*
***Department of Statistics, University of Glasgow, Glasgow, UK*
‡*Hamilton Institute, National University of Ireland, Maynooth, Co. Kildare, Ireland*

Received April 2002 and accepted July 2004

As a result of their good performance in practice and their desirable analytical properties, Gaussian process regression models are becoming increasingly of interest in statistics, engineering and other fields. However, two major problems arise when the model is applied to a large data-set with repeated measurements. One stems from the systematic heterogeneity among the different replications, and the other is the requirement to invert a covariance matrix which is involved in the implementation of the model. The dimension of this matrix equals the sample size of the training data-set. In this paper, a Gaussian process mixture model for regression is proposed for dealing with the above two problems, and a hybrid Markov chain Monte Carlo (MCMC) algorithm is used for its implementation. Application to a real data-set is reported.

Keywords: Gaussian process, heterogeneity, hybrid Markov chain Monte Carlo, mixture models, nonparametric curve fitting

1. Introduction

Gaussian processes have been used in many applications. Initially proposed in O'Hagan (1978), Gaussian process priors have recently been used in Bayesian approaches to regression, classification and other areas; see reviews by Williams (1998) and MacKay (1999). However, two major problems arise when the Gaussian process regression model is applied to a large data-set with repeated measurements. Such data can often be regarded as consisting of a number of 'batches' of values, and one source of difficulty results from possible heterogeneity among the different batches. For example, the application we discuss later concerns data collected during standing-up manoeuvres of paraplegia patients. In practice a few hundred data points are collected during each standing-up of a given patient, and the procedure is repeated several times for each of a number of patients. The data from a single standing-up manoeuvre constitutes a 'batch' in this context. Obviously, the mechanism underlying different standings-up is quite similar, but possibly not the same, even for the same patient. This results in heterogeneity among the replications. The other problem is that implementation of the model requires the inversion of a covariance matrix of dimension $N \times N$, where N is the sample size of the training data. This takes time $O(N^3)$. Even though computing speed has rapidly increased and some approximation methods have been proposed (see for example Gibbs and MacKay 1996), implementation is still time-consuming for a large training data-set. Some approaches, such as the Bayesian committee machine (Tresp 2000), have been developed to deal with the second problem.

In this paper, we use a Gaussian process mixture model for regression to deal with both problems. Mixture models represent a flexible approach (see e.g. Titterington, Smith and Makov 1985, McLachlan and Peel 2000) for modelling a large data-set when there might be heterogeneity and a 'pure' model might be inadequate. The idea of a mixture model involving Gaussian processes has been reported before in the literature. For example, Lemm (1999) used mixtures of Gaussian process priors to model data with arbitrary density and applied the model to image analysis. Rasmussen and Ghahramani (2002) used a mixture model of Gaussian process experts for data in a single batch. They assume that each observation in the batch comes from one of a number of Gaussian processes but the identity of that Gaussian process is not observed, and can vary from

observation to observation. The same is true of the method of Tresp (2001).

Our approach is different. We assume that each *batch* of observations comes from one of a set of Gaussian processes, with all observations within a batch coming from the same process. The familiar hierarchical structure of mixture models is then created by the assumption that the identity of the Gaussian process underlying a given batch is missing. A Bayesian approach is used for analyzing the resulting hierarchical model.

Our problem can be thought of as one of curve fitting with high-dimensional input variables. This is a difficult problem, for which neural network models are often used in practice (see e.g. Cheng and Titterington 1994). However, our experience with our dataset is that the Gaussian process regression model gives a better fit than does the neural network model; see Section 4. Certain nonparametric approaches, such as spline smoothing, can also be used for curve fitting, but implementation is very complicated unless the dimensionality of the input variables is very small.

The paper is organized as follows. Section 2 gives a brief review of Gaussian process models for regression. Section 3 proposes the hierarchical mixture model, and gives details of the algorithm, which implements a Bayesian analysis of the problem. Section 4 examines the performance of the model and the algorithm on a numerical example. Some discussion and further development are given in Section 5.

2. Gaussian process priors for regression

We are given training data consisting of N data points $\{y_i, \boldsymbol{x}_i, i = 1, \ldots, N\}$, where, for each i, $\boldsymbol{x}_i$ is a Q-dimensional vector of *inputs* (independent variables), and y_i is the *output* (dependent variable, target). A Gaussian process regression model is defined by

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i, \tag{1}$$

where $\epsilon_i \sim N(0, \sigma_v^2)$ is an error term. Errors on different data points are independent. The function $f(\boldsymbol{x}_i)$ is a nonlinear function of $\boldsymbol{x}_i$. The prior for this function is assumed to correspond to a Gaussian process; i.e., for each i, $f(\boldsymbol{x}_i)$ has a multivariate normal distribution with zero mean, and there exists a covariance function $C(\boldsymbol{x}_i, \boldsymbol{x}_j) := \mathrm{Cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j))$. An example of such a covariance function is

$$\begin{aligned} C(\boldsymbol{x}_i, \boldsymbol{x}_j) &= C(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) \\ &= v_0 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q \left(x_{iq} - x_{jq}\right)^2\right) \\ &\quad + a_0 + a_1 \sum_{q=1}^{Q} x_{iq} x_{jq}, \end{aligned} \tag{2}$$

where $\boldsymbol{\theta} = (w_1, \ldots, w_Q, v_0, a_0, a_1, \sigma_v^2)$ denotes the set of unknown parameters. Therefore, $\boldsymbol{y} = (y_1, \ldots, y_N)'$ has a normal distribution with zero mean and covariance matrix

$$\Psi(\boldsymbol{\theta}) = \boldsymbol{C}(\boldsymbol{\theta}) + \sigma_v^2 \boldsymbol{I}, \tag{3}$$

where $\boldsymbol{I}$ is an identity matrix, $\boldsymbol{C}(\boldsymbol{\theta})$ is an $N \times N$ matrix with elements as given in (2), and $\Psi(\boldsymbol{\theta})$ is an $N \times N$ matrix.

The covariance function (2) is often used in practice. The first term recognises high correlation between the outputs of cases with nearby inputs, while the other terms are a bias term and a linear regression term; see O'Hagan (1978) and Williams and Rasmussen (1996), among others. More discussion about the choice of covariance function can be found in MacKay (1999).

Given a covariance function and a set of training data,

$$\mathcal{D} = \{\boldsymbol{y}, \boldsymbol{X}\} = \{(y_1, \boldsymbol{x}_1), \ldots, (y_N, \boldsymbol{x}_N)\},$$

the log-likelihood is $L(\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{\Psi}(\boldsymbol{\theta})| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Psi}(\boldsymbol{\theta})^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi$, and the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ can be calculated with the help of an iterative optimization method, such as the conjugate gradient method. This requires the evaluation of $\boldsymbol{\Psi}(\boldsymbol{\theta})^{-1}$, which takes time $O(N^3)$. Efficient implementation with particular reference to approximation of the matrix inversion has been well developed; see for example Gibbs (1997) and MacKay (1999). However, the method is still time-consuming for large sets of training data.

The Gaussian process framework also includes a straightforward way of predicting an output based on the relevant test inputs and on the training data. Let $\boldsymbol{x}^*$ be the test inputs and let $f(\boldsymbol{x}^*)$ be the related nonlinear function. The distribution of $f(\boldsymbol{x}^*)$ given $\boldsymbol{x}^*$ and the training data $\mathcal{D}$ is also a Gaussian distribution, with mean and variance given by

$$E(f(\boldsymbol{x}^*) \mid \mathcal{D}) = \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{y}, \tag{4}$$

$$\mathrm{Var}(f(\boldsymbol{x}^*) \mid \mathcal{D}) = C(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(\boldsymbol{x}^*), \tag{5}$$

where $\boldsymbol{\psi}(\boldsymbol{x}^*) = (C(\boldsymbol{x}^*, \boldsymbol{x}_1), \ldots, C(\boldsymbol{x}^*, \boldsymbol{x}_N))^T$ and $\boldsymbol{\Psi}$ is the covariance matrix of $(y_1, \ldots, y_N)$ given in (3). If y^* is the related output, then its predictive distribution is also Gaussian, with mean given by (4) and variance $(\mathrm{Var}(f(\boldsymbol{x}^*) \mid \mathcal{D}) + \sigma_v^2)$.

Clearly these recipes for prediction involve the parameters $\boldsymbol{\theta}$. In non-Bayesian analysis the mean (4), evaluated at the MLE of $\boldsymbol{\theta}$, is generally used as a prediction for y^*.

In our Bayesian approach, prior information about the unknown parameter $\boldsymbol{\theta}$ is summarised in the form of a prior density $p(\boldsymbol{\theta})$. Then the posterior density for $\boldsymbol{\theta}$ given the training data $\mathcal{D}$ is

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\boldsymbol{\theta})p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}), \tag{6}$$

where $p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta})$ is the density function of an N-dimensional multivariate normal distribution with zero mean and covariance matrix $\boldsymbol{\Psi}(\boldsymbol{\theta})$, such as is defined by (3). Since the form of the covariance function is complicated in terms of $\boldsymbol{\theta}$, it is not feasible to carry out analytical inference based on the above posterior distribution. A Markov chain Monte Carlo approach is generally used; see Neal (1997) and MacKay (1999).

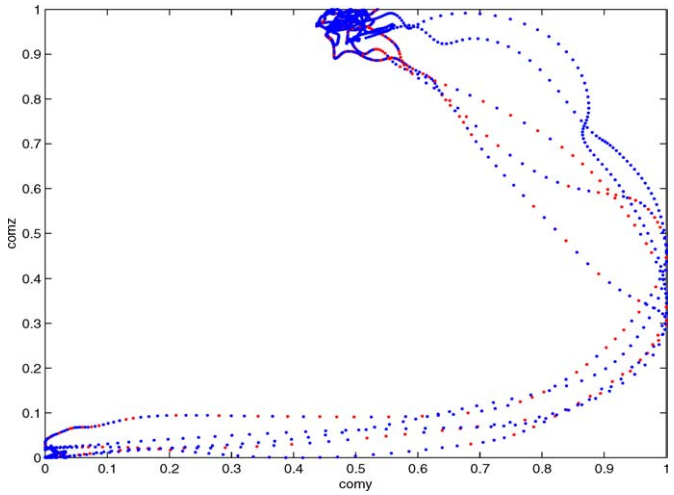

Fig. 1. *Paraplegia data for one patient: trajectory of the body COM for five standings-up for one patient, where* comy *and* comz *represent horizontal and vertical position respectively*

3. Hierarchical mixture models

3.1. *The hierarchical models*

We first use the paraplegia data to illustrate the data structure; the details will be given in Section 4. As mentioned in Section 1, in this example, we study standing-up manoeuvres made by paraplegic patients. The *output* is the pair of horizontal and vertical trajectories of the body centre of mass, and the *input* variables include a range of different measures such as forces and torques under the patient's feet and the arm support. Our main objective is to model and predict the above output using the input variables. In one standing-up, a few hundred (training) data points (involving output and input variables) are recorded. In Fig. 1, each curve along the x-axis represents the output horizontal trajectory ($comy$) for one standing-up manoeuvre, and the y-axis represents the output vertical trajectory ($comz$). Each curve constitutes a set of data points for $comy$ and $comz$, each of which can be modelled by a Gaussian process regression model as discussed in the previous section, and prediction can be based on the posterior mean of the nonlinear function $f(\boldsymbol{x})$ as given in (4).

In fact, Fig. 1 presents 5 batches of data, corresponding to 5 standings-up. Clearly the basic model structure seems to be the same for different batches and yet there is evidence of heterogeneity between different batches. Arguably this heterogeneity could represent just random variability, but incorporation of the possibility of systematic heterogeneity by fitting mixture models appears to be justified by the resulting improvement in fit shown later, for example in Fig. 5.

In general, suppose that there are M different batches of data and that, in the mth batch, N_m observations are collected. The observations are assumed independent for the different batches. Let the observations be $y_{mn},\ m = 1, \ldots, M,\ n = 1, \ldots, N_m$. Similarly to (1), the data in the mth batch can be modelled by

$$y_{mn} = f_m(\boldsymbol{x}_{mn}) + \epsilon_{mn}. \tag{7}$$

Let $\mathcal{D}_m$ be the data, both outputs and inputs, collected in the mth batch. The model for the nonlinear function $f_m(\boldsymbol{x})$ is assumed to correspond to a Gaussian process defined by (2), and this is denoted by

$$f_m(\boldsymbol{x}) \sim GP(\boldsymbol{\theta}_m). \tag{8}$$

If there is no heterogeneity among the different batches, or if we are happy to assume that a pure Gaussian process model is adequate, we can assume that all the $\boldsymbol{\theta}_m$'s are the same. However, in the paraplegia example, the patient may use different techniques in different standings-up, so that we need to accommodate the possibility of heterogeneity for the five batches of data presented in Fig. 1. We also need to analyse data collected from different patients, and then the heterogeneity is likely to be more severe, because of factors such as the different ages, weights, heights and injury levels for the different patients.

A random-effect-type approach is one way of dealing with heterogeneity. For example, we can use a hierarchical approach in which Gaussian process models (7) and (8) are combined with a parametric model,

$$\boldsymbol{\theta}_m \sim g(\cdot), \tag{9}$$

where $g(\cdot)$ is the density function of a known distribution, such as a normal distribution. However, since the dimension of $\boldsymbol{\theta}$ is generally very large and the meaning of $\boldsymbol{\theta}$ is not clear, it is very difficult to justify such a parametric model (in fact, in Section 5.1 we do report the results obtained from an asymptotic approach with a random-effects flavour). Instead, we choose a finite mixture model, in which

$$f_m(\boldsymbol{x}) \sim \sum_{k=1}^{K} \pi_k GP(\boldsymbol{\theta}_k), \tag{10}$$

where K is the number of components in the mixture model, and π_k is the weight corresponding to the kth component. We assume that K has a given fixed value in this paper; discussion about how to choose K will be given in the next two sections. We shall assume that the K component Gaussian process models have the same structure, defined in (2), but with different values of the parameter $\boldsymbol{\theta}_k$. However, the theory and the algorithm developed in the following sections can also be used without substantial difficulty for mixtures of Gaussian processes with different structures.

The model in (10) can be regarded as a hierarchical model, if we introduce an unobservable latent indicator variable z_m. If the value of z_m is given, as k, say, which can take any value from 1 to K, the model for batch m is a Gaussian process regression model $GP(\boldsymbol{\theta}_k)$, i.e.

$$f_m(\boldsymbol{x}) \,|\, (z_m = k) \ \sim \ GP(\boldsymbol{\theta}_k). \tag{11}$$

The higher-level model for the latent indicator variable takes the simple form in which

$$P(z_m = k) = \pi_k, \quad k = 1, \ldots, K, \tag{12}$$

independently for each m.

This hierarchical model offers certain advantages. First, it is easy to extend it to a more general model. For example, the distribution of the latent indicator variable z may depend on some information $\boldsymbol{u}_m$ related to the particular group, such as the age, sex and height of the patient in our paraplegia data, so that an allocation model of the form $z_m \sim F(\boldsymbol{u}_m)$ may be used as a higher-level model in (12), along the lines of Thompson *et al.* (1998). Secondly, the latent indicator variable can be used in implementation; see the discussion in the rest of this section.

3.2. *Bayesian inference for $\boldsymbol{\theta}$*

3.2.1. Priors

Let $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, and let $\mathcal{D}$ be the collection of training data. The posterior density of the unknown parameters is given by

$$p(\boldsymbol{\Theta}, \boldsymbol{\pi} \mid \mathcal{D}) \propto p(\boldsymbol{\Theta}, \boldsymbol{\pi}) p(\mathcal{D} \mid \boldsymbol{\Theta}, \boldsymbol{\pi}), \tag{13}$$

where

$$p(\mathcal{D} \mid \boldsymbol{\Theta}, \boldsymbol{\pi}) = \prod_{m=1}^{M} \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}_m \mid \boldsymbol{\theta}_k, \boldsymbol{X}_m).$$

We assume that, a priori, $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}$ are independent, and the $\boldsymbol{\theta}_k$ are independent and identically distributed, so that

$$p(\boldsymbol{\Theta}, \boldsymbol{\pi}) = p(\boldsymbol{\pi}) \prod_{k=1}^{K} p(\boldsymbol{\theta}_k).$$

We will use the covariance function defined in (2), and adopt the priors given in Rasmussen (1996); see also Neal (1997). Thus, each w_i has an inverse Gamma distribution:

$$w^{-1} \sim Ga\left(\frac{\alpha}{2}, \frac{\alpha}{2\mu}\right).$$

Note that $E(w^{-1}) = \mu$ and that small values of α produce vague priors. The hyperparameter μ is assumed to take the value $\mu_0 Q^{2/\alpha}$, with $\alpha = 1$, $\mu_0 = 1$. The priors on $\log(\sigma_v^2)$, a_0 and a_1 are taken as Gaussian, $N(-3, 3^2)$, corresponding to fairly vague priors, and the prior on $\log(v_0)$ is $N(-1, 1)$ (Rasmussen 1996).

As in the general setting of mixture models, we assume that $(\pi_1, \ldots, \pi_K)$ has a Dirichlet distribution, i.e.

$$p(\pi_1, \ldots, \pi_k) \sim D(\delta, \ldots, \delta),$$

with $\delta = 1$, for example.

Obviously, it is very difficult to do analytical posterior analysis for (13). A hybrid MCMC algorithm is therefore proposed in this paper and the details are given in the next subsection.

3.2.2. The implementation

We use the Gibbs sampler (Geman and Geman 1984) to deal with (13). However, instead of generating a sample of $(\boldsymbol{\Theta}, \boldsymbol{\pi})$ from its posterior density (13) directly, we found that implementation is much simpler if the latent variables $\boldsymbol{z} = (z_1, \ldots, z_M)$ are simulated along with the unknown parameter $\boldsymbol{\Theta}$, as is common in the Bayesian analysis of mixture data. Inference about $\boldsymbol{\pi}$ can be easily obtained through $\boldsymbol{z}$ by model (12). The detailed description of one sweep of this procedure based on the Gibbs sampler is defined as follows:

(a) update $\boldsymbol{z}$ from $p(\boldsymbol{z} \mid \boldsymbol{\Theta}, \mathcal{D})$ given the current value of $\boldsymbol{\Theta}$; and
(b) update $\boldsymbol{\Theta}$ from $p(\boldsymbol{\Theta} \mid \boldsymbol{z}, \mathcal{D})$ given the current value of $\boldsymbol{z}$.

In Step (a), $p(z_1, \ldots, z_M \mid \boldsymbol{y}, \boldsymbol{\Theta})$ still has quite a complicated form. A Gibbs subalgorithm is therefore used in this step; we present the details in the Appendix.

In Step (b), if we assume that, a priori, the $\boldsymbol{\theta}_k$ are independent, for $k = 1, \ldots, K$, then the conditional density function of $\boldsymbol{\Theta}$ is

$$p(\boldsymbol{\Theta} \mid \mathcal{D}, \boldsymbol{z}) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k \mid \mathcal{D}, \boldsymbol{z}),$$

with

$$p(\boldsymbol{\theta}_k \mid \mathcal{D}, \boldsymbol{z}) \propto p(\boldsymbol{\theta}_k) \prod_{m \in \{z_m = k\}} p(\boldsymbol{y}_m \mid \boldsymbol{\theta}_k, \boldsymbol{X}_m). \tag{14}$$

Thus $\boldsymbol{\theta}_k, k = 1, \ldots, K$, are conditionally independent given $(z_1, \ldots, z_M)$, and we can deal with each $\boldsymbol{\theta}_k$ separately. Note that the right-hand side of (14) involves a product of factors of the form $p(\boldsymbol{y}_m \mid \boldsymbol{\theta}_k, \boldsymbol{X}_m)$, which just requires the inversion of a covariance matrix of dimension N_m, and this is generally much smaller than the total sample size of $N = N_1 + \cdots + N_M$. As a consequence, the computational burden is much less than that incurred by modelling the data-set by a single Gaussian process regression model.

However, the dimension of $\boldsymbol{\theta}_k$ is $Q + 4$ for the covariance function defined in (2), where Q may vary from one to a few dozen. Moreover, the above conditional density function may have a complex form, and may be multi-modal. It is still quite challenging to simulate from such a density function. In this paper, we adopt the Hybrid MC method (Duane, Kennedy and Roweth 1987), the details of which are given in the Appendix. The discussion in Rasmussen (1996) and Neal (1997) indicates that this is a good method for sampling from the above conditional distribution.

Therefore, our algorithm consists of a Gibbs subalgorithm in Step (a) and a Hybrid Monte Carlo algorithm in Step (b). The algorithm still converges to the correct stationary distribution provided the chains from the subalgorithms are aperiodic and irreducible; see for example Section 5.4.4 in Carlin and Louis (2000). We shall refer to the algorithm as Hybrid Markov chain Monte Carlo (Hybrid MCMC).

3.3. *Prediction*

Using the algorithm discussed above, we generate T, say, samples of the parameters of interest $\boldsymbol{\Theta}$ and the latent indicator variables $\boldsymbol{z}$ from their joint posterior distribution. Denote the set of samples by $\{\boldsymbol{\theta}_1^{(t)}, \ldots, \boldsymbol{\theta}_K^{(t)}, \boldsymbol{z}^{(t)}, t = 1, \ldots, T\}$. The idea of the Bayesian sampling-based approach is to use this set of samples to do posterior inference, including prediction.

For prediction, we need the posterior density of $f_m(\boldsymbol{x})$ at $\boldsymbol{x}^*$, namely

$$\begin{aligned} p(f_m(\boldsymbol{x}) \mid \mathcal{D}, \boldsymbol{x}^*) &= \int p(f_m(\boldsymbol{x}) \mid \mathcal{D}_m, \boldsymbol{x}^*, \boldsymbol{\theta}, z_m) \\ &\quad \times p(\boldsymbol{\theta}, z_m \mid \mathcal{D}_m)\, d\boldsymbol{\theta}\, dz_m \\ &\simeq \frac{1}{T} \sum_{t=1}^{T} p\big(f_m(\boldsymbol{x}) \mid \mathcal{D}_m, \boldsymbol{x}^*, \boldsymbol{\theta}^{(t)}, z_m^{(t)}\big). \end{aligned} \tag{15}$$

The distribution corresponding to $p(f_m(\boldsymbol{x}) \mid \mathcal{D}_m, \boldsymbol{x}^*, \boldsymbol{\theta}^{(t)}, z_m^{(t)})$ is Gaussian with mean of the form (4) and variance of the form (5). In general, we use the predictive mean of (15) as a prediction for a new set of test inputs in the mth batch, calculated by

$$\hat{y}_m^* = \big(\hat{y}_m^{*(1)} + \cdots + \hat{y}_m^{*(T)}\big)/T, \tag{16}$$

where $\hat{y}_m^{*(t)}$ is given by (4) for the particular value $\boldsymbol{\theta}^{(t)}$. The variance associated with the prediction can be calculated similarly, as

$$\hat{\sigma}_m^{*2} = \frac{1}{T} \sum_{t=1}^{T} \hat{\sigma}_m^{*2(t)} + \frac{1}{T} \sum_{t=1}^{T} \big(\hat{y}_m^{*(t)}\big)^2 - (\hat{y}_m^*)^2, \tag{17}$$

where $\hat{\sigma}_m^{*2(t)}$ is given by (5). The predictive variance is $(\hat{\sigma}_m^{*2} + \hat{\sigma}_v^2)$.

Batches $1, \ldots, M$ provide an empirical distribution of the set of all possible batches. This empirical distribution can be written as

$$\hat{P}(\text{batch is batch } m) = \frac{1}{M}, \tag{18}$$

for $m = 1, \ldots, M$. We can use this for the batch identifier of any new set of data. Therefore, the prediction for the response associated with a test input $\boldsymbol{x}^*$ in a new batch is

$$\hat{y}^* = \sum_{m=1}^{M} \hat{y}_m^*/M \tag{19}$$

and the variance is

$$\hat{\sigma}^{*2} = \sum_{m=1}^{M} \hat{\sigma}_m^{*2}/M + \left(\sum_{m=1}^{M} \hat{y}_m^{*2}/M - \hat{y}^{*2} \right), \tag{20}$$

where $\hat{y}_m^*$ and $\hat{\sigma}_m^{*2}$ are given by (16) and (17) respectively. Note that $\hat{\sigma}^{*2}$ is larger than the average of the variances, $\sum_{m=1}^{M} \hat{\sigma}_m^{*2}/M$. The second item in (20) represents the heterogeneity among the different batches. The predictive variance is $(\hat{\sigma}^{*2} + \hat{\sigma}_v^2)$.

4. Application to the modelling of standing-up manoeuvres

Our application concerns FES-assisted standing-up manoeuvres performed by paraplegic patients. The acronym 'FES' stands for 'Functional Electrical Stimulation': patients stand up with the help of an arm support along with electrical stimulation of their paralyzed lower extremities. The Functional Electrical Stimulation artificially invokes muscle contractions and thus creates torques in the body joints. In the case of standing up, the knee joint extensor muscles, the quadriceps group, are stimulated by two surface electrodes on each leg. In the experiments, the stimulation level was constant and was triggered by the user via push-buttons; for more details see Kamnik, Bajd and Kralj (1999). The stimulation sequences were determined on the basis of known subject body position and arm reactions. Using Goniometers, accelerometers, other sensors and the related algorithms, we can arrange for the body position and other information to be fed back to the simulator control system. However, the equipment is very expensive and it is a tedious job to set the sensors. This method can therefore only be used in the simulation or laboratory environment; it is not suitable for implementation in home or clinical praxis. For this reason, the supportive forces acting at the interaction points with the paraplegic's environment are considered as an alternative feedback source; for more details see Kamnik *et al.* (2003). To use the supportive force feedback information, we need a model that relates the supportive forces to the output trajectory. In this paper, we select as outputs the horizontal (*comy*) and vertical (*comz*) trajectories of the body COM (centre of mass), and select 14 input variables, such as the forces and torques under the patient's feet, under the arm support handle and under the seat while the body is in contact with it. In one standing-up, output and inputs are recorded for a few hundred time steps. The experiment was repeated several times for one patient, and there are total of 8 patients involved in this project. The data are standardized by height and weight of the patient (see the details in Kamnik *et al.* 2003).

First we study the data-set in Fig. 1, which shows the trajectories of the body COM for the five standings-up for a single patient; there are a few hundred data points for each standing up. From the whole data-set, we randomly select about half of the data points from the first three standings-up as training data; the rest are used as test data. The sample sizes of the training data are 101, 76 and 91 respectively for the three batches. We apply the hierarchical mixture model defined by (11) and (12). For each mixture component, we use the same covariance function (2), but with different values of the parameter $\boldsymbol{\theta}_k$.

We assume that the number of components is $K = 2$, and use the hybrid MCMC algorithm to generate samples from the relevant posterior distribution. The algorithm converges very quickly. On the basis of traces of the values of the log-likelihood and other criteria (see e.g. Gelman 1996), the algorithm tends to stabilise after about 1200 iterations (see Fig. 2). In this example, we discard the first 2000 iterations. In order to have

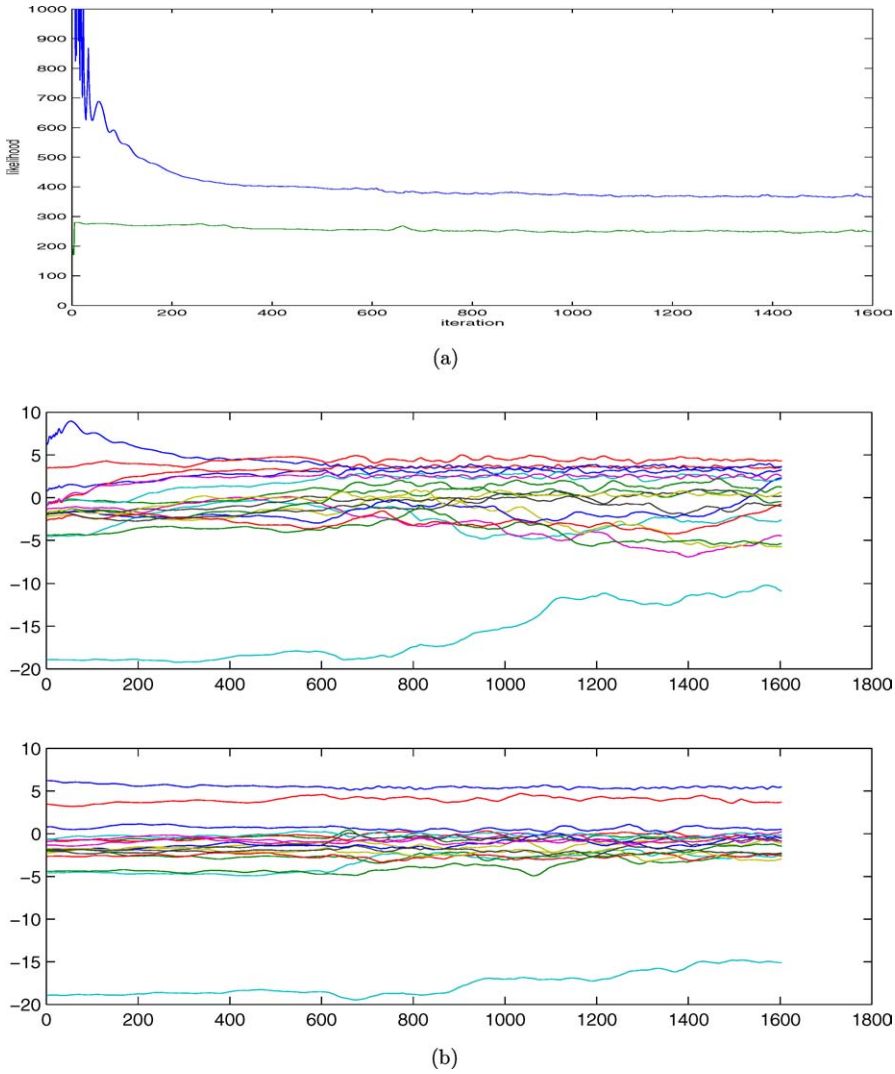

Fig. 2. *Paraplegia data for one patient: (a) The values of log-likelihood for two mixture components. (b) Samples of unknown parameters generated from their posterior distributions for two mixture components*

approximately independent draws, we select one sample from each 20 iterations, and a total of 100 samples are selected altogether. Those 100 samples are approximately independently and identically distributed according to the related posterior distribution. They form the basis of posterior inference, such as the creation of predictions for test data.

To measure the performance of the model and the algorithm, the actual output values of the test data are compared with the predictions. The results are plotted in Fig. 3 and presented in Table 1, where $rmse$ is root mean squared error between the prediction and the true test value, and r is the related correlation coefficient. There are two kinds of test data. One is made up of the other half of the data points in the first three standings-up. We expect that in this case the predictions should be very close to the true data. The numerical results in Table 1 and Fig. 3 confirm this expectation. The other set of test data comes from the last two standings-up. We use the training data from the first three standings-up to simulate those two manoeuvres; this is one of the major objectives of this engineering project. The results are also presented in Table 1 and Fig. 3. The values of *rmse* are 0.0097 and 0.0052, and the sample correlation coefficients are 0.9638 and 0.9963, for *comy* and *comz* respectively. From those summary statistics and from Fig. 3, the fit is seen to be very good. The method has also been compared with neural network models in Kamnik *et al.* (2002). The results obtained from the Gaussian process mixture model are much better than those achieved by the neural network model. For example, the value of *rmse* achieved by the former model for the first three standings-up in Fig. 3 is about half of the value obtained with the latter model; for details see Kamnik *et al.* (2002).

We have discussed how to predict a new standing-up manoeuvre using data from the same patient. A more interesting problem is to simulate a standing-up manoeuvre for a patient different from those who contributed to the training data. To

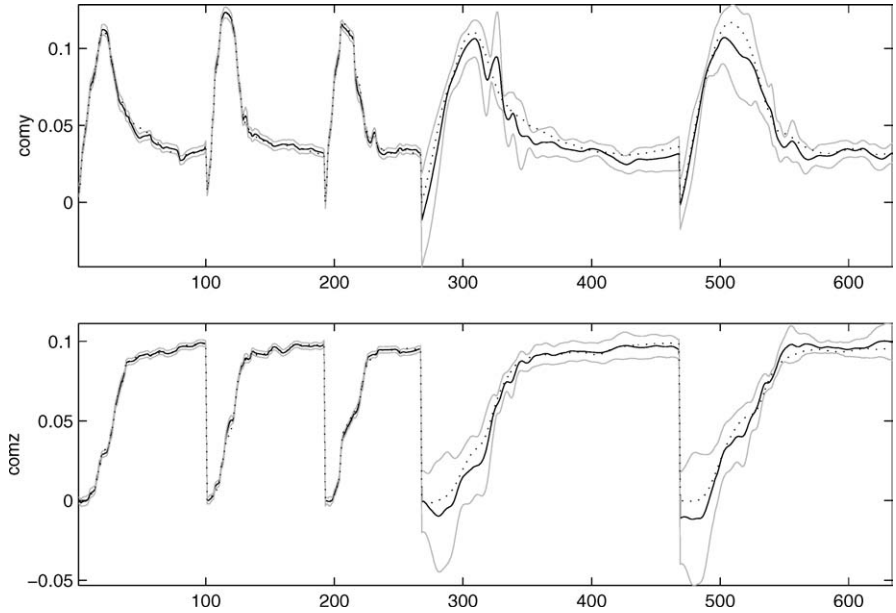

Fig. 3. *Paraplegia data for one patient: The true test data (points), the predictions and the 95% confidence intervals (lines)*

illustrate this, we use a training data-set that includes half the data points from the first three standings-up for five patients. There are therefore a total of 15 'batches' of data. We use a Gaussian process regression mixture model with four mixture components to build a predictor that we apply to a new patient. The final results are presented in Table 1 and Fig. 4. As expected the results are not as good as with prediction based on data from the same patient (see the last two standings-up in Fig. 3). However, if we bear in mind the complexity of the problem and compare the results with those of other approaches, such as neural network models, the overall performance is good. More discussion of this issue will be given in the next section.

We now discuss some problems in the selection of the model and its implementation. The first issue is that of the number of mixture components, which is related to the number of 'clusters' among the different batches. Here we choose this number empirically. Biomechanics research has shown that patients usually use the following three ways of standing up: the static manner, in which they bring their upper body forward prior to rising and then they rise primarily in the vertical direction; the dynamical manner, in which the manoeuvre is fast and consists of two phases, namely forward motion with which they pull their upper body forward and vertical motion when they rise vertically; and in the third way patients stand up primarily with the help of their arm support. (However, information about the type of standing-up underlying our data is not available to us, and in general it is difficult to know in practice which method a patient used.) Bearing in mind the differences among different patients, we use a mixture model with four mixture components when we work on the training data from five patients. Figure 5 shows the results obtained by using the mixture model with $K = 4$ and $K = 1$; the results from the former are much better than those from the latter. We have also tried the model with three components; the final results are almost the same as the results in Table 1 for $K = 4$. For the case when the training data come from the same patient, since the heterogeneity among the different standings-up is not very substantial, we choose the model with two mixture components.

Table 1. rmse *and correlation coefficient (r) between true and predicted responses*

	comy		comz	
Training data: Half of first three standings-up Model: GP regression mixture model with two components				
Test data	rmse	r	rmse	r
First three standings-up	0.0023	0.9967	0.0012	0.9994
Last two standings-up	0.0097	0.9638	0.0052	0.9963
Training data: Half of first three standings-up for 5 patients Model: GP regression mixture model with four components				
	comy		comz	
Test data	rmse	r	rmse	r
Five standings-up for new patient	0.0195	0.4596	0.0291	0.9269

The version of the hybrid MCMC algorithm used in this paper is quite efficient and converges very quickly. For the mixture model, the dimension of the covariance matrix that requires to be inverted is equal to the sample size of each batch, and the CPU time for running one iteration on our SPARC station 20 is about 2 seconds in this example, comparing to about 23 seconds for the conventional method, which treats all the training data as a single 'batch'. The approach is also quite robust. When we choose different values of the hyperparameters in the prior distribution, the final results are almost the same; the sample

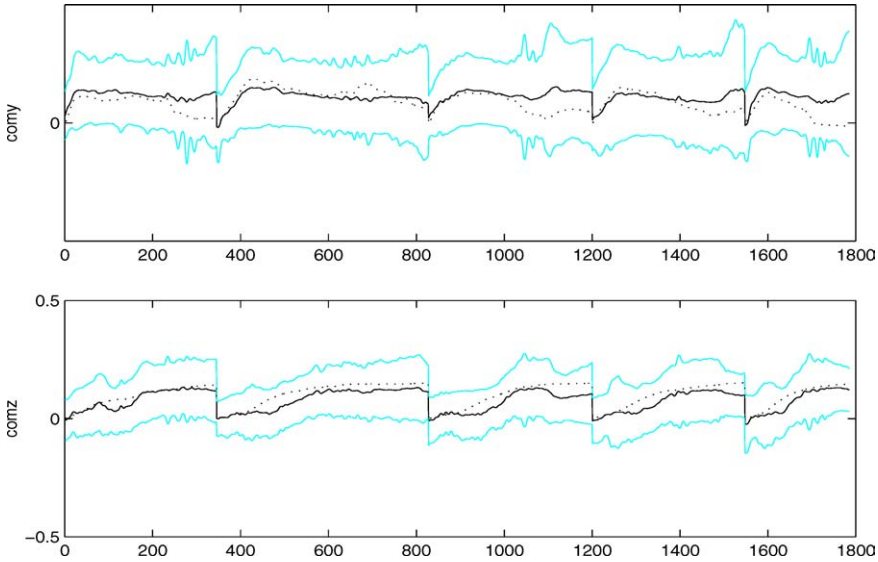

Fig. 4. *Prediction for standing-up manoeuvre for a new patient based on training data from five others: The true test data (points), the predictions and the 95% confidence intervals (lines)*

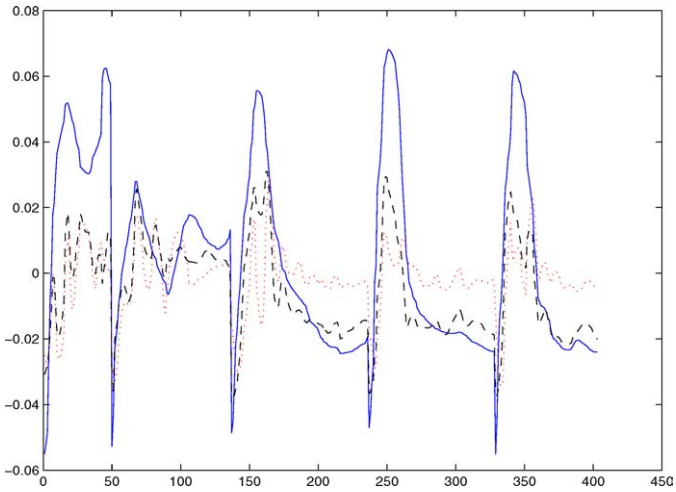

Fig. 5. *Prediction for standing-up manoeuvre for a new patient using the mixture model with $K = 1$ (dotted line, rmse = 0.0270) and $K = 4$ (dashed line, rmse = 0.0199). The solid line represents true values*

size is generally quite large for these engineering problems, so the data dominate the prior.

If the number of input variables is large, the number of unknown parameters is also large. We should choose the starting point carefully to avoid divergence of the algorithm, especially when the number of mixture components and the number of batches are also large. One way of achieving this is to choose the means of the prior distribution as the starting points. For some complicated problems, we may consider the following approach: divide the batches into K 'clusters' using the knowledge and information obtained in collecting data, such as the different ways of standing up; then use a single GP regression model in each cluster separately. The estimates from this single model are used as the starting point of the final mixture model and the starting values of the indicator variables are related to those clusters. Both approaches were used in our example. Both sets of final results were good and were very similar to each other.

5. Discussion

5.1. *Other methods for prediction*

In Section 3.3, we assume that the empirical distribution is (18), and use (19) and (20) to calculate a prediction and its variance for a new set of test inputs $\boldsymbol{x}^*$. An alternative approach is to use the following asymptotic result:

$$\hat{f}_m(\boldsymbol{x}^*) \sim N\big(f_m(\boldsymbol{x}^*),\ \hat{\sigma}_m^{*2}\big), \tag{21}$$

for $m = 1, \ldots, M$, where $\hat{f}_m(\boldsymbol{x}^*)$ is the posterior mean of the nonlinear function $f_m(\boldsymbol{x})$ corresponding to the mth batch, given by (16), and $\hat{\sigma}_m^{*2}$ is given by (17). If we assume that the $f_m(\boldsymbol{x}^*)$ are the same for all M batches, then a weighted least squares calculation estimates the prediction $f(\boldsymbol{x}^*) = f_m(\boldsymbol{x}^*)$ by

$$\frac{\sum_m \hat{f}_m(\boldsymbol{x}^*)/\hat{\sigma}_m^{*2}}{\sum_m 1/\hat{\sigma}_m^{*2}},$$

with variance

$$\frac{1}{\sum_m 1/\hat{\sigma}_m^{*2}}.$$

The idea is quite similar to the Bayesian committee machine (BCM) which divides the whole data-set into M different batches, but does not accommodate heterogeneity (Tresp 2000).

Heterogeneity can be modelled by incorporating the following lower-level model:

$$f_m(\boldsymbol{x}^*) \sim N(f(\boldsymbol{x}^*),\ \tau^2).$$

The estimate of the overall mean is then

$$\frac{\sum_m \hat{f}_m(\boldsymbol{x}^*)/\left(\hat{\sigma}_m^{*2}+\hat{\tau}^2\right)}{\sum_m 1/\left(\hat{\sigma}_m^{*2}+\hat{\tau}^2\right)},$$

and the estimated variance is

$$\frac{1}{\sum_m 1/\left(\hat{\sigma}_m^{*2}+\hat{\tau}^2\right)}, \tag{22}$$

where $\hat{\tau}^2$ is an estimate of τ^2, obtained for example by maximum likelihood. This is in fact a random-effects-type approach and contains features that are similar to aspects of the method given in Section 3.3. The second term in (20) has an effect similar to that of τ^2 here, in that both represent heterogeneity among different groups.

We compared these different approaches by calculating the pointwise 95% Bayesian credible regions for the paraplegia data. The results are shown in Fig. 6. The dashed line represents the results obtained from the above model with heterogeneity. These results seem reasonable. The method discussed in Section 3.3 gave a very similar result, which is not presented here.

The dotted lines in Fig. 6 represent the Bayesian credible regions calculated from the model by the BCM, which are unrealistically narrow for the example; empirical coverage rates are also presented. The model without consideration of heterogeneity, i.e. with $\tau = 0$, gave a very similar result to this.

However, if the data are collected from different sources that show considerable variety, the Bayesian credible regions calculated by the method in Section 3.3 or the above model with heterogeneity may be very wide. This phenomenon is shown in Fig. 3. Though the point predictions were quite good, the credible regions gave little information. Since different patients will have different heights, weights, levels of injury, etc., the variety among them is substantial. Therefore, the variance in (22) will be dominated by τ^2, which is the variance related to heterogeneity. A way of dealing with this problem is to model the indicator variable (12) using further contextual information about individual patients. Research along these lines is in progress.

5.2. *Further developments*

We have assumed that the number of mixture components K is fixed, and we use an ad hoc approach to determine this number. There is much literature concerning the selection of K. For the Bayesian approach discussed in this paper, a possible approach is to maximize, over K, the scoring function $P(\mathcal{D}, K) = p(K)p(\mathcal{D} \mid K)$, where

$$p(\mathcal{D} \mid K) = \int p(\mathcal{D} \mid \boldsymbol{\Theta}_K, K)p(\boldsymbol{\Theta}_K \mid K)\, d\boldsymbol{\Theta}_K,$$

and $p(K)$ is a prior probability that there are K components. Here $\boldsymbol{\Theta}_K$ denotes all unknown parameters including $\{\pi_k\}$, and the dimension of $\boldsymbol{\Theta}_K$ is generally very large, so that this integral is intractable. It is therefore of interest to find an approximation to the above integral or an alternative approach to model selection. Ideally we would wish to tackle the problems of assessing the value of K and parameter estimation simultaneously using methods such as those in Richardson and Green (1997) and Stephens (2000). Research along these lines is currently in progress.

In our application, the output trajectory and the input supportive forces are all functions of time. Functional data analysis (Ramsay and Silverman 1997) is an ideal alternative approach

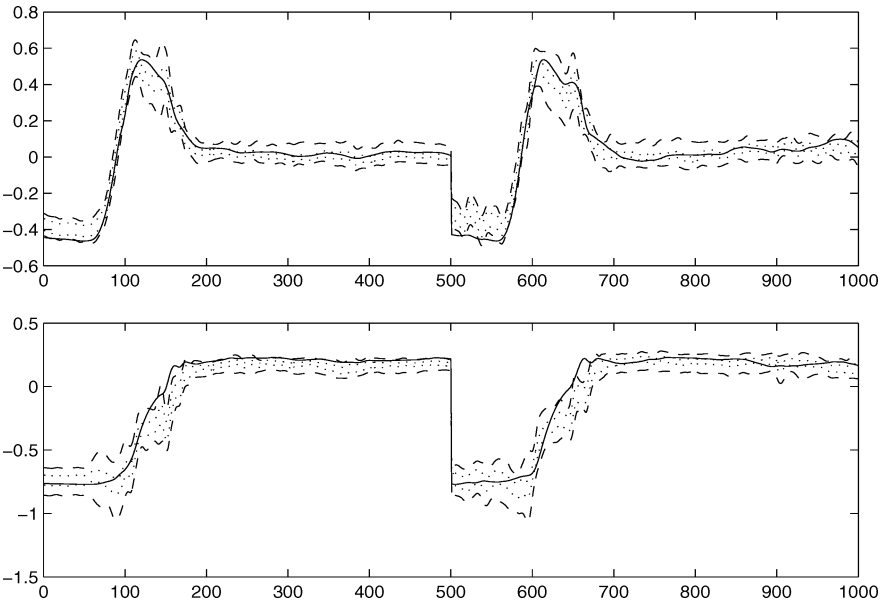

Fig. 6. *Predictions with 95% Bayesian credible regions: Solid line—the true value; dashed lines—regions calculated from the model with heterogeneity, giving coverage rates of 0.966 and 0.902 for each manoeuvre in the upper panel and 0.818 and 0.936 for the lower; dotted lines—regions calculated from BCM, for which the coverage rates are 0.408, 0.460, 0.188 and 0.484 respectively*

for modelling such relationships. However, implementation is very difficult, even for the functional linear model, when the output response and the input covariates are all treated as functions. It therefore requires further research to develop some efficient algorithms and to study functional nonlinear models.

Appendix: Hybrid MCMC algorithm

The details of the subalgorithms for the Hybrid MCMC algorithm discussed in Section 3.2.2 are as follows.

Step (a) Sampling from $p(z_1, \ldots, z_m \mid \boldsymbol{y}, \boldsymbol{\Theta})$

Let c_k be the number of observations for which $z_m = k$, over all $m = 1, \ldots, M$. Then

$$p(z_1, \ldots, z_M \mid \pi_1, \ldots, \pi_K) = \prod_{k=1}^{K} \pi_k^{c_k},$$

and

$$\begin{aligned} p(z_1, \ldots, z_M) &= \int p(z_1, \ldots, z_M \mid \pi_1, \ldots, \pi_K) \\ &\quad \times p(\pi_1, \ldots, \pi_K) d\pi_1 \ldots d\pi_K \\ &= \frac{\Gamma(K\delta)}{\Gamma(M + K\delta)} \prod_{k=1}^{K} \frac{\Gamma(c_k + \delta)}{\Gamma(\delta)}. \end{aligned}$$

The conditional density function of z_m is

$$p(z_m = k \mid \boldsymbol{z}_{-m}) = \frac{c_{-m,k} + \delta}{M - 1 + K\delta},$$

where the subscript $-m$ indicates all indices except m and $c_{-m,k}$ is the number of observations for which $z_i = k$ for all $i \neq m$. A Gibbs subalgorithm is used to update z_m by sampling from the following density:

$$\begin{aligned} p(z_m = k \mid \boldsymbol{z}_{-m}, \boldsymbol{y}, \boldsymbol{\Theta}) &\propto p(z_m = k \mid \boldsymbol{z}_{-m}) p(\boldsymbol{y} \mid \boldsymbol{\Theta}, \boldsymbol{z}) \\ &\propto p(z_m = k \mid \boldsymbol{z}_{-m}) p(\boldsymbol{y}_m \mid \boldsymbol{\theta}_k). \end{aligned}$$

We used the fact that $p(\boldsymbol{y}_m \mid \boldsymbol{\theta}, z_m)$ is the density function of the Gaussian distribution with zero mean and covariance matrix $\boldsymbol{\Psi}(\boldsymbol{\theta}_k)$ if $z_m = k$.

An alternative approach is to treat $(\pi_1, \ldots, \pi_K)$ as missing variables as well. One sweep of the procedure for sampling $\boldsymbol{z}$ and $\boldsymbol{\pi}$ is as follows:

(i) sample z_m from $p(z_m = k \mid \boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\pi}) \propto \pi_k p(\boldsymbol{y}_m \mid \boldsymbol{\theta}_k)$;
(ii) sample $(\pi_1, \ldots, \pi_K)$ from $p(\pi_1, \ldots, \pi_K) \sim D(\delta + c_1, \ldots, \delta + c_K)$.

In this approach, a sample of $\boldsymbol{\pi}$ is also generated.

Step (b) Sampling from $p(\boldsymbol{\theta}_k \mid \mathcal{D}, \boldsymbol{z})$ in (14).

We write $p(\boldsymbol{\theta}_k \mid \mathcal{D}, \boldsymbol{z}) \propto \exp(-\mathcal{E})$, where $\mathcal{E}$ is called potential energy. If we assume that, a priori, the $\boldsymbol{\theta}_k$ are independent for $k = 1, \ldots, K$, then the conditional density function of $\boldsymbol{\Theta}$ is

$$p(\boldsymbol{\Theta} \mid \mathcal{D}, \boldsymbol{z}) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k \mid \mathcal{D}, \boldsymbol{z})$$

with

$$p(\boldsymbol{\theta}_k \mid \mathcal{D}, \boldsymbol{z}) \propto p(\boldsymbol{\theta}_k) \prod_{m \in \{z_m = k\}} p(\boldsymbol{y}_m \mid \boldsymbol{\theta}_k).$$

Thus $\boldsymbol{\theta}_k, k = 1, \ldots, K$, are conditionally independent given $(z_1, \ldots, z_M)$, and we can deal with each $\boldsymbol{\theta}_k$ separately. (For simplicity we omit the subscript k from $\boldsymbol{\theta}_k$ in the rest of this Appendix.) The idea of the Hybrid MC method (Duane, Kennedy and Roweth 1987) is to create a fictitious dynamical system where the parameter vector $\boldsymbol{\theta}$ of interest, called the position variables, is augmented by a set of latent variables $\boldsymbol{\phi}$, called the momentum variables, with the same dimension as that of $\boldsymbol{\theta}$. The kinetic energy is a defined as a function of the associated momenta: $\mathcal{K}(\boldsymbol{\phi}) = \frac{1}{2} \sum \phi_i / \lambda$. The momentum variables are therefore independent and Gaussian with zero mean and variance λ. The total energy $\mathcal{H}$ of the system is the sum of the kinetic energy $\mathcal{K}$ and the potential energy $\mathcal{E}$. The Hybrid MC samples are drawn from the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{\phi} \mid \mathcal{D}, \boldsymbol{z}) \propto \exp(-\mathcal{H}) = \exp(-\mathcal{E} - \mathcal{K})$.

One sweep of a variation of the Hybrid MC Algorithm (Horowitz 1991, see also Neal 1993, Rasmussen 1996) is as follows.

(i) Starting from the current state $(\boldsymbol{\theta}, \boldsymbol{\phi})$, calculate the new state $(\boldsymbol{\theta}(\epsilon), \boldsymbol{\phi}(\epsilon))$ by the following 'Leapfrog' steps with step size ϵ:

$$\begin{aligned} \phi_i\left(\frac{\epsilon}{2}\right) &= \phi_i - \frac{\epsilon}{2} \frac{\partial \mathcal{E}}{\partial \theta_i}(\boldsymbol{\theta}), \\ \theta_i(\epsilon) &= \theta_i + \epsilon \phi_i\left(\frac{\epsilon}{2}\right) / \lambda, \\ \phi_i(\epsilon) &= \phi_i\left(\frac{\epsilon}{2}\right) - \frac{\epsilon}{2} \frac{\partial \mathcal{E}}{\partial \theta_i}(\boldsymbol{\theta}(\epsilon)), \end{aligned}$$

where $\partial \mathcal{E}(\boldsymbol{\theta}) / \partial \theta_i$ is the first derivative of $\mathcal{E}$ evaluated at $\boldsymbol{\theta}$.

(ii) The new state $(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*)$ is such that

$$(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*) = \begin{cases} (\boldsymbol{\theta}(\epsilon), \boldsymbol{\phi}(\epsilon)) & \text{with probability} \\ & \min(1, p(\boldsymbol{\theta}, \boldsymbol{\phi}) / p(\boldsymbol{\theta}(\epsilon), \boldsymbol{\phi}(\epsilon))) \\ (\boldsymbol{\theta}, -\boldsymbol{\phi}) & \text{otherwise,} \end{cases}$$

where $p(\boldsymbol{\theta}, \boldsymbol{\phi}) / p(\boldsymbol{\theta}(\epsilon), \boldsymbol{\phi}(\epsilon)) = \exp[\mathcal{H}(\boldsymbol{\theta}(\epsilon), \boldsymbol{\phi}(\epsilon)) - \mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi})]$.

(iii) Generate v_i from the standard Gaussian distribution, and update ϕ_i to $\alpha \phi_i^* + \sqrt{1 - \alpha^2} v_i$.

Rasmussen (1996) suggests setting $\epsilon = 0.5 N_m^{-1/2}$, $\lambda = 1$ and $\alpha = 0.95$.

Acknowledgments

The authors would like to gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council for grant GR/M76379/01, *Modern Statistical Approaches to Off-equilibrium Modelling for Nonlinear System Control*. RM-S is grateful for support from Science Foundation Ireland grant

00/PI.1/C067. We would also like to thank Dr. R. Kamnik and Prof. T. Bajd of the Laboratory of Biomedical Engineering of the University of Ljubljana for allowing us to use their experimental data. The authors are also grateful to the reviewers for many helpful comments.

References

Carlin B.P. and Louis T.A. 2000. Bayes and Empirical Bayes Methods for Data Analysis, 2nd edition. Chapman & Hall/CRC, London.

Cheng B. and Titterington D.M. 1994. Neural networks: A review from a statistical perspective (with discussion). Statistical Science 9: 2–54.

Duane S., Kennedy A.D., and Roweth D. 1987. Hybrid Monte Carlo. Physics Letters B 195: 216–222.

Gelman A. 1996. Inference and monitoring convergence. In: Gilks W.R., Richardson S., and Spiegelhalter D.J. (Eds.), Markov Chain Monte Carlo in Practice, Chapman Hall, London, pp. 131–144.

Geman S. and Geman D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6: 721–741.

Gibbs M.N. 1997. Bayesian Gaussian Processes for Regression and Classification. PhD thesis, Cambridge University. (Available from http://wol.ra.phy.cam.ac.uk/mng10/GP/)

Gibbs M.N. and MacKay D.J.C. 1996. Efficient implementation of Gaussian processes for interpolation. Technical report. Cambridge University. (Available from http://wol.ra.phy.cam.ac.uk/mackay/GP/)

Horowitz A.M. 1991. A generalized guided Monte Carlo algorithm. Physics Letters B 268: 247–252.

Kamnik R., Bajd T., and Kralj A. 1999. Functional electrical stimulation and arm supported sit-to-stand transfer after paraplegia: A study of kinetic parameters. Artificial Organs 23: 413–417.

Kamnik R., Shi J.Q., Murray-Smith R., and Bajd T. 2003. Feedback information in FES supported standing-up in paraplegia. Technical Report. University of Glasgow. (Available from http://www.staff.ncl.ac.uk/j.q.shi/ps/roman.pdf).

Lemm J.C. 1999. Mixtures of Gaussian process priors. In: Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN99), IEE Conference Publication No. 470. Institution of Electrical Engineers, London.

MacKay D.J.C. 1999. Introduction to Gaussian processes. Technical report. Cambridge University. (Available from http://wol.ra.phy.cam.ac.uk /mackay/GP/)

McLachlan G.J. and Peel D. 2000. Finite Mixture Distributions. Wiley, New York.

Neal R.M. 1997. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical report 9702. Dept of Computing Science, University of Toronto. (Available from http://www.cs.toronto.edu/~radford/)

O'Hagan A. 1978. On curve fitting and optimal design for regression (with discussion). Journal of the Royal Statistical Society B 40: 1–42.

Ramsay J.O. and Silverman B.W. 1997. Functional Data Analysis. Springer, New York.

Rasmussen C.E. 1996. Evaluation of Gaussian Processes and Other Methods for Non-linear Regression. PhD Thesis. University of Toronto. (Available from http: //bayes.imm.dtu.dk)

Rasmussen C.E. and Ghahramani Z. 2002. Infinite mixtures of Gaussian process experts. In: Dietterich T., Becker S., and Ghahramani Z. (Eds.), Advances in Neural Information Processing Systems 14, MIT Press.

Richardson S. and Green P.J. 1997. On Bayesian analysis of mixtures with an unknown number of components (with discussion). Journal of the Royal Statistical Society B 59: 731–758.

Stephens M. 2000. Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods. The Annals of Statistics 28: 40–74.

Thompson T.J., Smith P.J., and Boyle J.P. 1998. Finite mixture models with concomitant information: Assessing diagnostic criteria for diabetes. Applied Statistics 47: 393–404.

Titterington D.M., Smith A.F.M., and Makov U.E. 1985. Statistical Analysis of Finite Mixture Distribution. Wiley, Chichester, New York.

Tresp V. 2000. The Bayesian committee machine. Neural Computation 12: 2719–2741.

Tresp V. 2001. Mixtures of Gaussian processes. In: Leen T.K., Diettrich T.G., and Tresp V. (Eds.), Advances in Neural Information Processing Systems, 13, MIT Press.

Williams C.K.I. 1998. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In: Jordan M.I. (Ed.), Learning and Inference in Graphical Models, Kluwer, pp. 599–621.

Williams C.K.I. and Rasmussen C.E. 1996. Gaussian process for regression. In: Touretzky D.S. et al. (Eds.), Advances in Neural Information Processing Systems 8, MIT Press.