# Curve Prediction and Clustering with Mixtures of Gaussian Process Functional Regression Models

Shi, J. Q. *and Wang, B.
School of Mathematics & Statistics, University of Newcastle, UK

December 24, 2006

**Abstract**

Shi *et al.* (2006) proposed a Gaussian process functional regression (GPFR) model to model functional response curves with a set of functional covariates. Two main problems are addressed by this method: modelling nonlinear and nonparametric regression relationship and modelling covariance structure and mean structure simultaneously. The method gives very good results for curve fitting and prediction but side-steps the problem of heterogeneity. In this paper we present a new method for modelling functional data with 'spatially' indexed data, i.e., the heterogeneity is dependent on factors such as region and individual patient's information. For data collected from different sources, we assume that the data corresponding to each curve (or batch) follows a Gaussian process functional regression model as a lower-level model, and introduce an allocation model

*Address for correspondence:* School of Mathematics and Statistics, University of Newcastle, NE1 7RU, UK. Email: j.q.shi@ncl.ac.uk

for the latent indicator variables as a higher-level model. This higher-level model is dependent on the information related to each batch. This method takes advantage of both GPFR and mixture models and therefore improves the accuracy of predictions. The mixture model has also been used for curve clustering, but focusing on the problem of clustering functional relationships between response curve and covariates. The model is examined on simulated data and real data.

*Key words:* Curve clustering, Curve prediction, Functional data analysis, Gaussian process, Gaussian process functional regression model, allocation model, batch data

# 1 Introduction

Shi *et al.* (2006) proposed a Gaussian process functional regression (GPFR) model for modelling functional response curves in terms of a set of functional covariates. Their methods have two main contributions: modelling nonlinear and nonparametric regression relationship and modelling covariance structure and mean structure simultaneously. In Figure 1(a), the mean curve is shown as the solid line. The dotted line is the curve with mean plus independent random errors, while the dashed line is the curve with dependent errors. In practice, many data-sets are similar to the dashed line, having common mean structures but with dependent errors, for example the curves in dark colour as shown in Figure 1(b). Certainly, if we model the mean structure only, we can only model the solid line in Figure 1(a), which is systematically different from the dashed line – the real curve. It is therefore better to model both the mean and covariance structure for such data. If the curve depends on time or a one-dimensional

covariate $t$, Rice and Silverman (1991) defined a stochastic process model. Suppose that there are $M$ curves, with the $m$-th curve defined as

$$y_m(t) = \mu_m(t) + \tau_m(t), \quad \text{for } m = 1, \ldots, M, \tag{1}$$

where $\mu_m(t) = E(y_m(t))$ and $\tau_m(t)$ is a stochastic process with zero mean and kernel covariance function $C(t, t') = Cov(y(t), y(t'))$. Shi *et al.* (2006) extended the idea to deal with a functional regression problem involving a number of functional covariates $\boldsymbol{x} = (x_1, \ldots, x_Q)'$ (the dimension $Q$ of $\boldsymbol{x}$ is usually quite large):

$$y_m(t, \boldsymbol{x}) = \mu_m(t) + \tau_m(\boldsymbol{x}), \tag{2}$$

where $t$ is time or some other one-dimensional temporal or spatial covariate. In many applications, the response curve $y_m$ depends on $t$ as well as on other covariates $\boldsymbol{x}$. In model (2), the mean structure is modelled by a linear functional regression model and the covariance structure is modelled by a Gaussian process regression model:

$$\mu_m(t) = \boldsymbol{u}_m{}'\beta(t), \quad \text{and} \quad \tau_m(\boldsymbol{x}) \sim GP(\boldsymbol{x}; \boldsymbol{\theta}), \tag{3}$$

where the mean structure model depends on $t$ and some non-functional $p$-dimensional covariate $\boldsymbol{u}_m = (u_{m1}, \ldots, u_{mp})'$. The regression relationship between functional response $y$ and the functional covariates $\boldsymbol{x}$ is mainly modelled by the covariance structure through a nonparametric Gaussian process regression model.

The main purpose of this paper is to extend the GPFR model to functional clustering curves or 'spatially' indexed data, which may be collected from different sources. The heterogeneity is dependent on factors such as region and subject. In our paraplegia project (see the details in Section 3), data are collected from each standing-up from different patients (we will refer to all the data collected in each standing-up as

a batch). The data in each batch include both the response curve (the trajectory of the patient's body centre of mass, which can only be observed in a lab environment) and a set of functional covariates (which are easy to observe in any environment). We therefore want to use those functional covariates to reconstruct the response curve. There are several challenging problems for this project. One is the nonlinear and non-parametric relationship between the response curves and the functional covariates; for this the GPFR model seems a good choice (see Shi *et al.*, 2006). Another problem is the heterogeneity of the regression relationships among different batches, depending on the subject's personal circumstances such as the level of injury, height, weight and even gender, depending on the tactic they used for standing up and on other factors. Thus, it is essential to cluster the batches and give larger weight to the batches which have similar situations to the new patient whose response curve we want to reconstruct (i.e. if we can cluster the data, we use the data in the same or similar clusters for prediction). We propose a mixture of GPFR models to address this problem.

Mixture models have been studied for many decades in their two main roles of modelling heterogeneity for data coming from different populations and as a convenient form of flexible population density (see e.g. Titterington, Smith and Makov, 1985 and McLachlan and Peel, 2000). Recently, a mixture of Gaussian process regressions model has been used to fit correlated data in a heterogeneous population (Shi, Murray-Smith and Titterington, 2005). For spatially indexed data, the heterogeneity may depend on region or subject. An allocation model may be defined to address such a problem. For example, Green and Richardson (2000) used a Potts model, and Fernandez and Green (2002) used the density of a Markov random field to define an allocation model. As discussed above, the heterogeneity in the paraplegia data is dependent on the subject,

the tactic used for standing up and other factors. We will therefore define an allocation model which depends on those factors.

The mixture model has also been used for curve clustering. Curve clustering has been widely studied and various methods have been proposed; see for example, Müller (2005), James and Sugar (2003). However, most existing methods concern the clustering of longitudinal data or time-varying functional data, in which the clustering is essentially based on the shapes of the curves. In this paper, we deal with data depending on a number of functional covariates, and what we are interested in is how the observations are affected by the functional covariates for different groups or subjects; that is, we want to cluster relationships between the observations and the input covariates. Gaffney and Smyth (2003) discussed a similar problem but assuming a functional linear relationship between response curve and covariates. Here, we assume a nonlinear and nonparametric GPFR model.

The paper is organized as follows. Section 2.1 defines a mixture model of Gaussian process functional regressions with an allocation model. In Section 2.2 an EM algorithm is discussed for estimating all the unknown parameters. Sections 2.3 and 2.4 discuss the two main problems concerned in this paper, i.e., curve prediction and clustering. The problem of model selection is also discussed in Section 2.4. A number of examples with simulated and real data are presented in Section 3. Finally, we conclude the paper by some discussions in Section 4.

# 2 Methodology

## 2.1 Hierarchical mixture of GPFR models with an allocation model

Suppose that there are $M$ different groups of data which come from different sources. A hierarchical mixture model of GPFRs can be defined for such data by the following hierarchical structure: a lower-level model is assumed for the data corresponding to each batch separately, the structures of those models being similar but with some mutual heterogeneity; a higher-level model is defined for modelling the heterogeneity among different batches. The lower-level model is defined as

$$y_m(t, \boldsymbol{x})|_{z_m=k} \sim GPFR_k(t, \boldsymbol{x}; \boldsymbol{\Theta}_k) \tag{4}$$

independently for $m = 1, \cdots, M$, where $z_m$ is an unobservable latent indicator variable, and $GPFR(t, \boldsymbol{x}; \boldsymbol{\Theta})$ stands for a GPFR model defined in (2) and (3), where $\boldsymbol{\Theta}$ is the set of all the unknown parameters. Thus, $GPFR_k(t, \boldsymbol{x}; \boldsymbol{\Theta}_k)$ is the $k$-th component, with unknown parameters $\boldsymbol{\Theta}_k$.

The association among the different groups is introduced by the latent variable $z_m$, for which

$$P(z_m = k) = \pi_k, \quad k = 1, \cdots, K,$$

for each $m$. However, the heterogeneity among those different batches often depends 'spatially' on the information about each batch as discussed in the previous section. Suppose that the covariates $\boldsymbol{v}_m$ in the $m$-th batch influences the heterogeneity, or that $\boldsymbol{v}_m$ can determine to which 'cluster' the batch will belong. We define a logistic

allocation model as follows:

$$P(z_m = k) = \pi_{mk} = \frac{\exp\{\boldsymbol{v}_m{}'\boldsymbol{\gamma}_k\}}{1 + \sum_{j=1}^{K-1} \exp\{\boldsymbol{v}_m{}'\boldsymbol{\gamma}_j\}}, \quad k = 1, \ldots, K-1, \tag{5}$$

and $P(z_m = K) = \pi_{mK} = 1 - \sum_{j=1}^{K-1} \pi_{mj}$, where $\{\boldsymbol{\gamma}_k, k = 1, \ldots, K-1\}$ are unknown parameters to be estimated.

The model defined in (4) and (5) provides a convenient form of flexible relationship of correlation and accommodates for multivariate cases. The logistic allocation model can be easily replaced by other models such as the Potts model (Green and Richardson, 2000). We may also use a nonparametric Gaussian process classification model (see Shi *et al.*, 2003).

## 2.2 Estimation

From (3), the $k$-th component of the mixture model $GPFR_k$ defined in (4) includes two parts

$$\mu_{mk}(t) = \boldsymbol{u}_m{}'\beta_k(t), \quad \text{and} \quad \tau_{mk}(\boldsymbol{x}) \sim GP_k(\boldsymbol{x}; \boldsymbol{\theta}_k). \tag{6}$$

The covariance structure is modelled by a Gaussian process regression model (see, e.g. Shi *et al.* (2005)) with zero mean and a kernel covariance function. In this paper, we will use the following covariance function (see MacKay (1999) for the selection of the covariance function):

$$C(\boldsymbol{x}_{mi}, \boldsymbol{x}_{mj}; \boldsymbol{\theta}_k) = v_1^k \exp\Big( -\frac{1}{2} \sum_{q=1}^{Q} w_q^k (x_{miq} - x_{mjq})^2 \Big) + a_1^k \sum_{q=1}^{Q} x_{miq} x_{mjq} + v_0^k \delta_{ij}, \tag{7}$$

where $\boldsymbol{\theta}_k \triangleq (w_1^k, \ldots, w_Q^k, v_1^k, a_1^k, v_0^k)$ denotes the set of unknown parameters. The last term of the above equation corresponds to the independent random error.

It is not straightforward to estimate the mean functions from the observations. In this paper we propose to use a B-spline approximation. Let $\boldsymbol{\Phi}(t) = (\boldsymbol{\Phi}_1(t), \ldots, \boldsymbol{\Phi}_D(t))'$

be a set of B-spline basis functions. Then the mean function can be represented by

$$\mu_{mk}(t) = \boldsymbol{u}_m{}'\beta_k(t) = \boldsymbol{u}_m{}'\boldsymbol{B}_k{}'\boldsymbol{\Phi}(t),$$

where $\boldsymbol{B}_k{}' = (B_k^1, \ldots, B_k^D)$ is a $p \times D$ unknown B-spline coefficient matrix. In practice, the specification of $D$ and locations of the knots for B-spline is important. However, we have not found this difficult in our examples, since our data in each batch are quite dense and the results are rather insensitive to the specification; a relatively small number of equally-spaced knots are enough. In the examples discussed later we used 20 equally spaced knots for the splines. More details on selection of the form and number of the basis functions are discussed by Faraway (1997, 2001).

Suppose that $N_m$ observations are obtained for the $m$-th batch/group. The data collected in the $m$-th batch are

$$\mathcal{D}_m = \{(y_{mi}, t_{mi}, \boldsymbol{x}_{mi}) \text{ for } i = 1, \ldots, N_m;\ \boldsymbol{u}_m \text{ and } \boldsymbol{v}_m\}, \tag{8}$$

where $\boldsymbol{u}_m = (u_{m1}, \ldots, u_{mp})'$ and $\boldsymbol{v}_m = (v_{m1}, \ldots, v_{mr})'$ are non-functional covariates which are used in the mean structure model (3) and the logistic allocation model (5) respectively. The totality of observed data are denoted by $\mathcal{D}$. Let $\boldsymbol{y}_m$ be the vector of $\{y_{mi}, i = 1, \ldots, N_m\}$ (we can similarly define $\boldsymbol{t}_m = \{t_{mi}, i = 1, \ldots, N_m\}$, $\boldsymbol{x}_m = \{\boldsymbol{x}_{mi}, i = 1, \ldots, N_m\}$ and so on). From (4), the model for the data is therefore

$$\boldsymbol{y}_m|_{z_m=k} = \boldsymbol{\mu}_{mk} + \boldsymbol{\tau}_{mk}, \quad \text{for } \ k = 1, \ldots, K, \tag{9}$$

where

$$\boldsymbol{\mu}_{mk} = \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m, \quad \text{and } \ \boldsymbol{\tau}_{mk} \ \sim \ N(\boldsymbol{0}, \boldsymbol{C}(\boldsymbol{\theta}_k)),$$

$\boldsymbol{\Phi}_m$ is an $N_m \times D$ matrix with $(i, d)$-th element $\Phi_d(t_{mi})$, and $\boldsymbol{C}(\boldsymbol{\theta}_k)$ is an $N_m \times N_m$ covariance matrix with $(i, j)$-th element $C(\boldsymbol{x}_{mi}, \boldsymbol{x}_{mj}; \boldsymbol{\theta}_k)$ given for example by (7). Thus, the

unknown parameters include $\boldsymbol{B} = \{\boldsymbol{B}_k,\ k = 1, \ldots, K\}$, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k,\ k = 1, \ldots, K\}$ and $\boldsymbol{\gamma} = \{\boldsymbol{\gamma}_k,\ k = 1, \ldots, K-1\}$, involved in the above mean structure, covariance structure and the allocation model (5) respectively. The log-likelihood of $\boldsymbol{\Theta} = (\boldsymbol{B}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ is

$$L(\boldsymbol{B}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{m=1}^{M} \log\{\sum_{k=1}^{K} \pi_{mk} p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m)\}, \tag{10}$$

where $p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m)$ is the density function of the $N_m$-dimensional normal distribution defined in (9).

As a result of the large number of unknown parameters and the complicated covariance structure, it is quite tricky to carry on the estimation. We will use the EM algorithm in this paper. The basic idea is to treat $\boldsymbol{z}$ as missing. For convenience, we define a new variable $z_{mk}$, which takes the value 1 if $z_m = k$ and is 0 otherwise. It is obvious that $\{z_{mk}\}$ and $z_m$ are identical, and we will use $\boldsymbol{z}$ to represent either of them. The log-likelihood for complete data $(\boldsymbol{y}, \boldsymbol{z})$ is

$$L_c(\boldsymbol{\Theta}) = \sum_{k=1}^{K} \sum_{m=1}^{M} z_{mk} \{\log \pi_{mk} + \log p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m)\}. \tag{11}$$

In E-step of the $i$-th iteration, we need to calculate the conditional expectation of $L_c(\boldsymbol{\Theta})$, given $\mathcal{D}$ and based on the current estimate $\boldsymbol{\Theta}^{(i)}$,

$$\begin{aligned} Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)}) &\triangleq E_{\Theta^{(i)}}\{L_c(\boldsymbol{\Theta}) | \mathcal{D}\} \\ &= \sum_{k=1}^{K} \sum_{m=1}^{M} E_{\Theta^{(i)}}(z_{mk} | \mathcal{D}) \{\log \pi_{mk} + \log p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m)\} \\ &= \sum_{k=1}^{K} \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) \{\log \pi_{mk} + \log p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m)\}, \end{aligned}$$

with

$$\alpha_{mk}(\boldsymbol{\Theta}) = E_{\boldsymbol{\Theta}}(z_{mk} | \mathcal{D}) = \frac{\pi_{mk}\ p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m)}{\sum_{j=1}^{K} \pi_{mj}\ p(\boldsymbol{y}_m | \boldsymbol{B}_j, \boldsymbol{\theta}_j, \boldsymbol{x}_m)}. \tag{12}$$

The M-step includes the following two separate maximizations:

- Estimate $\boldsymbol{\gamma}_k$ by maximizing $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)})$;

- Estimate $\boldsymbol{B}_k$ and $\boldsymbol{\theta}_k$ by maximizing $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)})$.

The details of the M-step are given in Appendix. We can also calculate the related standard errors, which are also given in Appendix.

## 2.3 Prediction

We consider two types of prediction. First suppose that we have already observed some training data in a new batch, the $(M+1)$-th batch say, and want to predict the output for a set of new inputs. In addition to the training data observed in the first $M$ batches, we assume that $N$ observations are obtained in the new $(M+1)$-th batch and are denoted by $\{y_{M+1,i}, i = 1, \ldots, N\}$. They are observed at $\{t_1, \ldots, t_N\}$. The corresponding input vectors are $\{\boldsymbol{x}_{M+1,1}, \ldots, \boldsymbol{x}_{M+1,N}\}$, and we also know the batch-based covariates $\boldsymbol{v}_{M+1}$ and $\boldsymbol{u}_{M+1}$. We still use $\mathcal{D}$ to denote all the training data observed in the $M+1$ batches. It is of interest to predict $y^*$ at a new test data point $t^*$ in the $(M+1)$-th batch. Let $\boldsymbol{x}^* = \boldsymbol{x}(t^*)$ be the test inputs. If the component from which this new batch comes is known, the $k$-th say, the corresponding predictive distribution of $y^*$, given training data $\mathcal{D}$, is also a Gaussian distribution. Its predictive mean and variance are given by (see Shi *et al.*, 2006)

$$\begin{aligned} \hat{y}_k^* = E(y^*|\mathcal{D}, z_{M+1} = k) &= \hat{\mu}_{M+1,k}(t^*) + H_k'(\boldsymbol{y}_{M+1} - \hat{\mu}_{M+1,k}(\boldsymbol{t})), \\ \hat{\sigma}_k^{*2} = Var(y^*|\mathcal{D}, z_{M+1} = k) &= [\boldsymbol{C}_k(\boldsymbol{x}^*, \boldsymbol{x}^*) - H_k'\boldsymbol{C}_k H_k](1 + \boldsymbol{u}_{M+1}'(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{u}_{M+1}), \end{aligned}$$

where $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M)'$ is an $M \times p$ matrix, $\boldsymbol{t} = (t_1, \ldots, t_N)$, $\hat{\mu}_{M+1,k}(\cdot) = \boldsymbol{u}_{M+1}'\hat{\boldsymbol{B}}_k'\boldsymbol{\Phi}(\cdot)$, $H_k' = [\boldsymbol{C}_k(\boldsymbol{x}^*, \boldsymbol{x}_{M+1})]'\boldsymbol{C}_k^{-1}$, $\boldsymbol{C}_k(\boldsymbol{x}^*, \boldsymbol{x}_{M+1})$ is the $N$-dimensional vector of the covariance

between $y^*$ and $\boldsymbol{y}_{M+1} = (y_{M+1,1}, \ldots, y_{M+1,N})'$ given $z_{M+1} = k$, i.e.

$$\boldsymbol{C}_k(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) = (C(\boldsymbol{x}^*, \boldsymbol{x}_{M+1,1}; \boldsymbol{\theta}_k), \ldots, C(\boldsymbol{x}^*, \boldsymbol{x}_{M+1,N}; \boldsymbol{\theta}_k))',$$

and $\boldsymbol{C}_k$ is the $N \times N$ covariance matrix of $\boldsymbol{y}_{M+1}$. The covariance is calculated by (7) with the parameters evaluated at $\hat{\boldsymbol{\theta}}_k$.

Therefore, the overall prediction for $y^*$ is given by

$$\hat{y}^* = E(y^*|\mathcal{D}) = \sum_{k=1}^{K} \hat{\alpha}_{M+1,k}\ \hat{y}_k^*, \tag{13}$$

$$\hat{\sigma}^{*2} = Var(y^*|\mathcal{D}) = \sum_{k=1}^{K} \hat{\alpha}_{M+1,k}\ \hat{\sigma}_k^{*2} + \sum_{k=1}^{K} \hat{\alpha}_{M+1,k}\ \hat{y}_k^{*2} - \hat{y}^{*2}, \tag{14}$$

where $\hat{\alpha}_{M+1,k}$ is the conditional probability that the $(M+1)$-th curve belongs to the $k$-th component. It can be estimated by

$$\hat{\alpha}_{M+1,k} = E_{\hat{\Theta}}(z_{M+1,k}|\mathcal{D}) = \frac{\hat{\pi}_{M+1,k}\ p(\boldsymbol{y}_{M+1}|\hat{\boldsymbol{B}}_k, \hat{\boldsymbol{\theta}}_k, \boldsymbol{x}_{M+1})}{\sum_{j=1}^{K}[\hat{\pi}_{M+1,j}\ p(\boldsymbol{y}_{M+1}|\hat{\boldsymbol{B}}_j, \hat{\boldsymbol{\theta}}_j, \boldsymbol{x}_{M+1})]},$$

where

$$\hat{\pi}_{M+1,k} = \exp\{\boldsymbol{v}_{M+1}{}'\hat{\boldsymbol{\gamma}}_k\}/[1 + \sum_{j=1}^{K-1} \exp\{\boldsymbol{v}_{M+1}{}'\hat{\boldsymbol{\gamma}}_j\}]. \tag{15}$$

The second type of prediction is to predict for a completely new batch, which is one of the major problems we want to address in this paper. We still refer to the new batch as the $(M+1)$-th batch. The observed data $\mathcal{D}$ include the training data collected from batches 1 to $M$, and for the $(M+1)$-th batch we only observe the batch-based covariates $\boldsymbol{v}_{M+1}$ and $\boldsymbol{u}_{M+1}$. We want to predict $y^*$ at $(t^*, \boldsymbol{x}^*)$ in the $(M+1)$-th batch. Since we have not observed any data for the response variable in the $(M+1)$-th batch, we cannot use the result discussed above which is based on the assumption that the covariance part of $\boldsymbol{y}_{M+1}$ and $y^*$ comes from the same Gaussian process. However, batches $1, \ldots, M$ provide an empirical distribution of the set of all possible batches,

$$P(y^* \text{ belongs to the } m\text{-th batch}) = w_m. \tag{16}$$

Shi *et al.* (2006) assumed that $w_m = 1/M$, i.e., each of the training batches is given the same 'weight' in predicting the new data. In the paraplegia example, this means that we use the information from each patient in the database equally to predict the trajectory of a completely new patient. This may lead to a very inaccurate result since the natures of the patients in the database may be quite different. A better way is to use the data collected from patients who are similar to the new patient to do prediction. In other words, we should give relatively large weights in the empirical distribution (16) to training batches 'close' to the new patient and small weights for the others. The closeness of two batches is taken to be equivalent to how close the distribution of $z_m$ is to the distribution of $z_{M+1}$, which can be measured by Kullback-Leibler divergence:

$$KL(z_{M+1}, z_m) = \sum_{k=1}^{K} \hat{\pi}_{M+1,k} \log \frac{\hat{\pi}_{M+1,k}}{\hat{\pi}_{mk}}, \quad m = 1, \ldots, M,$$

where $\hat{\pi}_{mk}$ is represented by (5) with the parameter evaluated at $\hat{\boldsymbol{\gamma}}_k$. Since $KL(z_{M+1}, z_m)$ is not symmetric, we can also use $KL(z_{M+1}, z_m) + KL(z_m, z_{M+1})$ or some other quantities in practice. The weights in (16) are therefore defined as

$$w_m \propto 1/KL(z_{M+1}, z_m).$$

The Kullback-Leibler divergence is equal to zero for two identical distributions. If the distribution of $M_0$, say, $z_m$'s is exactly the same as that of $z_{M+1}$, we define the weights for those batches to be $1/M_0$, and the weights for the other batches to be zero.

Let $y_m^*$ and $\sigma_m^{*2}$ be the prediction mean and variance if the new batch belongs to the $m$-th batch, which can be calculated by (13) and (14). Then the overall prediction

for $y^*$ is given by

$$\hat{y}^* = E(y^*|\mathcal{D}) = \sum_{m=1}^{M} w_m y_m^*, \tag{17}$$

$$\hat{\sigma}^{*2} = Var(y^*|\mathcal{D}) = \sum_{m=1}^{M} w_m \sigma_m^{*2} + \sum_{m=1}^{M} w_m y_m^{*2} - \hat{y}^{*2}. \tag{18}$$

## 2.4 Curve clustering and model selection

The problem of curve clustering has been drawn attention recently. However, most of the existing methods are essentially based on the shapes of the curves; see for example, James and Sugar (2003). In this section, we will cluster the batches based on the relationships between the response curves and the input covariates, not just the shapes of the response curves. Gaffney and Smyth (2003) discussed a similar problem but they assume a (functional) linear regression model. Here, we assume a nonlinear and nonparametric model as we discussed in the previous sections.

Suppose that the data are generated according to a mixture of Gaussian process functional regressions with $K$ components, as defined in (4). The model can then be fitted as in Section 2.2 and we denote the estimates of the parameters by $\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{B}}_k, \hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\gamma}}_k,\ k = 1, \ldots, K\}$ with $\hat{\boldsymbol{\gamma}}_K = 0$. Thus, given an observed curve, $\boldsymbol{y}^*$ say, its corresponding functional input covariates $\boldsymbol{x}^*$ and the batch-based covariates $\boldsymbol{v}^*$ and $\boldsymbol{u}^*$, the posterior distribution of the latent variable $z^* = (z_1^*, \ldots, z_K^*)'$ is given by

$$P(z_k^* = 1|\boldsymbol{y}^*) = \frac{\pi_k^* p(\boldsymbol{y}^*|\hat{\boldsymbol{B}}_k, \hat{\boldsymbol{\theta}}_k, \boldsymbol{x}^*)}{\sum_{j=1}^{K} \pi_k^* p(\boldsymbol{y}^*|\hat{\boldsymbol{B}}_j, \hat{\boldsymbol{\theta}}_j, \boldsymbol{x}^*)},$$

where

$$\pi_k^* = \frac{\exp\{\boldsymbol{v}^{*'}\hat{\boldsymbol{\gamma}}_k\}}{1 + \sum_{j=1}^{K-1} \exp\{\boldsymbol{v}^{*'}\hat{\boldsymbol{\gamma}}_j\}}.$$

The curve can be classed as belonging to the $k^*$-th cluster if $P(z_k^* = 1|\boldsymbol{y}^*)$ takes its maximum value at $k = k^*$ for $k = 1, \ldots, K$.

An important but difficult problem in cluster analysis is to choose the number of clusters, i.e., the value of $K$. In this paper, we use Bayes factors. Since its value is difficult to calculate exactly, we use its approximate form, BIC (Schwarz, 1978). If we recall that $L(\cdot)$ is the log-likelihood function as defined in (10) and let $\hat{\mathbf{\Theta}}$ be the maximum likelihood estimate, the BIC value is given by

$$\text{BIC} = -2L(\hat{\mathbf{\Theta}}) + G\log(N),$$

where $G$ is the total number of parameters and $N = N_1 + \cdots + N_M$ is the sample size.

An alternative approach is to assume that $K$ is a random number, whose value can be estimated from its posterior distribution. In this case, a reverse jump algorithm can be used (Green, 1995). However, since the number of unknown parameters involved in each component is usually large, and the conditional density functions are quite complicated particularly for the parameters involved in the covariance structure, the computation is likely to be tedious. We will not discuss this approach in this paper.

# 3 Examples

In this section we demonstrate our methods with some simulated data and real data.

## 3.1 Simulation study for prediction and multiple-step-ahead forecasting

We first consider a mixture model with two components, where the mean functions are $\mu_{m1}(t) = 0.5\sin((0.5t)^3)$ and $\mu_{m2}(t) = -3.0\exp(-t^2/8)/\sqrt{2\pi}+0.7$, while the covariance

functions are

$$C(x_i, x_j) = v_1 \exp\Big( -\frac{1}{2} w_1 (x_i - x_j)^2 \Big) + a_1 x_i x_j + v_0 \delta_{ij}, \tag{19}$$

with $\boldsymbol{\theta}_1 = (1.0, 0.0, 0.2, 0.0025)'$ and $\boldsymbol{\theta}_2 = (0.5, 0.0, 0.25, 0.0025)'$ respectively, where $\boldsymbol{\theta} = (w_1, a_1, v_1, v_0)'$. The data points $t = t_i$ are 50 equally spaced points between -4 and 4. The functional covariate $x$ for each batch is a sample of another Gaussian process on $(-4, 4)$. The mixing coefficients are given by (5) with $\gamma_1 = 2.0$. Sixty curves are generated and are presented in Figure 1(b), where $\boldsymbol{v}_m = 2.0$ for batches 1 to 30 (i.e. $m = 1, \ldots, 30$) and $\boldsymbol{v}_m = -1$ for batches 31 to 60 (i.e. $m = 31, \ldots, 60$).

### 3.1.1 Prediction

In this simulated example, we have bivariate functional covariates $(t, x(t))$, with the mean structure depending on $t$ and the covariance structure depending on $(t, x(t))$. Four models are compared: mixture of GPFRs (models (4) and (5), denoted by *mix-gpfr*), single GPFR ($K = 1$, denoted by *gpfr*), mixture of mean models (model (2) includes mean structure only, denoted by *mix-mean*) and mixture of GP models (model (2) includes a constant mean and covariance structure modelled by a Gaussian process regression model, denoted by *mix-gp*). To assess the performance of these models, different types of prediction are calculated and compared with the real data. An example is shown in Figure 2, where the first two columns correspond to Type I prediction as discussed in Section 2.3, i.e., we have observed some data in a new batch. However, the first column corresponds to the problem of interpolation, in which we randomly select half of the generated data in the whole range $(-4, 4)$ as training data, and the rest are used as test data; the second column concerns extrapolation, in which we se-

lect the data generated in the range $(-4, 0)$ as training data, and predict the curve located in the range $(0, 4)$. The third column corresponds to Type II prediction: there are no training data observed for this new batch and we need to predict a completely new curve. In Figure 2, the solid line stands for the real simulated curve, the dashed line is the prediction and the dotted lines are pointwise 95% confidence bands. We also calculated the values of root of mean squared errors (rmse) and the correlation coefficient between the real data and the prediction. A simulation study with 50 replications was conducted, and the results are reported in Table 1. From Figure 2 and Table 1, we have the following findings. First, for the mixture data, a single GPFR model gives uniformly poor results when compared to a mixture of GPFR models. It is therefore essential to consider mixture models if we think that there are mixture structures underlying the data. We will give a further discussion in Section 3.2 when we consider model selection. Secondly, comparing the cases modelling both mean and covariance structures (*mix-gpfr*), with cases modelling mean structure only (*mix-mean*) and cases modelling covariance structure only (*mix-gp*), we find that *mix-gpfr* performs uniformly better than the others. For the problem of interpolation, *mix-gp* also gives a quite good result although not as good as *mix-gpfr*, and both of them perform much better than *mix-mean*. For the problem of extrapolation, *mix-gpfr* still performs better than *mix-mean*, but *mix-gp* usually gives bad results, especially when the test data point for prediction moves away from the training data (see Figure 2(k) and the values of $rmse$ and $r$ in Table 1). This finding coincides with the results for the related single models discussed in Shi *et al.* (2006).

### 3.1.2 Multiple-step-ahead forecasting

As discussed in Shi *et al.* (2006), the GPFR model is particular useful in multiple-step-ahead forecasting. We conduct a simulation study to assess the performance of mixture models here. We still use the data simulated in the previous subsection, but will use the data collected up to time $t_i$, say, to predict the value $y(t_{i+\nu})$ in $\nu$-step-ahead forecasting. Sixty batches of mixture data are simulated as training data. To predict $y(t_{i+\nu})$, we use $y(t)$ at $t = t_i$ as an additional covariate, i.e., we have three-dimensional covariates $(t_{i+\nu}, x_{i+\nu}, y_i)$, where $t_i$ is used in the mean structure and the others are used in the covariance structure. For comparison, four models as discussed in the previous subsection are applied to the same data-set.

A simulation study with 100 replications was conducted, and the results related to $1-$, $3-$ and 6-step-ahead forecasts are reported in Table 1. The result of one typical replication is presented in Figure 3. Multiple-step-ahead forecasting is similar to the prediction problem of extrapolation, and the simulation results here are consistent with the results discussed in the previous subsection. First of all, *mix-gpfr* gives a much better result than a single GPFR, and is consistently better than *mix-mean* and *mix-gp*. When $\nu$ is small, as in 1-step-ahead forecasting, *mix-gp* still gives a reasonably good result and is better than *mix-mean*, although not as good as *mix-gpfr*. However, when $\nu$ is large, as in 6-step-ahead forecasting, *mix-mean* is better than *mix-gp*.

## 3.2 Clustering and model selection

We now consider the problem of curve clustering. We first calculate the posterior distribution of $\boldsymbol{z}$ and then classify the curve by using the largest posterior probability.

We consider a mixture of two Gaussian processes which have mean functions $\mu(t) = \exp(t/5) - 1.5$ and $\mu(t) = 0.8\text{atan}(t)$, respectively. The covariance functions are given by (19), and the mean components and 60 sample curves are given in Figure 4. From Figure 4(a), the two mean components are quite similar and, thus, so are the shapes of all the curves. Three methods, *mix-gpfr*, *mix-mean* and *mix-gp*, are used to train the data and produce a clustering. The error rate for these three methods are 2%, 13.5% and 31.5%, respectively. Here, the *mix-mean* model clusters the curves according to their shapes, while the *mix-gpfr* model makes clusters based on the functional regression relationship between the response curves and the two-dimensional functional covariates $(t, x(t))$. It is therefore not surprising that the performance of *mix-gpfr* is much better than that of *mix-mean* in this example. For the sample curves shown in Figure 1(b), *mix-mean* gives a better result with an error rate of only 1.5%, since the shapes of the two components are more distinguishable than in this example. However, *mix-gpfr* gives a even better result, with zero error rate.

The BIC approach is used for model selection in this paper. For the data discussed in the previous subsection and the previous paragraph, BIC takes the smallest value at $k = 2$. For the more complicated example shown in Figure 5(a), which contains 90 curves with three components, the values of BIC are given in Figure 5(b). It indicates that BIC works well for model selection of functional data. Curve clustering is also conducted for this example. The error rates for *mix-gpfr*, *mix-mean* and *mix-gp* are 0.5%, 35.5% and 11%, respectively. Figure 5(a) shows that the third component is hardly recognised from their shapes, which is why about one-third of the curves are misclassified by *mix-mean*.

## 3.3 Modelling of standing-up manoeuvres

Our application involves the analysis of the standing-up manoeuvre in paraplegia, considering the body supportive forces as a potential feedback source in functional electrical stimulation (FES)-assisted standing-up. The analysis investigates the significance of arm, feet and seat reaction signals for the reconstruction of the human body centre-of-mass (COM) trajectory. The motion kinematics, reaction forces and other quantities were measured for modelling; for more details see Kamnik *et al.* (1999, 2005). Here we model the vertical trajectory of the body COM as output, and select 8 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 each patient. Since the time scales are different for different standings-up, some registration methods are used (Ramsay and Silverman, 1997). This data set has been analysed by Shi *et al* (2006). In this paper, we are interested in the vertical trajectories of the body COM after registration for 40 standings-up, 5 for each of 8 patients.

As discussed before, there is severe heterogeneity because of the different circumstances of different patients. The allocation model uses the patient's height, weight and standing-up strategy as $\boldsymbol{v}_m$, and the mean structure uses the patient's height as $\boldsymbol{u}_m$. All eight functional covariates are used in modelling the covariance structure. We first use the mixture of GPFR models with two components, and compare it with the results obtained by using a single GPFR model ($K = 1$). We use the data collected from seven of the eight patients as training data, and then use them to predict the

standing-up curves of the eighth patient as completely new curves (Type II prediction). We calculate predictions for all eight patients in turn, and the results are given in Table 2. The mixture model gives better results than the single model except for patient 'ak'. On average, the mixture model reduces *rmse* by about 13% compared to the single GPFR model. As an example, Figures 6(a) and 6(b) show the predictions and the true values for patient 'mk' obtained by using the mixture model and the single model respectively.

The values of BIC are shown in Figure 6(c). It shows that the two-component mixture model has the smallest BIC value. When we use the patient's height, weight and standing-up strategy as covariates in the allocation model, patients 'ak', 'mk', 'sb' and 'tm' are clustered in one group and the others are in the other group. The scatter-plot of their weights and heights is given in Figure 6(d). It is interesting to note that the patients in the cluster one are relatively thin, while the patients in the other cluster are relatively heavy comparing to their heights.

We have data collected from only eight patients. If more data were available, we would have more information for clustering and would be able to give better predictions for the patients like 'ak'.

# 4 Discussion

In model (2), the mean structure $\mu_m(t)$ can be replaced by $\mu_m(\boldsymbol{x})$ if we have some knowledge about the mean relationship between the response curve and the covariates $\boldsymbol{x}$, for example a linear regression $\mu(\boldsymbol{x}) = \boldsymbol{h}'(\boldsymbol{x})\boldsymbol{\beta}$ as used in Kennedy and O'Hagan (2001) with a vector of known functions $\boldsymbol{h}(\cdot)$, or a varying-coefficient linear model

$\mu(\boldsymbol{x}) = \sum_j g_j(\boldsymbol{\beta}'\boldsymbol{x})x_j$ with unknown functions $g_j(\cdot)$ (see for example Fan, Yao and Cai, 2003). However, it is usually difficult to justify such mean structures. In this case, we can use the nonparametric model defined in (2), whose mean structure $\mu_m(t)$ depends on some non-functional covariates $\boldsymbol{u}_m$ only. The mean structure in this model is usually easy to understand and justify. For our paraplegia example, the functional covariates are forces and torques, etc, and the output is the trajectory of body COM. Since the relationship between those functional variables is unknown, it is difficult to justify any form of mean structure $\mu_m(\boldsymbol{x})$. However, the meaning of $\mu_m(t)$ in (2) is clear. For example, if we include only a one-dimensional covariate $\boldsymbol{u}_m$ of the patient's sex, then the mean structure just gives the point-wise averages of all the curves for men and women respectively. The regression relationship between the response curve $y$ and the functional covariates $\boldsymbol{x}$ will be described mainly by the covariance structure adjusted by the men's or the women's mean curve. The model therefore involves a very weak assumption and can be used in a wide range as a nonparametric and nonlinear approach.

As discussed in Section 3, the mixture model gives much better prediction uniformly than the related single model if there is heterogeneity among curves, which is the case for most of the real data sets. We also discussed a model-based method for the problem of curve clustering, which is conducted based on the relationship between the response curve and the set of the covariate curves nonparametrically. This method differs from most of the existing methods which are based on the shape of the curves. The simulation results given in the last section showed the advantage of our method.

Model selection is an interesting but difficult problem for a mixture model, especially for the models with very complicated forms. We use the BIC to select the number

of mixture components. It works fine for the data discussed in this paper. It might be worth a further research for the model with an unknown number of components by using a reverse jump algorithm (Green, 1995) or by using a birth-death MCMC algorithm (Stephens, 2000; Cappé, Robert and Rydén, 2003).

## Acknowledgements

We would like to thank Professor Mike Titterington of Glasgow University, UK, for very helpful discussion on this work and valuable comments on the manuscript. 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. Dr. Wang is grateful for support of the UK Engineering and Physical Sciences Research Council for grant EPSRC GR/T29253/01.

## References

Cappé, O., Robert, C. P. and Rydén, T. (2003). Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B* **65**, 679-700.

Fan, J., Yao, Q. and Cai, Z. (2003). Adaptive varying-coefficient linear models. *Journal of the Royal Statistical Society, Series B* **65**, 57-80.

Faraway, J. (1997). Regression analysis for a functional response. *Technometrics* **39**, 254-261.

Faraway, J. (2001). Modelling hand trajectories during reaching motions. Technical Report #383. Department of Statistics, University of Michi-

gan.

Fernandez, C. and Green, P. (2002). Modelling spatially correlated data via mixtures: a Bayesian approach. *Journal of the Royal Statistical Society, Series B* **64**, 805-826.

Gaffney, S. and Smyth, P. (2003). Curve clustering with random effects regression mixtures. In C. Bishop and B. Frey (Eds.), *Proc. Ninth Inter. Workshop on Artificial Intelligence and Statistics.*

Green, P. J. (1995). Reversible jump MCMC computation and Bayesian model determination. *Biometrika* **82**, 711-732.

Green, P. J. and Richardson, S. (2000). Spatially correlated allocation models for count data. Technical report. University of Bristol.

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. (2005). Nonlinear modelling of FES-supported standing up in paraplegia for selection of feedback sensors. *IEEE Transactions on Neural Systems & Rehabilitation Engineering* **13**, 40-52.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society, Series B* **63**, 425-464.

James, G. and Sugar, C. (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association* **98**, 397-408.

Louis, T. A. (1982). Finding the observed information matrix when using

the EM algorithm. *Journal of the Royal Statistical Society, Series B* **44**, 226-233.

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 models*. New York: Wiley.

Müller, H.G. (2005). Functional modelling and classification of longitudinal data. *Scandinavian J. Statistics*, **32**, 223-240.

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

Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance nonparametrically when the data are curves. *Journal of the Royal Statistical Society, Series B* **53**, 233-243.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**, 461-464.

Shi, J. Q., Murray-Smith, R. and Titterington, D. M. (2003). Bayesian regression and classification using mixtures of Gaussian process. *International Journal of Adaptive Control and Signal Processing*. **17**, 149-161.

Shi, J. Q., Murray-Smith, R. and Titterington, D. M. (2005). Hierarchical Gaussian process mixtures for regression. *Statistics and Computing* **15**, 31-41.

Shi, J. Q., Wang, B., Murray-Smith, R. and Titterington, D. M. (2006). Gaussian process functional regression modelling for batch data. *Bio-*

*metrics* (in press).

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components–an alternative to reversible jump methods. *The Annals of Statistics*, 40-74.

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

# Appendix

## I. M-step

In M-step, we need to maximise $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)})$ with respect to $\boldsymbol{\gamma}_k$, $\boldsymbol{B}_k$ and $\boldsymbol{\theta}_k$. Unfortunately, there are no analytic solutions, therefore we use the following sub-algorithms.

(1) Maximise $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)})$ with respect to $\boldsymbol{\gamma}_k$ for $k = 1, \ldots, K-1$. It is equivalent to maximising

$$L_1(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{K-1}) \triangleq \sum_{m=1}^{M} \sum_{k=1}^{K} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) \{ \boldsymbol{v}_m{}' \boldsymbol{\gamma}_k - \log \left[ 1 + \sum_{j=1}^{K-1} \exp\{ \boldsymbol{v}_m{}' \boldsymbol{\gamma}_j \} \right] \}$$

which is independent of $\boldsymbol{B}_k$ and $\boldsymbol{\theta}_k$. It is similar to the log-likelihood for a multinomial logit model ($\alpha_{mk}(\boldsymbol{\Theta}^{(i)})$'s are corresponding to the observations) and can be maximised by iteratively re-weighted least square algorithm.

(2) Maximise $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)})$ with respect to $\boldsymbol{B}_k$ and $\boldsymbol{\theta}_k$, $k = 1, \ldots, K$. It is equivalent to maximising

$$L_2(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K) \triangleq \sum_{k=1}^{K} \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) \log p(\boldsymbol{y}_m | \boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m).$$

We denote the covariance matrix by $\boldsymbol{C}_{mk} = (C_{mk}^{ij}),\ i, j = 1, \ldots, N_m$, whose element

$C_{mk}^{ij}$ is calculated by (7), i.e.,

$$C_{mk}^{ij} = v_1^k \exp\Big( -\frac{1}{2}\sum_{q=1}^{Q} w_q^k (x_{miq} - x_{mjq})^2 \Big) + a_1^k \sum_{q=1}^{Q} x_{miq} x_{mjq} + v_0^k \delta_{ij}. \tag{20}$$

Thus the density function of $N_m$-dimensional normal distribution is given by

$$\begin{aligned} p(\boldsymbol{y}_m|\boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{x}_m) &= (2\pi)^{-N_m/2} |\boldsymbol{C}_{mk}|^{-1/2} \\ &\quad \cdot \exp\Big\{ -\frac{1}{2}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)' \boldsymbol{C}_{mk}^{-1} (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m) \Big\}, \end{aligned}$$

where $\boldsymbol{\Phi}_m$ is an $N_m \times D$ matrix given by

$$\begin{pmatrix} \Phi_1(t_{m1}) & \cdots & \Phi_D(t_{m1}) \\ \vdots & \ddots & \vdots \\ \Phi_1(t_{mN_m}) & \cdots & \Phi_D(t_{mN_m}) \end{pmatrix}.$$

Denoting by $\theta_{kj} \in \boldsymbol{\theta}_k$ any parameter involved in (20), after a straightforward calculation we obtain

$$\begin{aligned} \frac{\partial L_2}{\partial \theta_{kj}} &= \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) \big[ -\frac{1}{2} \mathrm{tr}(\boldsymbol{C}_{mk}{}^{-1} \frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}}) \\ &\quad + \frac{1}{2}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)' \boldsymbol{C}_{mk}^{-1} \frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}} \boldsymbol{C}_{mk}^{-1} (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m) \big], \end{aligned} \tag{21}$$

$$\frac{\partial L_2}{\partial \mathrm{vec}(\boldsymbol{B}_k)} = \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)' \boldsymbol{C}_{mk}^{-1} (\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m')', \tag{22}$$

where, $\mathrm{vec}(A)$ denotes the stacked columns of $A$ and $\otimes$ denotes Kronecker product.

Letting $\partial L_2 / \partial \mathrm{vec}(\boldsymbol{B}_k) = 0$, we get, for $k = 1, \ldots, K$,

$$\begin{aligned} \mathrm{vec}(\boldsymbol{B}_k) &= \Big\{ \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) (\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m') \boldsymbol{C}_{mk}^{-1} (\boldsymbol{u}_m{}' \otimes \boldsymbol{\Phi}_m) \Big\}^{-1} \\ &\quad \cdot \Big\{ \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)}) (\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m') \boldsymbol{C}_{mk}^{-1} \boldsymbol{y}_m \Big\} \\ &\triangleq F(\boldsymbol{\theta}_k). \end{aligned} \tag{23}$$

Since explicit expression for optimising $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(i)})$ in $\boldsymbol{\theta}_k$ does not exist, we use an iterative procedure to complete Stage (2) as follows: for $k = 1, \ldots, K$,

(2.1) Update $\boldsymbol{B}_k$ by (23) given $\boldsymbol{\theta}_k$;

(2.2) Update $\boldsymbol{\theta}_k$ by maximising $L_2$ given $\boldsymbol{B}_k$.

To speed up convergence, we usually repeat (2.1) and (2.2) for several times. In (2.1), the gradient (21) is used such that the maximisation procedure is implemented most efficiently.

## II. Standard errors

Louis (1982) has provided a powerful numerical technique for computing the observed information matrix and standard errors when the EM algorithm is used to find maximum likelihood estimates. We denote by $L_c^m(\boldsymbol{\Theta})$ the log-likelihood for the complete data $(\boldsymbol{y}_m, z_m)$ of the $m$-th batch. Since our observed curves are independent, we have $L_c(\boldsymbol{\Theta}) = \sum_{m=1}^{M} L_c^m(\boldsymbol{\Theta})$.

From (11), we have

$$\begin{aligned}
\frac{\partial L_c^m}{\partial \boldsymbol{\gamma}_k} &= (z_{mk} - \pi_{mk})\boldsymbol{v}_m', && (24)\\
\frac{\partial L_c^m}{\partial \theta_{kj}} &= z_{mk}\Big[ -\frac{1}{2}\mathrm{tr}({\boldsymbol{C}_{mk}}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}}) \\
&\quad +\frac{1}{2}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m\boldsymbol{B}_k\boldsymbol{u}_m)'\boldsymbol{C}_{mk}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}}\boldsymbol{C}_{mk}^{-1}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m\boldsymbol{B}_k\boldsymbol{u}_m)\Big], && (25)\\
\frac{\partial L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_k)} &= z_{mk}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m\boldsymbol{B}_k\boldsymbol{u}_m)'\boldsymbol{C}_{mk}^{-1}(\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m')'. && (26)
\end{aligned}$$

Note that $\theta_{kj}$ in (25) denotes any element of $\boldsymbol{\theta}_k$ involved in (20).

Let $S_m(\boldsymbol{\Theta})$ be the gradient vectors of $L_c^m(\boldsymbol{\Theta})$ and $\hat{\boldsymbol{\Theta}}$ be the maximum likelihood estimates, then we have

$$S_m(\boldsymbol{\Theta}) = (\frac{\partial L_c^m}{\partial \boldsymbol{\gamma}_1}, \ldots, \frac{\partial L_c^m}{\partial \boldsymbol{\gamma}_{K-1}}, \frac{\partial L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_1)}, \ldots, \frac{\partial L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_K)}, \frac{\partial L_c^m}{\partial \boldsymbol{\theta}_1}, \ldots, \frac{\partial L_c^m}{\partial \boldsymbol{\theta}_K})',$$

and it is easy to get $E_{\hat{\boldsymbol{\Theta}}}\{S_m(\hat{\boldsymbol{\Theta}})\}$ since $E_{\hat{\boldsymbol{\Theta}}}\{z_{mk}\} = \alpha_{mk}(\hat{\boldsymbol{\Theta}})$ from (12). The computation of $E_{\hat{\boldsymbol{\Theta}}}\{S_m(\hat{\boldsymbol{\Theta}})S_m'(\hat{\boldsymbol{\Theta}})\}$ is tedious but straightforward by using $E_{\hat{\boldsymbol{\Theta}}}\{z_{mk}^2\} = \alpha_{mk}(\hat{\boldsymbol{\Theta}})$

and $E_{\hat{\boldsymbol{\Theta}}}\{z_{mk}z_{mj}\} = 0$ for $k \neq j$.

Furthermore, it follows from (24) - (26) that

$$\begin{aligned}
\frac{\partial^2 L_c^m}{\partial {\boldsymbol{\gamma}_k}^2} &= (\pi_{mk}^2 - \pi_{mk})\boldsymbol{v}_m {\boldsymbol{v}_m}', \\
\frac{\partial^2 L_c^m}{\partial \boldsymbol{\gamma}_k \partial \boldsymbol{\gamma}_j} &= \pi_{mk}\pi_{mj}\boldsymbol{v}_m {\boldsymbol{v}_m}', \\
\frac{\partial^2 L_c^m}{\partial [\mathrm{vec}(\boldsymbol{B}_k)]^2} &= -z_{mk}(\boldsymbol{u}_m \otimes \boldsymbol{\Phi}'_m)\boldsymbol{C}_{mk}^{-1}(\boldsymbol{u}_m \otimes \boldsymbol{\Phi}'_m)', \\
\frac{\partial^2 L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_k)\partial \theta_{kj}} &= z_{mk}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)'\boldsymbol{C}_{mk}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}}\boldsymbol{C}_{mk}^{-1}(\boldsymbol{u}_m \otimes \boldsymbol{\Phi}'_m)',
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial^2 L_c^m}{\partial \theta_{ki}\partial \theta_{kj}} = z_{mk}\Bigg\{ &-\frac{1}{2}\mathrm{tr}\Big[{\boldsymbol{C}_{mk}}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}}{\boldsymbol{C}_{mk}}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{ki}} + \boldsymbol{C}_{mk}^{-1}\frac{\partial^2 \boldsymbol{C}_{mk}}{\partial \theta_{ki}\partial \theta_{kj}}\Big] \\
&+ (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)'\boldsymbol{C}_{mk}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{ki}}\boldsymbol{C}_{mk}^{-1}\frac{\partial \boldsymbol{C}_{mk}}{\partial \theta_{kj}}\boldsymbol{C}_{mk}^{-1}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m) \\
&+ \frac{1}{2}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)'\boldsymbol{C}_{mk}^{-1}\frac{\partial^2 \boldsymbol{C}_{mk}}{\partial \theta_{ki}\partial \theta_{kj}}\boldsymbol{C}_{mk}^{-1}(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m)\Bigg\}.
\end{aligned}$$

The other second-order derivatives, such as

$$\frac{\partial^2 L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_k)\partial \boldsymbol{\gamma}_j}, \ \frac{\partial^2 L_c^m}{\partial \boldsymbol{\gamma}_k \partial \boldsymbol{\theta}_j}, \ \frac{\partial^2 L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_k)\partial \mathrm{vec}(\boldsymbol{B}_j)}(j \neq k), \ \frac{\partial^2 L_c^m}{\partial \mathrm{vec}(\boldsymbol{B}_k)\partial \boldsymbol{\theta}_j}(j \neq k), \ \frac{\partial^2 L_c^m}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_j}(j \neq k)$$

are all zero.

Therefore, setting $T_m(\boldsymbol{\Theta})$ being the negatives of the second order derivative matrices of $L_c^m(\boldsymbol{\Theta})$, it is straightforward to compute $E_{\hat{\boldsymbol{\Theta}}}\{T_m(\hat{\boldsymbol{\Theta}})\}$ thanks to $E_{\hat{\boldsymbol{\Theta}}}\{z_{mk}\} = \alpha_{mk}(\hat{\boldsymbol{\Theta}})$.

From Louis (1982) the observed information matrix can be computed as

$$\begin{aligned}
I_Y = &\sum_{m=1}^{M} E_{\hat{\boldsymbol{\Theta}}}\{T_m(\hat{\boldsymbol{\Theta}})\} - \sum_{m=1}^{M} E_{\hat{\boldsymbol{\Theta}}}\{S_m(\hat{\boldsymbol{\Theta}})S'_m(\hat{\boldsymbol{\Theta}})\} \\
&- 2\sum_{l<m}^{M} E_{\hat{\boldsymbol{\Theta}}}\{S_l(\hat{\boldsymbol{\Theta}})\}E_{\hat{\boldsymbol{\Theta}}}\{S_m(\hat{\boldsymbol{\Theta}})\}'.
\end{aligned}$$

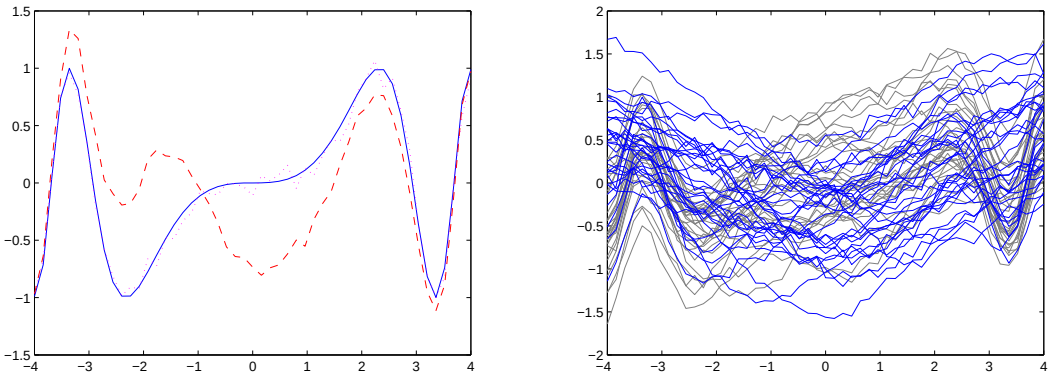
The observed information matrix $I_Y$ can then be inverted to obtain the standard errors.

Table 1: Simulation study: values of $rmse$ and $r$

| | Prediction | | | | | |
|---|---|---|---|---|---|---|
| | Type I | | Type II | | New curve | |
| | $rmse$ | $r$ | $rmse$ | $r$ | $rmse$ | $r$ |
| *mix-gpfr* | 0.0595 | 0.9871 | 0.2492 | 0.8390 | 0.3970 | 0.7680 |
| *gpfr* | 0.1229 | 0.9425 | 0.4124 | 0.6442 | 0.4575 | 0.6082 |
| *mix-mean* | 0.3809 | 0.6331 | 0.3786 | 0.7775 | 0.4023 | 0.7557 |
| *mix-gp* | 0.1926 | 0.8564 | 0.4729 | 0.1402 | 0.5018 | 0.4219 |
| | $\nu$-step ahead forcasting | | | | | |
| | 1-step | | 3-step | | 6-step | |
| | $rmse$ | $r$ | $rmse$ | $r$ | $rmse$ | $r$ |
| *mix-gpfr* | 0.0712 | 0.9844 | 0.1081 | 0.9589 | 0.2128 | 0.8735 |
| *gpfr* | 0.1080 | 0.9648 | 0.2355 | 0.7873 | 0.3799 | 0.5567 |
| *mix-mean* | 0.4291 | 0.5483 | 0.4106 | 0.6388 | 0.4568 | 0.6727 |
| *mix-gp* | 0.1288 | 0.9504 | 0.3744 | 0.2722 | 0.4411 | 0.2281 |

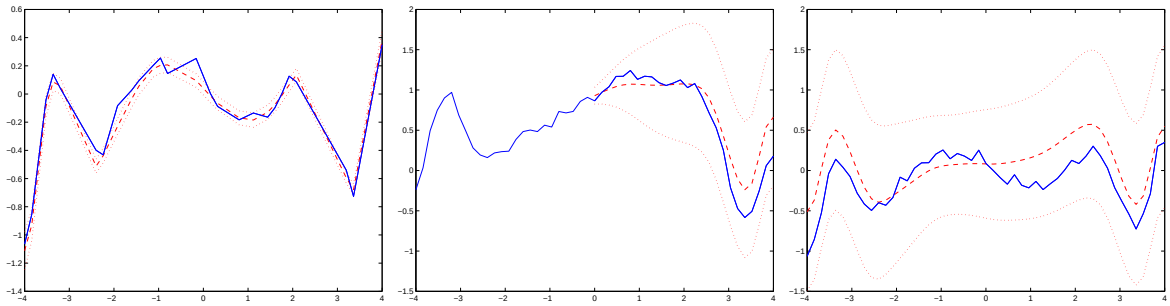
Table 2: Paraplegia data: values of $rmse$ and $r$ for prediction

| subject | *mix-gpfr* | | *gpfr* | |
|---|---|---|---|---|
| | $rmse$ | $r$ | $rmse$ | $r$ |
| Average | 0.2458 | 0.9868 | 0.2815 | 0.9828 |
| ak | 0.3172 | 0.9852 | 0.2125 | 0.9812 |
| bj | 0.2949 | 0.9889 | 0.3810 | 0.9837 |
| mk | 0.1008 | 0.9925 | 0.2450 | 0.9913 |
| mt | 0.3030 | 0.9896 | 0.2661 | 0.9890 |
| sb | 0.2799 | 0.9739 | 0.3363 | 0.9601 |
| tm | 0.1763 | 0.9911 | 0.2706 | 0.9815 |
| zb | 0.1517 | 0.9954 | 0.1836 | 0.9915 |
| zj | 0.3431 | 0.9774 | 0.3573 | 0.9841 |

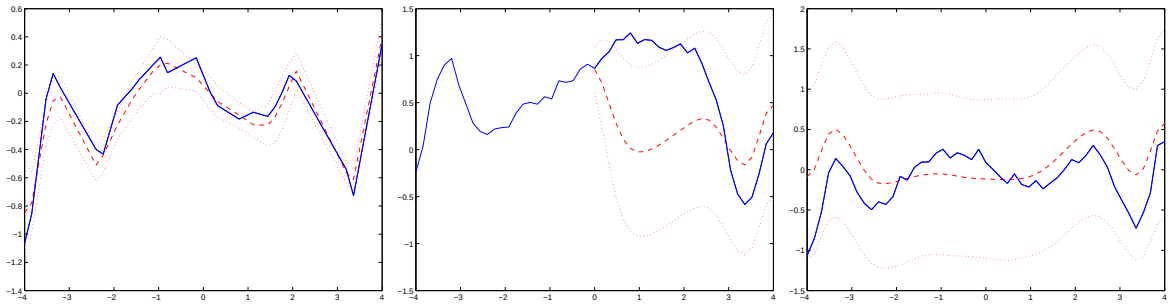



(a) (b)

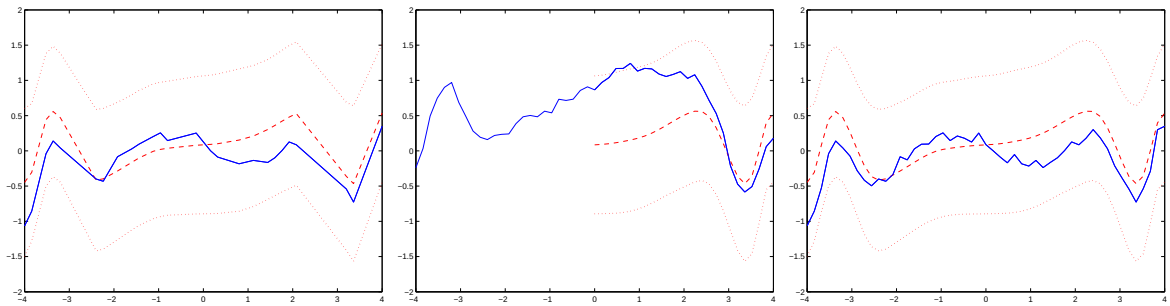
Figure 1: The sample curves. (a) Solid line—the true mean curve; dotted line—the curve with random errors; dashed line—the curve with errors having GP covariance structure depending on $x$. (b) Sample curves of mixture with two components.



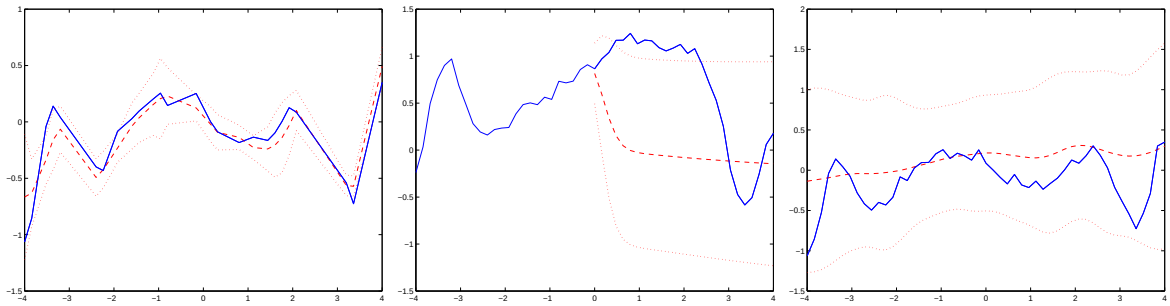
(a) *mix-gpfr*, *rmse*=.0620 (b) *mix-gpfr*, *rmse*=.2278 (c) *mix-gpfr*, *rmse*=.2888



(d) *gpfr*, *rmse*=.0983 (e) *gpfr*, *rmse*=.7408 (f) *gpfr*, *rmse*=.3655

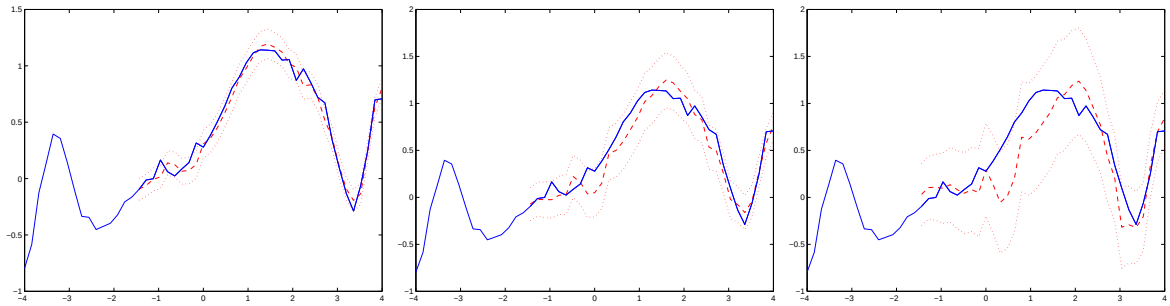


(g) *mix-mean*, *rmse*=.3346 (h) *mix-mean*, *rmse*=.6615 (i) *mix-mean*, *rmse*=.3055

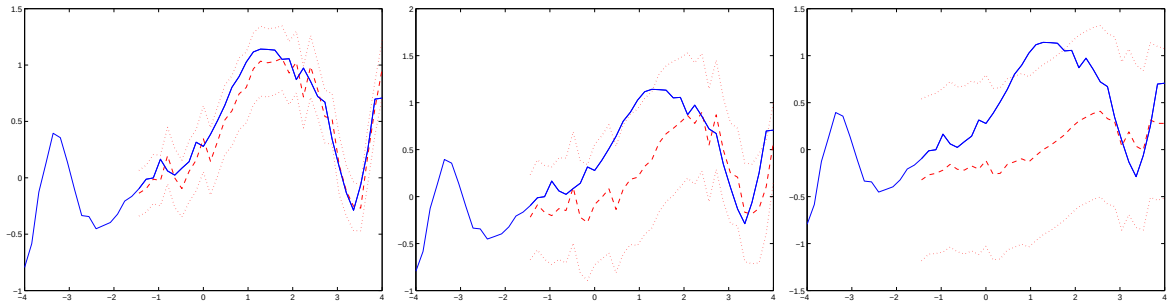


(j) *mix-gp*, *rmse*=.1470 (k) *mix-gp*, *rmse*=.8671 (l) *mix-gp*, *rmse*=.3536

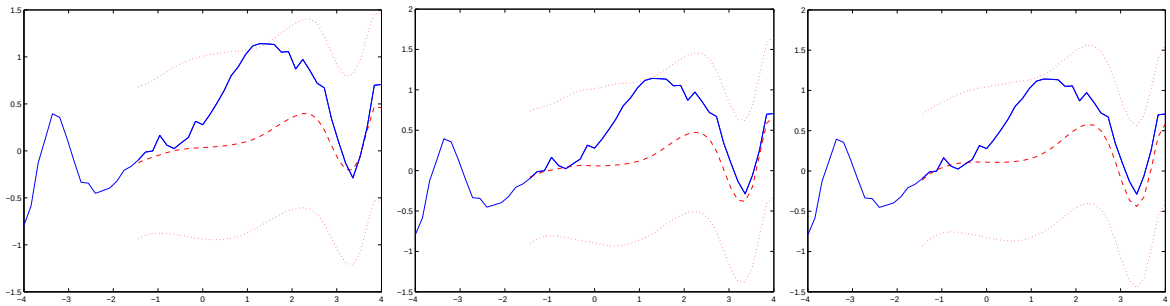
Figure 2: Prediction: solid line is the real curve, dashed line is prediction and dotted lines are its 95% confidence bands, where the first column is corresponding to the problem of interpolation, the second is for extrapolation and the third is for a completely new curve.



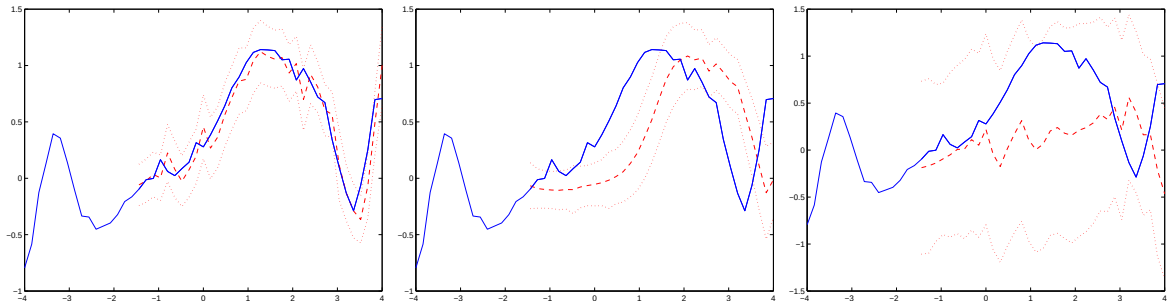
(a) 1-, *mix-gpfr*, *rmse*=.0759 (b) 3-, *mix-gpfr*, *rmse*=.1306 (c) 6-, *mix-gpfr*, *rmse*=.2603



(d) 1-, *gpfr*, *rmse*=.1428 (e) 3-, *gpfr*, *rmse*=.4220 (f) 6-, *gpfr*, *rmse*=.6334

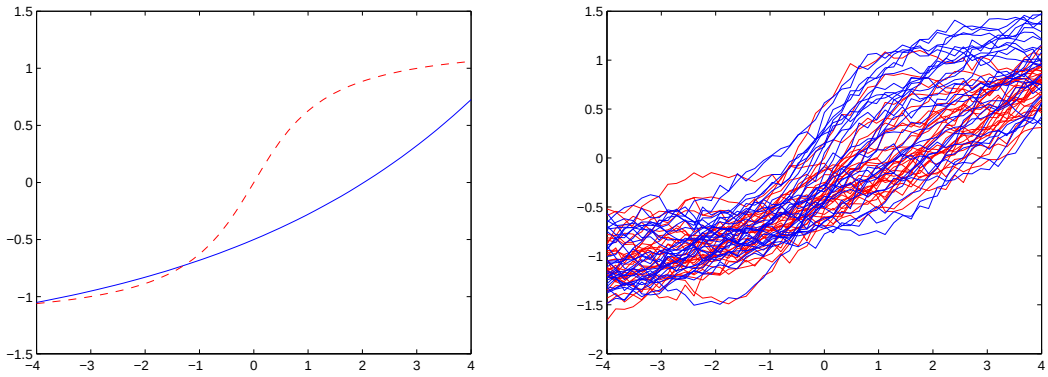


(g) 1-, *mix-mean*, *rmse*=.5103 (h) 3-, *mix-mean*, *rmse*=.4851 (i) 6-, *mix-mean*, *rmse*=.4515



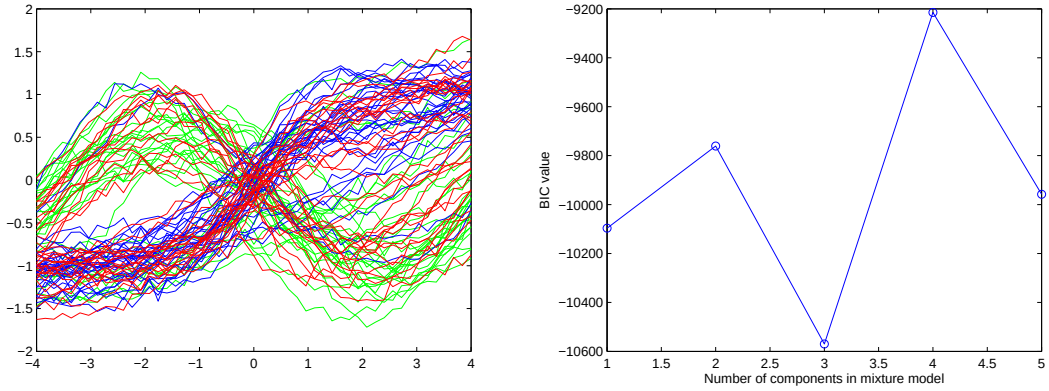
(j) 1-, *mix-gp*, *rmse*=.1423 (k) 3-, *mix-gp*, *rmse*=.4948 (l) 6-, *mix-gp*, *rmse*=.6154

Figure 3: $\nu$-step ahead forcasting: solid line is the real curve, dashed line is prediction and dotted lines are its 95% confidence bands, where the first column is corresponding to the 1-step ahead forcasting, and the second and third are for 3- and 6-step ahead forcasting respectively.



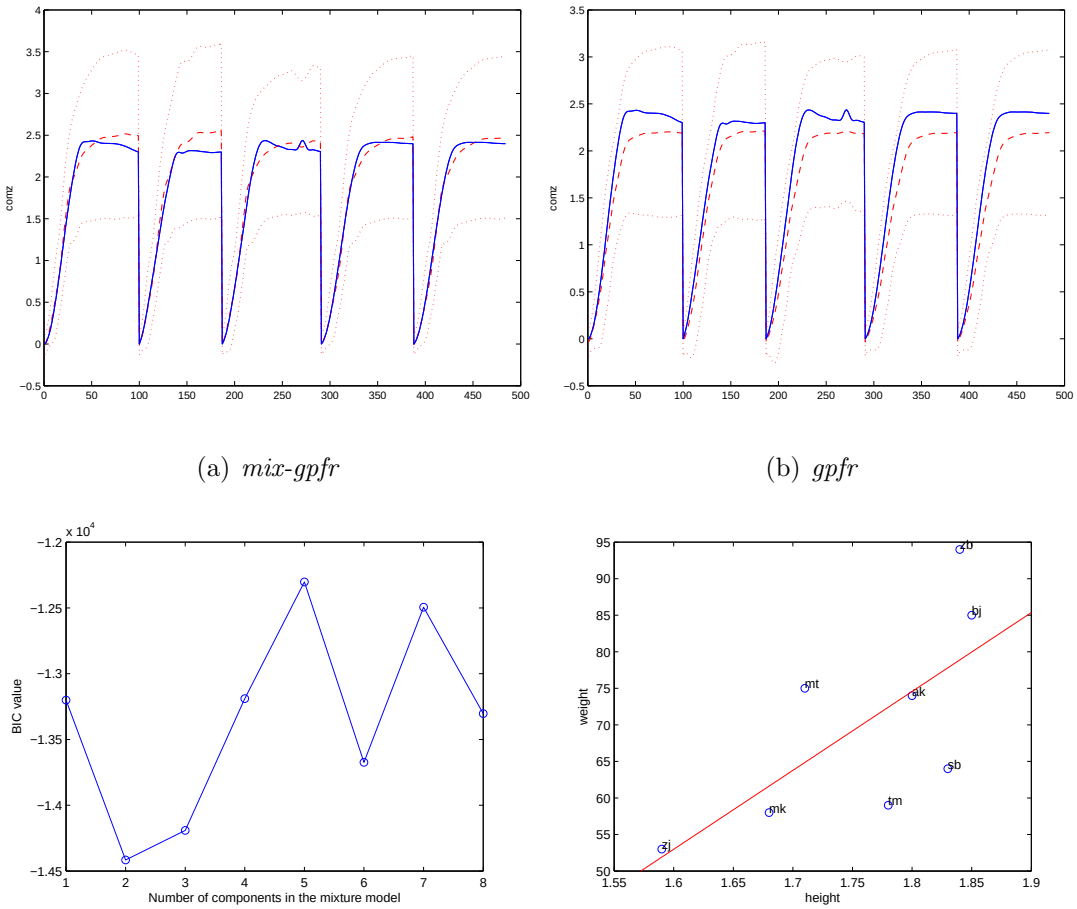
(a) Two mean curves (b) 60 sample curves

Figure 4: The mean curves and sample curves of a mixture model with two components




(a) 90 sample curves (b) BIC

Figure 5: The sample curves with three components and the values of BIC




(a) *mix-gpfr* (b) *gpfr*

(c) (d)

Figure 6: Paraplegia data. (a)-(b) The predictions for patient 'mk' by using a *mix-gpfr* and a single GPFR model: solid line is the real observation, the dashed line is the prediction and the dotted lines are 95% confidence bands. (c) The values of BIC. (d) Height and weight for each patient.