Hierarchical Gaussian Process Mixtures for Regression

Shi, J. Q.[a&b 1] , Murray-Smith, R.[a] and Titterington, D. M.[b]

[a]Department of Computing Science and [b]Department of Statistics

University of Glasgow, Glasgow, UK.

April 9, 2002

Summary

As a result of their good performance in practice and their desirable analytical properties, Gaussian process regression models are becoming increasingly of interest in engineering and other fields. However, there are two major problems when the model is applied to a large data-set with repeated measurements. One is the 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 mixture regression model of Gaussian processes is proposed, and a hybrid Markov chain Monte Carlo (MCMC) algorithm is used for the implementation. If we use this model and algorithm, the computational burden decreases dramatically. A real application is used to illustrate the mixture model and its implementation.

Keywords: Gaussian Process; Heterogeneity; Hybrid Markov chain Monte Carlo; Mixture models; Nonlinear regression.

[1] *Address for correspondence:* Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK. Email: shi@dcs.gla.ac.uk

1 Introduction

The theory of Gaussian processes has already been established. As a result of their good performance in practice and their desirable analytical properties, Gaussian processes have wide applications; for example, Wiener processes (a special case of a Gaussian process) are used as a basic model in Brownian motion. Initially proposed in O'Hagan (1978), Gaussian process priors have recently been used in regression, classification and other areas (see reviews by Williams, 1998, and MacKay, 1999). However, there are two major problems when the Gaussian process regression model is applied to a large data-set with repeated measurements. One is the heterogeneity among the different replications (or groups). For example, in the paraplegia data-set we discuss in this paper, a few hundred data points are collected in each standing-up of a paraplegia patient, the procedure being repeated several times for each of eight patients. Obviously, the mechanism for every standing-up is quite similar, but not the same, even for the same patient. This results in heterogeneity among the replications. The other problem is that the implementation of the model requires the inversion of a covariance matrix, of which the dimension is $N \times N$, 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 e.g. Gibbs and MacKay, 1996), implementation is still time-consuming for a large training data-set.

For the data-set with repeated measurements discussed above, we can define a model with the following hierarchical structure: a lower-level model is applied separately to each group to model the basic structure of the data; then the set of lower-level models have similar structures but with some mutual heterogeneity, and a higher-level model is used among groups to model the heterogeneity. In this paper, we fit a separate Gaussian process regression model to the data corresponding to each group. Since the number of unknown parameters involved in the Gaussian process regression model is generally quite large, it is quite difficult to define a parametric higher-level model. A mixture model represents a good semi-parametric approach (see e.g. Titterington, Smith and Makov, 1985) for modelling a large data-set with the above hierarchical structure, and we therefore propose a hierarchical mixture model of Gaussian processes for regression in this paper.

We use the Bayesian approach in this paper to analyze the above hierarchical structure. However, the posterior density function of the unknown parameters involves an multi-dimensional integral, and it is natural to consider using a Gibbs sampler (Geman and Geman, 1984) algorithm. We treat the indicator variable of the mixture model as a latent variable. In each iteration, we consider the conditional distribution of unknown parameters given the values of the latent indicator variables; then we consider the conditional distribution of the latent variables given the value of those unknown parameters. Conditional on the latent indicator variables, we can treat the data-set for each group separately, and it therefore just requires the inversion of a covariance matrix corresponding to the size of a sub-sample of the training data for each group. As a consequence, the computational burden decreases dramatically.

The problem of curve fitting with high dimensional input variables is difficult. Neural network models are often used in practice (see e.g. Cheng and Titterington, 1994). However, our experience is that the Gaussian process regression model generally gives a better fit than the neural network model; see Section 4. Some nonparametric approaches, such as spline smoothing, can also be used for curve fitting. However, the implementation is very complicated if the dimension of the input variables is large.

The idea of a mixture model with Gaussian processes has been used on several different problems in the literature. For example, Lemm (1999) used mixtures of Gaussian process priors to model data with arbitrary density and applied the model in image analysis.

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 steps of the algorithm. 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 N data points of training data $\{y_n, \boldsymbol{x}_n, n = 1, \cdots, N\}$, where $\boldsymbol{x}$ is a Q-dimensional vector of *inputs* (independent variables), and y is the *output* (dependent variable, target). A Gaussian process is defined in such a way that $y(\boldsymbol{x})$ has a Gaussian prior distribution with zero mean and covariance function $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = Cov(Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j))$. An example of such a covariance function is

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

where $\boldsymbol{\theta} = (w_1, \cdots, w_Q, v_0, a_0, a_1, \sigma_v^2)$, and $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. This covariance function is often used in practice. The first term recognises high correlation between the outputs of cases with nearby inputs, while the rest are a bias term, a linear regression term and a noise term respectively; 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 $\boldsymbol{y} = (y_1, \cdots, y_N)^T$, the log-likelihood is

$$L(\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{\Psi}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Psi}^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi, \tag{2}$$

where $\boldsymbol{\Psi} = \boldsymbol{\Psi}(\boldsymbol{\theta})$ is the covariance matrix of $\boldsymbol{y}$ with dimension $N \times N$. The maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ can be calculated by maximizing the above log-likelihood. An iterative optimization method, such as the conjugate gradient method, can be applied. It requires the evaluation of $\boldsymbol{\Psi}^{-1}$, 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, it still becomes time-consuming for large sets of training data.

If prior information is to be incorporated, a Bayesian approach is generally used. Let $p(\boldsymbol{\theta})$ be the prior density function of $\boldsymbol{\theta}$ and let $\mathcal{D} = \{\boldsymbol{y}, \boldsymbol{x}\}$ be the training data. Then the posterior density of $\boldsymbol{\theta}$ given the training data is

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}), \tag{3}$$

where $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ 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 (1). Since the form of the covariance function is complicated in terms of $\boldsymbol{\theta}$, it is infeasible to do any analytical inference based on the above posterior distribution. A Markov chain Monte Carlo approach is generally used; see Neal (1997) and MacKay (1999).

One major goal in engineering and other fields is to predict an output based on the training data. This problem can be solved thanks to the attractive analytical properties of Gaussian processes. Let $\boldsymbol{x}^*$ be the test inputs and y^* be the output. The predictive distribution is the conditional distribution of y^* given $\boldsymbol{x}^*$ and training data $\mathcal{D}$, which is also a Gaussian distribution with mean and variance given by

$$\hat{y}^* = \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{y}, \tag{4}$$

$$\hat{\sigma}^{*2} = C(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(\boldsymbol{x}^*); \tag{5}$$

where $\boldsymbol{\Psi}$ is the covariance matrix of $(y_1, \cdots, y_N)$, $\boldsymbol{\psi}(\boldsymbol{x}^*) = (C(\boldsymbol{x}^*, \boldsymbol{x}_1), \cdots, C(\boldsymbol{x}^*, \boldsymbol{x}_N))^T$. The mean (4), evaluated at the MLE of $\boldsymbol{\theta}$, is generally used as a prediction of y^*.

3 Hierarchical mixture models

3.1 The hierarchical models

In many areas of empirical modelling we are faced with repeated experiments on similar objects and processes. However, those data may come from different sources. For the paraplegia data discussed in this paper, the data come from 8 patients, of whom the ages range from 17 to 57 years, the weights range from 58 to 95kg, and the heights range from 159 to 185cm. Therefore, a simple model, such as the single Gaussian process regression model discussed in last section, may not fit the data well. A mixture model is a natural choice for modelling a large data-set collected from different sources.

A mixture model can be defined to fit the data collected in such experiments. Suppose that there are M different groups of data (replications). In the mth group, N_m observations are collected. Let the observations be $y_{mn},\ m = 1, \cdots, M,\ n = 1, \cdots, N_m$. A mixture model of Gaussian process regression can be defined as follows:

$$\boldsymbol{y}_m = (y_{m1}, \cdots, y_{mN_m}) \sim \sum_{k=1}^{K} \pi_k GP_k(\boldsymbol{\theta}_k) \tag{6}$$

independently for $m = 1, \cdots, M$, where $GP_k(\boldsymbol{\theta}_k)$ stands for the density function of a Gaussian process regression model $GP_k(\boldsymbol{\theta}_k)$, as defined in the last section. A special case corresponds to $GP_k(\boldsymbol{\theta}_k) = GP(\boldsymbol{\theta}_k)$, i.e. the different $GP_k(\boldsymbol{\theta})$ have exactly the same structure, but with different values of the parameter $\boldsymbol{\theta}_k$. K is the number of components of the mixture model. We assume that K has a fixed given value in this paper.

The above model is equivalent to the model with the following hierarchical structure: a lower-level model is assumed for the data corresponding to each replication (i.e. within a group) separately, and the structures of those models are similar but with some mutual heterogeneity; a higher-level model is defined to model the heterogeneity among different replications (groups). In this paper, a hierarchical mixture model of Gaussian processes regression has the following structure:

$$\boldsymbol{y}_m | z_m = k \ \sim \ GP_k(\boldsymbol{\theta}_k), \tag{7}$$

where z_m is an unobservable latent indicator variable. The model for group m is a Gaussian process regression model $GP_k(\boldsymbol{\theta}_k)$ if $z_m = k$ is given. The association among the different groups is introduced by the latent variable z_m, for which

$$P(z_m = k) = \pi_k, \quad k = 1, \cdots, K, \tag{8}$$

for each m. There are several advantages of using this hierarchical model. First, it is easy to extend it to some more general model. For example, the distribution of the latent indicator variable z may depend on some information related to the particular group such as the age, sex and height of the patient in our paraplegia data. Therefore, an allocation model $z_m \sim F(\boldsymbol{u}_m)$ may be used as a higher-level model in (8). Research along this line is currently in progress. Secondly, the latent indicator variable can be used in implementation; see the discussion in the rest of this section.

3.2 Bayesian inference and priors

We adopt the Bayesian approach in this paper. For convenience of presentation, we assume that $GP_k(\boldsymbol{\theta}_k) = GP(\boldsymbol{\theta}_k)$ for each k; they have the same covariance function, such as (1), but with different parameter vectors $\boldsymbol{\theta}_k$. It will be seen that the approach discussed in this section can be extended to more general models without any substantial difficulty.

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

$$p(\boldsymbol{\Theta}, \boldsymbol{\pi}|\mathcal{D}) \propto p(\boldsymbol{\Theta}, \boldsymbol{\pi}) p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{\pi}), \tag{9}$$

where

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

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 (1), and adopt the priors given in Rasmussen (1996); see also Neal (1997). Each w_i has an inverse Gamma distribution:

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

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

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

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

with $\delta = 1$, for example.

Obviously, it is very difficult to do analytical posterior analysis for (9). A hybrid MCMC algorithm is therefore proposed in this paper. The details are given in the next subsection.

3.3 The implementation

From (9), the density of $(\boldsymbol{\Theta}, \boldsymbol{\pi}) = \{\boldsymbol{\theta}_k, \pi_k, k = 1, \cdots, K\}$ has a very complicated form. It is very difficult to do inference based on this posterior density directly. We therefore use the Gibbs sampler (Geman and Geman, 1984). Instead of generating a sample of $(\boldsymbol{\Theta}, \boldsymbol{\pi})$ from its posterior density (9), we found from our study that the implementation is much more simple and efficient if the latent variable $\boldsymbol{z} = (z_1, \cdots, z_M)$ is augmented with the unknown parameter $\boldsymbol{\Theta}$. Inference about $\boldsymbol{\pi}$ can be easily obtained through $\boldsymbol{z}$ by model (8). The detailed description of one sweep of this procedure based on the Gibbs sampler is defined as follows:

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

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

In step (a), $\boldsymbol{Z} = (z_1, \cdots, z_m)$ and $p(z_1, \cdots, z_m|\boldsymbol{y}, \boldsymbol{\Theta})$ has a still quite 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 $k = 1, \cdots, K$, then the conditional density function of $\boldsymbol{\Theta}$ is

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

with

$$p(\boldsymbol{\theta}_k|\mathcal{D}, \boldsymbol{z}) \propto p(\boldsymbol{\theta}_k) \prod_{m \in \{z_m = k\}} p(\boldsymbol{y}_m|\boldsymbol{\theta}_k). \tag{10}$$

Thus $\boldsymbol{\theta}_k, k = 1, \cdots, K$, are conditionally independent given $(z_1, \cdots, z_M)$, and we can deal with each $\boldsymbol{\theta}_k$ separately. Note that the right-hand side of (10) involves a product of $p(\boldsymbol{y}_m|\boldsymbol{\theta}_k)$, which just requires the inversion of a covariance matrix of dimension N_m, which is generally much less than the total sample size $N = N_1 + \cdots + N_M$. 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 (1), 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 a challenging topic in statistics to simulate from such a density function. In this paper, we adopt the Hybrid MC method (Duane, Kennedy and Roweth, 1987). The discussion in Rasmussen (1996) and Neal (1997) indicates that this is a good method for sampling from the above conditional distribution. The details will be given in Appendix.

Therefore, the algorithm used for the hierarchical mixtures of Gaussian process regression includes two steps; a Gibbs subalgorithm is used in Step (a) and a Hybrid Monte Carlo algorithm is used in Step (b). This algorithm still converges to the correct stationary distribution provided the chains from the subalgorithms are aperiodic and irreducible; see for example §5.4.4 in Carlin and Louis (2000). This algorithm is referred to as Hybrid Markov chain Monte Carlo (Hybrid MCMC).

Using the algorithm discussed above, we generate samples of the parameters of interest $\mathbf{\Theta}$ and the latent indicator variables $\boldsymbol{z}$ from their posterior distribution. Denote the set of samples by $\{\boldsymbol{\theta}_1^{(t)}, \cdots, \boldsymbol{\theta}_K^{(t)}, \boldsymbol{z}^{(t)}, t = 1, \cdots, T\}$. The idea of the Bayesian sampling-based approach is to use this set of samples to do posterior inference. We use this approach to do prediction, which is a major objective in system control.

Suppose $\boldsymbol{x}^*$ is a test input, known to come from the mth group, so that the predictive density of the corresponding output is approximated by

$$\begin{aligned} p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*) &= \int p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}, z_m)p(\boldsymbol{\theta}, z_m|\mathcal{D})d\boldsymbol{\theta}dz_m \\ &\simeq \frac{1}{T}\sum_{t=1}^{T} p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}^{(t)}, z_m^{(t)}). \end{aligned} \tag{11}$$

The predictive distribution $p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}^{(t)}, z_m^{(t)})$ is Gaussian with mean (4) and variance (5). In general, we use the predictive mean as a prediction, calculated by

$$\hat{y}_m^* = (\hat{y}_m^{*(1)} + \cdots + \hat{y}_m^{*(T)})/T, \tag{12}$$

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:

$$\hat{\sigma}_m^{*2} = \frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}_m^{*2(t)} + \frac{1}{T}\sum_{t=1}^{T}(\hat{y}_m^{*(t)})^2 - (\hat{y}_m^*)^2, \tag{13}$$

where $\hat{\sigma}_m^{*2(t)}$ is given by (5).

If there is no information about the particular group to which the test input $\boldsymbol{x}^*$ belongs, we may suppose that this test point is in the mth group with probability M^{-1} for all $m = 1, \cdots, M$. Therefore, the prediction is

$$\hat{y}^* = \sum_{m=1}^{M}\hat{y}_m^*/M \tag{14}$$

and the variance is

$$\hat{\sigma}^{*2} = \sum_{m=1}^{M}\hat{\sigma}_m^{*2}/M + \sum_{m=1}^{M}\hat{y}_m^{*2}/M - \hat{y}^{*2}, \tag{15}$$

where $\hat{y}_m^*$ and $\hat{\sigma}_m^{*2}$ are given by (12) and (13) respectively. Note that $\hat{\sigma}^{*2}$ is larger than the average of the variances, $\sum_{m=1}^{M}\hat{\sigma}_m^{*2}/M$.

4 Application to the modelling of standing-up manoeuvres

We analyzed data related to FES-assisted standing-up manoeuvres 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 lower paralyzed extremities. The Functional Electrical Stimulation artificially invokes muscle contractions and thus obtains 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 was constant and triggered by the user via push-buttons; for more details see Kamnik, Bajd and Kralj (1999). The supportive forces are considered as a potential feedback source. To use the supportive force feedback information, we need a model relating the supportive forces and output trajectory. In this paper, as our illustrative example, we select as output 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, Shi, Murray-Smith and Bajd, 2002).

First we study a data-set of 5 standings-up for one patient. A few hundred data points are recorded for each standing-up. The trajectories of the body COM for the five standings-up are presented in Figure 1, which shows that the basic model structure for the five standings-up should be the same, while there is heterogeneity among different standings-up. Thus, the hierarchical mixture model of Gaussian processes discussed in the last section seems a good choice of model for this data-set. 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 groups. We apply the hierarchical mixture model defined by (7) and (8). For each mixture component, we use the same covariance function (1), but with different values of the parameter $\boldsymbol{\theta}_k$.

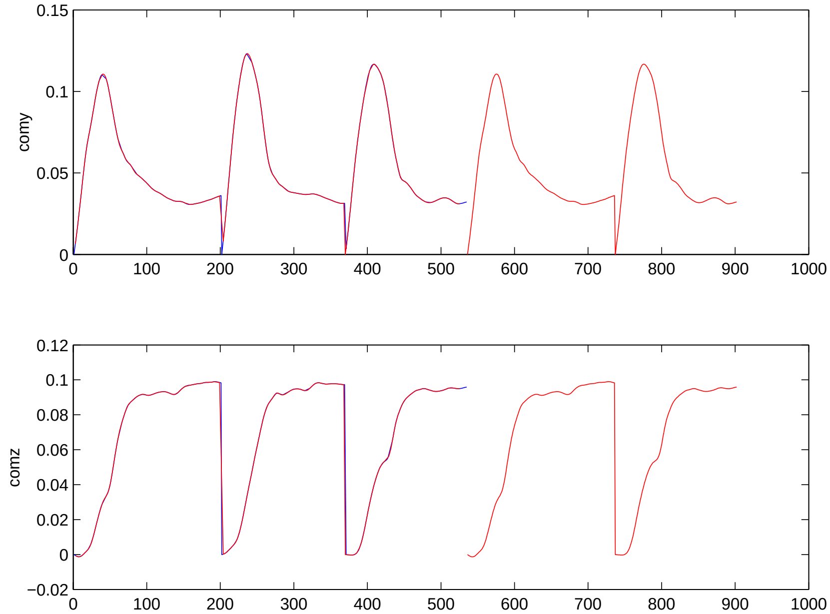

Figure 1. Paraplegia Data for one patient: trajectory of the body COM for five standings-up for one patient, where *comy* and *comz* represent horizontal and vertical position respectively.

We assume that the number of components is two, and use the hybrid MCMC algorithm to generate samples from the relevant posterior distribution. The algorithm converges very quickly. On the basis of the values of the log-likelihood and other criteria (see, e.g. Gelman, 1996), the algorithm tends to stabilise after about 1000 iterations. After the algorithm has converged, we select one sample from each 20 iterations, and a total of 100 samples are selected. Those 100 samples are approximately independently and identically distributed according to the related posterior distribution. They are used to do posterior inference, such as predicting 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 Figure 2 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 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 Figure 2 confirm this expectation. This result is important in practice, since it helps us to determine how many data points should be recorded in an experiment. The other batch 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 Figure 2. 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 Figure 2, the fit is very good. The method is also compared with neural network models. The results obtained from the Gaussian process mixture model are much better than those achieved by the neural network model, for example, the vlaue of *rmse* by the former model for the first three standings-up in Figure 2 is about half of the value by the latter model; see the details in Kamnik *et al* (2002).

Table 1. *rmse* and correlation coefficient (r)

Training data: half of first three standings-up				
Model: GP regression mixture model with two components				
	comy		comz	
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

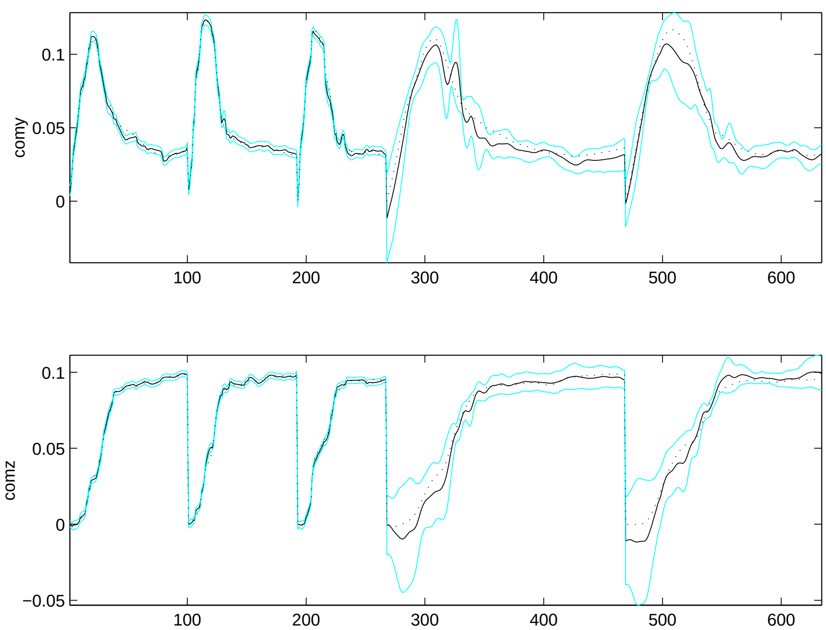

Figure 2. Paraplegia Data for one patient: the true test data (points), the predictions and the 95% confidence intervals (lines).

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 new patient using data from others. To illustrate this, we use a training data-set that includes half the data points for the first three standings-up for five patients. There are a total of 15 groups. 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 Figure 3. Though the results are not as good as the prediction by the data from the same patient (see the last two standings-up in Figure 2), as expected, if we bear in mind the complexity of the problem and compare the results with those of other approaches, the overall performance is quite good; see the detailed discussion in Kamnik *et al* (2002).

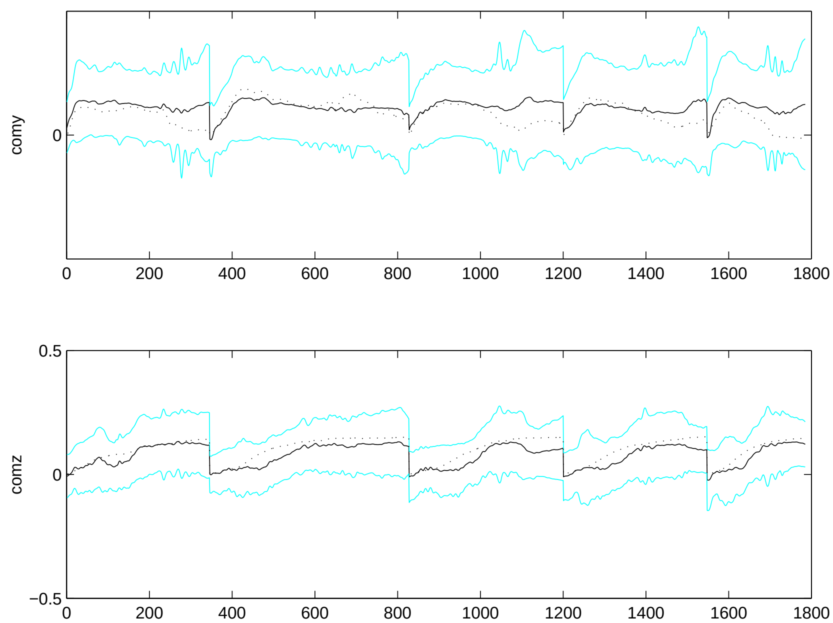

Figure 3. Prediction for standing-up manoeuvre for a new patient based on training data from five others: the true test data (points), the predictions and the 95% confidence intervals (lines).

We now discuss some problems in the selection of the model and its implementation. The first issue is the number of mixture components, which is related to the number of 'clusters' among the different groups. We use an empirical method to choose this number. Biomechanics research has shown that patients usually use the following three ways of standing up: the static manner, meaning that they bring their upper body forward prior to rising and then they rise primarily in the vertical direction; the dynamical manner, meaning that 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. 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. We have also tried the model with three components; the final results are almost the same as the results in Table 1. 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.

The version of the hybrid MCMC algorithm used in this paper is quite efficient and converges very quickly. For the mixture model, since the dimension of the covariance matrix that requires to be inverted is the sample size of each group, the CPU time for running one iteration on our SPARC station 20 is just a few seconds in this example. 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 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 the 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 groups are also large. One way is to choose the means of the prior distribution as the starting points. For some complicated problems, we may consider the following approach. We divide the whole groups into several 'clusters' by the knowledge and information obtained in collecting data such as the different ways of standing up . We 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 the final results were good and were very similar.

5 Discussion

In this paper we propose a hierarchical mixture regression model of Gaussian processes (7) and (8) for a large data-set with repeated measurements. The model has the following two important features. First, the heterogeneity among groups is modelled by a mixture model, and the approach is very flexible because few assumptions are required. Secondly, the observations $\boldsymbol{y}_m = \{y_{m1}, \cdots, y_{mN_m}\}$ in every group are independent for different m, given z_m. The dependence among the different groups is introduced by the latent variable z_m. The vector $\boldsymbol{y}_m$ is a N_m-dimensional Gaussian process. Thus, all the inference is based on dimension N_m, instead of dimension equal to the size of the total sample $N = N_1+\cdots+N_M$. The computational burden for the mixture model decreases dramatically compared with the conventional Gaussian process regression model.

We have assumed that the number of mixture components K is fixed, and we use an empirical 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 the scoring function $P(\mathcal{D}, K) = p(K)p(\mathcal{D}|K)$, where

$$p(\mathcal{D}|K) = \int p(\mathcal{D}|\mathbf{\Theta}_K, K)p(\mathbf{\Theta}_K|K)d\mathbf{\Theta}_K.$$

Since in general the dimension of $\mathbf{\Theta}_K$ is large, 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 for modelling this complex relationship. However, implementation is very difficult even for the functional linear model when output response and the input covariates are all treated as functions. It therefore requires further research to develop some efficient algorithms and study the functional nonlinear models.

Acknowledgements

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.* 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.

Appendix: Hybrid MCMC algorithm

The details of the subalgorithms for the Hybrid MCMC algorithm discussed in Section 3.3 are as follows.

Step (a) Sampling from $p(z_1, \cdots, z_m|\boldsymbol{y}, \mathbf{\Theta})$

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

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

and

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

The conditional density function of z_m is

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

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:

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

We used the fact that $p(\boldsymbol{y}_m | \boldsymbol{\theta}, z_m)$ 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 $(\pi_1, \cdots, \pi_K)$ 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 $p(z_m = k | \boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\pi}) \propto \pi_k p(\boldsymbol{y}_m | \boldsymbol{\theta}_k)$;

(ii) sample $(\pi_1, \cdots, \pi_K)$ from $p(\pi_1, \cdots, \pi_K) \sim D(\delta + c_1, \cdots, \delta + c_K)$.

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

Step (b) Sampling from $p(\boldsymbol{\theta} | \mathcal{D}, \boldsymbol{z}) \propto p(\boldsymbol{\theta}_k)$ in (10).

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

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

with

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

Thus $\boldsymbol{\theta}_k, k = 1, \cdots, K$, are conditionally independent given $(z_1, \cdots, z_M)$, and we can deal with each $\boldsymbol{\theta}_k$ separately. The dimension of $\boldsymbol{\theta}_k$ is $Q + 4$ for the covariance function defined in (1), 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. The discussion in Rasmussen (1996) and Neal (1997) indicates that the Hybrid MC method (Duane, Kennedy and Roweth, 1987) is a good method for sampling from the above conditional distribution. 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 $p(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathcal{D}, \boldsymbol{z}) \propto \exp(-\mathcal{H}) = \exp(-\mathcal{E} - \mathcal{K})$. The discussion in Rasmussen (1996) and Neal (1997) indicates that the Hybrid MC method is a good method for sampling from a conditional density function with complex form, possibly multi-modal and with large-dimensional $\boldsymbol{\theta}$.

One sweep of a variation of the Hybrid MC Algorithm (Horowitz (1991), see also Neal, 1993 and 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:

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

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 $\epsilon = 0.5N_m^{-1/2}$, $\lambda = 1$ and $\alpha = 0.95$.

References

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

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 W.R. Gilks, S. Richardson and D. J. Spiegelhalter (Eds), *Markov Chain Monte Carlo in Practice*, 131-144. London: Chapman Hall.

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. (2002). Feedback information in FES supported standing-up in paraplegia. Technical Report. University of Glasgow.

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

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

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)

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. 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.

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

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

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