Gaussian Process Functional Regression Modelling for Batch Data

Shi, J. Q. [1] and Wang, B.

School of Mathematics & Statistics, University of Newcastle, UK

Murray-Smith, R. and Titterington, D. M.

Department of Computing Science and Department of Statistics,
University of Glasgow, UK

April 20, 2006

Abstract

A Gaussian process functional regression model is proposed for the analysis of batch data. Covariance structure and mean structure are considered simultaneously, with the covariance structure modelled by a Gaussian process regression model and the mean structure modelled by a functional regression model. The model allows the inclusion of covariates in both the covariance structure and the mean structure. It models the nonlinear relationship between a functional output variable and a set of functional and non-functional covariates. Several applications and simulation studies are reported and show that the method provides very good results for curve fitting and prediction.

Key words: Batch data, B-spline, Functional data analysis, Gaussian process regression model, Gaussian process functional regression model, Multiple-step-ahead forecasting, Non-parametric curve fitting

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

1 Introduction

We begin by discussing a motivating example. The application concerns data collected during standing-up manoeuvres of paraplegic patients. The *outputs* are the trajectories of the body centre of mass (COM), required for a simulator control system; for details see Section 3.3. However, it is very difficult to measure the body position unless some expensive equipment is used and set up in a laboratory environment. Thus, one of the aims of the project is to develop a model for reconstructing the trajectory of the body COM by using some easily measured quantities, such as the forces and torques under the patient's feet, under the arm support and under the seat while the body is in contact with it. More than thirty such *input* variables are observed in the project. Figure 1 depicts 40 curves of $y(t)$, the vertical trajectories of the body COM, denoted by *comz*. Each curve represents a standing-up manoeuvre. Eight patients participated in the experiment, and each of them repeated the experiment five times. Our aim is to find a 'model' f to model and predict $y(t)$ given covariates $\boldsymbol{x}(t) = (x_1(t), \ldots, x_Q(t))'$:

$$y(t) = f(t, x_1(t), \ldots, x_Q(t)) + \epsilon(t). \tag{1}$$

In each standing-up, both the output variable $y(t)$ and the input variables $\{x_1(t), \ldots, x_Q(t)\}$ are observed at a few hundred time points t_i, for $i = 1, \ldots, N$. If we observe M replications for a given individual, the m-th replication is called the m-th batch ($m = 1, \ldots, M$). The data collected are called batch data, terminology that is popular in the engineering community.

We have little information about the physical relationship between the output variable $y(t)$ and the input variables $\{x_1(t), \ldots, x_Q(t)\}$. For such data, it is typically not realistic to use parametric models, or nonparametric linear models such as the functional linear regression models discussed in Ramsay and Silverman (1997) and the varying-coefficient linear models described in Fan *et al.* (2003). Our idea is to treat the output curve, corresponding to a batch, as a stochastic process and then estimate the mean and covariance structure simultaneously:

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

where $\mu_m(t) = \mathrm{E}(y_m(t))$ and $\tau_m(\boldsymbol{x})$ is a stochastic process with zero mean and covariance kernel function $C(\boldsymbol{x}, \boldsymbol{x}^{'})$ with $\boldsymbol{x} = \boldsymbol{x}(t)$.

To be specific, we will use a Gaussian process regression (GPR) model for $\tau_m(\boldsymbol{x})$ in (2). The GPR model has been widely used as a nonparametric approach; see for example O'Hagan (1978), Williams (1998), Neal (1999) and Rasmussen and Williams (2006). In the model, $\tau = f(\boldsymbol{x})$ is an output curve, where $f(\cdot)$ is a function mapping an input $\boldsymbol{x} \in \mathcal{R}^q$ into $\tau \in \mathcal{R}$, and a Gaussian process prior is assumed for $f(\cdot) : f(\boldsymbol{x})$ has a normal distribution with zero mean and covariance kernel $C(\boldsymbol{x}, \boldsymbol{x}')$. The GPR approach models the nonlinear relationship between $y(t)$ and $\boldsymbol{x}(t)$ through a Gaussian process. Since we assume a zero mean or a given mean for the GPR model, we need first to standardize the data. This is straightforward for a single batch of data, which can be standardized by simply subtracting the sample mean. However, standardization is intractable for batch data, especially when it is used to generate predictions for a completely new batch; see Section 2.2. In Section 3.3, we will show that for the paraplegia example the performance of the GPR model is very good if we use training data collected from a particular patient to predict a new standing-up for the same patient; different standings-up for the same patient have similar mean structures. However, the performance of predictions for a new patient is not so good; mean structures of batches from different patients are quite variable (see Figure 1), because of differences in height, weight and other factors for the different patients. Another limitation of the GPR model is that, although it performs very well in interpolating test data, its extrapolation performance deteriorates very rapidly when the test data are 'distant' from the training data. A typical example is that of multiple-step-ahead forecasting. For k-step-ahead forecasting, GPR performs very well when k is small; see for example Girard and Murray-Smith (2005). However, when k is large, GPR usually fails in prediction; see Section 3.2. In this paper, we try to address these problems and improve the performance of the GPR model by considering the mean structure and covariance structure simultaneously.

To model the mean structure we will use a functional regression model (Ramsay and Silverman, 1997) with functional coefficient parameters and non-functional covariates (i.e. we use some batch-based information). We will show that using this type of mean structure within a GPR model can improve the performance substantially.

The idea of modelling the mean and covariance structure simultaneously has been reported elsewhere. For example, Rice and Silverman (1991) estimated the mean function using a cubic spline and estimated the covariance structure via smooth nonparametric estimates of the relevant eigen-functions. However, their method is limited to the case of a one-dimensional input variable. Eubank (2003) dealt with a forecasting problem by using a linear functional regression (LFR) model with a special covariance structure.

The paper is organised as follows. Section 2 proposes the Gaussian process functional regression model, and shows how to estimate unknown parameters and predict new test data. Several applications are reported in Section 3. Some discussion and further developments are given in Section 4.

2 The Gaussian process regression model with functional mean structure

2.1 The Gaussian process regression model for batch data

The discrete form of a Gaussian process regression model for a single batch of N observations is defined as

$$\boldsymbol{y} = (y_1, \dots, y_N)' \;\sim\; N(\boldsymbol{0}, \boldsymbol{C}), \tag{3}$$

where $\boldsymbol{C}$ is an $N \times N$ covariance matrix, of which the ij-th element $\boldsymbol{C}_{ij} = C(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a function of input covariates $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. An example of such a covariance function is

$$\begin{aligned} C(\boldsymbol{x}_i, \boldsymbol{x}_j) &= C(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) \\ &= v_0 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{iq} - x_{jq})^2\right) + a_0 + a_1 \sum_{q=1}^{Q} x_{iq} x_{jq} + \sigma_0 \delta_{ij}, \end{aligned} \tag{4}$$

where $\boldsymbol{\theta} = (w_1, \dots, w_Q, v_0, a_0, a_1, \sigma_0^2)$ denotes the set of unknown parameters, and δ_{ij} is the Kronecker delta. Some other types of covariance function and the related problems are discussed in MacKay (1999). Models (3) and (4) define a nonlinear model for y, given $\boldsymbol{x} = (x_1, \dots, x_Q)$; some recent developments can be found in Rasmussen and Williams (2006).

To deal with a data-set based on repeated experiments involving similar objects and processes, i.e. consisting of several batches, Shi *et al.* (2005) proposed a hierarchical Gaussian process regression model. Suppose that there are M different batches of data and that, in the mth batch, N_m observations are recorded. The data collected in the

m-th batch are

$$\mathcal{D}_m = \{(y_{mi}, t_{mi}, x_{m,1,i}, \ldots, x_{m,Q,i}) \text{ for } i = 1, \ldots, N_m; \text{ and } (u_{m1}, \ldots, u_{mp})\}, \quad (5)$$

where t_{mi} is the time point at which we record the data, $y_{mi} = y(t_{mi})$ is the output, for example the body *comz* position, recorded at t_{mi} and $x_{m,q,i} = x_q(t_{mi})$ is the measurement of the q-th input variable for $q = 1, \ldots, Q$. The elements of $\boldsymbol{u}_m = (u_{m1}, \ldots, u_{mp})'$ are not functional data, i.e. do not depend on t; they offer information for each curve, such as the patient's height and weight and the technique used in a particular standing-up. In Shi *et al.*'s hierarchical mixture model we have that

$$\boldsymbol{y}_m | z_m = k \ \sim \ GP_k(\boldsymbol{\theta}), \quad (6)$$

where $\boldsymbol{y}_m = (y_{m1}, \ldots, y_{mN_m})'$, and z_m is an unobservable latent indicator variable corresponding to the m-th batch. If $z_m = k$ is given, the model for the N_m correlated observations in batch m is a Gaussian process regression model $GP_k(\boldsymbol{\theta})$; that is, the process has a normal distribution with zero mean or a given mean μ, and a covariance kernel function $C_k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta})$. The association among the different batches is introduced by the latent variable z_m, for which

$$P(z_m = k) = \pi_k, \quad k = 1, \ldots, K, \quad (7)$$

for specified K. This model is used to address the problem of heterogeneity but without using the batch-based information $\boldsymbol{u}_m$. A special case is the model (6) with $K = 1$:

$$\boldsymbol{y}_m \ \sim \ GP(\boldsymbol{\theta}), \quad (8)$$

independently, for $m = 1, \ldots, M$. In this model, the heterogeneity among the different batches is ignored.

2.2 Gaussian process functional regression model

As already mentioned, the Gaussian process defined in (6) is usually assumed to have zero mean or a known mean μ. To apply it to batch data, we need to centre the data by subtracting its sample batch mean, calculated from the training data for each batch. However, this method has its limitations, as mentioned in Section 1. There are two main problems. First, if the training data are not distributed uniformly over the whole time range, if for example only the data corresponding to the first half of a standing-up were recorded, the batch sample mean calculated from such a data-set would be quite different from the real mean. Secondly, it is difficult to use this model to predict a new batch for which there are no training data, in particular for a new patient.

Instead, we model the mean structure as well as the covariance structure. We use a functional regression model to model the mean structure and a Gaussian process to model the covariance structure; that is, we use the model given in (2), namely $y_m(t) = \mu_m(t) + \tau_m(\boldsymbol{x})$ for $m = 1, \ldots, M$, where $\tau_m(\boldsymbol{x})$ is a zero-mean Gaussian process with covariance function $C(\boldsymbol{x}, \boldsymbol{x}')$, as defined in Section 2.1.

There is a large literature on fitting nonparametric curve $\mu_m(t)$ with a scalar input variable t. To model the dependence of the mean functions on the batch-based covariates, $\boldsymbol{u}_m$, we use a similar idea to that discussed in Ramsay and Silverman (1997) and take

$$\mu_m(t) = \boldsymbol{u}_m{}'\boldsymbol{\beta}(t), \tag{9}$$

where the coefficient $\boldsymbol{\beta}$ is functional.

If we have observed data in the dorm of $\mathcal{D}_m$ in (5) for the m-th batch, it is natural to consider a discrete form of the above model, given by

$$y_{mi} = \mu_{mi} + \tau_{mi}; \quad m = 1, \ldots, M, \quad i = 1, \ldots, N_m, \tag{10}$$

where $\mu_{mi} = \mu_m(t_{mi}) = \boldsymbol{u}_m{}'\boldsymbol{\beta}(t_{mi})$ and $\boldsymbol{\tau}_m = (\tau_{m1}, \ldots, \tau_{mN_m})'$ is defined in (6) or its special case (8). We will refer to this model as the Gaussian process functional regression (GPFR) model.

This model is quite distinct from the linear functional regression (LFR) model discussed in Ramsay and Silverman (1997). This is illustrated by the example given in Figure 2(a), in which the solid line represents the true mean curve whereas the dotted line represents the curve with a random error, fluctuating around the true mean curve. The dashed line represents a curve with dependent errors, which is a Gaussian process depending on a functional input variable x; for details see Section 3.1. This curve is systematically different from the true mean curve; thirty such sample curves are presented in Figure 2(b).

2.3 Estimation

It is not difficult to write down a penalized likelihood for model (10), and then estimate the unknown parameters involved in both the mean structure and the covariance structure. However, the implementation is tedious and there may be computational problems. We use a two-stage approach in this paper. In the first stage we estimate the mean structure by using B-spline smoothing; see for example Rice and Silverman (1991), Faraway (1997, 2001) and Ramsay and Silverman (1997). We approximate each curve $y_m(t)$ $(m = 1, \ldots, M)$ by

$$y_m(t) = \boldsymbol{A}_m\boldsymbol{\Phi}(t),$$

where $\boldsymbol{\Phi}(t) = (\boldsymbol{\Phi}_1(t), \ldots, \boldsymbol{\Phi}_K(t))'$ are the B-spline basis functions. The coefficients $\boldsymbol{A} = (\boldsymbol{A}_{mk})$ form an $M \times K$ matrix, whose elements are chosen to minimise

$$\int \left(y_m(t) - \sum_{k=1}^{K} \boldsymbol{A}_{mk} \boldsymbol{\Phi}_k(t) \right)^2 dt.$$

The functional parameters can be expanded as

$$\boldsymbol{\beta}(t) = \boldsymbol{B}\boldsymbol{\Phi}(t),$$

where $\boldsymbol{B}$ is a $p \times K$ matrix, which can be estimated by

$$\hat{\boldsymbol{B}} = (\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{U}'\boldsymbol{A},$$

where $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M)'$ is an $M \times p$ matrix. An estimate of the mean function is given by

$$\hat{\mu}_m(t) = \boldsymbol{u}_m{}'\hat{\boldsymbol{B}}\boldsymbol{\Phi}(t). \tag{11}$$

Selection of the form and number of the basis functions is discussed by Faraway (1997, 2001). From our empirical experience, the accuracy of the model depends mainly on successful modeling of the covariance structure when there is heterogeneity among different batches and the output curves depend on many input functional covariates. Thus, we do not need to use many basis functions in this first stage. For the examples discussed in the next section, we found that 20 basis functions were enough.

In the second stage, we replace μ_{mi} in equation (10) by its estimate $\hat{u}_{mi} = \hat{\mu}_m(t_{mi})$ from (11). Thus,

$$\hat{\tau}_{mi} = y_{mi} - \hat{\mu}_{mi}$$

is modelled by a Gaussian process as defined in (6) or (8) with covariance function $C(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta})$. A standard method such as a maximum likelihood or a MCMC approach can be used to estimate the unknown parameters $\boldsymbol{\theta}$, leading to $\hat{\boldsymbol{\theta}}$ say; see for example Rasmussen (1996) and Williams (1998). Implementation in the case of batch data is described by Shi *et al.* (2005).

2.4 Prediction

We now consider the generation of a prediction y^* at a new test point $(t^*, \boldsymbol{x}^*)$ with $\boldsymbol{x}^* = \boldsymbol{x}(t^*)$. From (2), $\hat{y}^* = \hat{\mu}(t^*) + \hat{\tau}(\boldsymbol{x}^*)$, where $\tau^* = \tau(\boldsymbol{x}^*)$ is predicted by its conditional mean $\mathrm{E}(\tau^*|\mathcal{D})$ through the Gaussian process with estimated covariance function $C(\boldsymbol{x}, \boldsymbol{x}^{'}; \hat{\boldsymbol{\theta}})$. We will discuss two types of prediction in this subsection. First we suppose that we have already observed some training data in a batch, the $(M+1)$-th batch say, and want to predict the output for a new set of inputs. In addition to the training data observed in the first M batches, we assume that N observations are obtained in the new batch, providing data

$$\mathcal{D}_{M+1} = \{(y_{M+1,i}, t_{M+1,i}, x_{M+1,1,i}, \ldots, x_{M+1,Q,i}) \text{ for } i = 1, \ldots, N; \text{and } \boldsymbol{u}_{M+1}\}.$$

We therefore have training data $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathcal{D}_{M+1}\}$. 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 observed test inputs. We use the Gaussian process assumption for the $(M+1)$-th batch,

$$(\tau_{M+1,1}, \ldots, \tau_{M+1,N}, \tau^*) \sim N(0,\ \boldsymbol{\Omega}) \tag{12}$$

where $\tau = y - \mu$ and $\boldsymbol{\Omega}$ is an $(N+1) \times (N+1)$ covariance matrix

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{C}(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) \\ \boldsymbol{C}'(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) & C(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix} \tag{13}$$

where $\boldsymbol{C}(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) = (C(\boldsymbol{x}^*, \boldsymbol{x}_{M+1,1}), \ldots, C(\boldsymbol{x}^*, \boldsymbol{x}_{M+1,N}))'$, is the covariance matrix between y^* and $\boldsymbol{y}_{M+1} = (y_{M+1,1}, \ldots, y_{M+1,N})'$, and $\boldsymbol{C}$ is the $N \times N$ covariance matrix of $\boldsymbol{y}_{M+1}$ or $\boldsymbol{\tau}_{M+1}$, which depends on $\boldsymbol{x}_{M+1}$. The covariance can be calculated by, for example, (4) with the parameters estimated as in the previous subsection. Thus, the predictive distribution of y^*, given training data $\mathcal{D}$ and the mean $\boldsymbol{\mu}(t)$, is also a Gaussian distribution. Its mean and variance are given by

$$\begin{aligned} \mathrm{E}(y^*|\mathcal{D},\boldsymbol{\mu}) &= \mu_{M+1}(t^*) + \boldsymbol{H}'(\boldsymbol{y}_{M+1} - \boldsymbol{\mu}_{M+1}(\boldsymbol{t})), & (14) \\ \sigma_{GP}^{*2} = \mathrm{Var}(y^*|\mathcal{D},\boldsymbol{\mu}) &= C(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{H}'\boldsymbol{C}\boldsymbol{H}, & (15) \end{aligned}$$

where $\boldsymbol{\mu}_{M+1}(\boldsymbol{t}) = (\mu_{M+1}(t_1),\ldots,\mu_{M+1}(t_N))'$ is the vector of means at data points $\boldsymbol{t} = (t_1,\ldots,t_N)$, and $\boldsymbol{H}' = [\boldsymbol{C}(\boldsymbol{x}^*,\boldsymbol{x}_{M+1})]'\boldsymbol{C}^{-1}$. Thus the prediction for y^* is given by

$$\hat{y}_{M+1}^* = \hat{\mu}_{M+1}(t^*) + \boldsymbol{H}'(\boldsymbol{y}_{M+1} - \hat{\boldsymbol{\mu}}_{M+1}(\boldsymbol{t})), \qquad (16)$$

where $\hat{\mu}_{M+1}(t^*)$ and $\hat{\boldsymbol{\mu}}_{M+1}(\boldsymbol{t})$ are given by (11), i.e., $\hat{\mu}_{M+1}(\cdot) = \boldsymbol{u}_{M+1}'\hat{\boldsymbol{B}}\boldsymbol{\Phi}(\cdot)$. The prediction variance can be calculated from the conditional variances in equations (14) and (15):

$$\begin{aligned} \hat{\sigma}_{M+1}^{*2} = \mathrm{Var}(y^*|\mathcal{D}) &= \mathrm{E}[\mathrm{Var}(y^*|\mathcal{D},\boldsymbol{\mu})] + \mathrm{Var}[\mathrm{E}(y^*|\mathcal{D},\boldsymbol{\mu})] \\ &= \hat{\sigma}_{GP}^{*2}\left(1 + \boldsymbol{u}_{M+1}'(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{u}_{M+1}\right), & (17) \end{aligned}$$

where $\hat{\sigma}_{GP}^{*2}$ is given by (15), in which all the parameters are replaced by their estimators. The derivation of equation (17) is given in Appendix I.

The second type of prediction is to predict for a completely new batch. We shall still refer to the new batch as the $(M+1)$-th batch, with batch-based covariate $\boldsymbol{u}_{M+1}$. We want to predict y^* at $(t^*, \boldsymbol{x}^*)$. In this case, the training data are $\mathcal{D} = \{\mathcal{D}_1,\ldots,\mathcal{D}_M\}$. Since we have not observed any data in the $(M+1)$-th batch, we cannot use the predictive mean and covariance matrix discussed above, which is based on the Gaussian process assumption (12). The argument in Shi *et al.* (2005) is that batches $1,\ldots,M$

provide an empirical distribution of the set of all possible batches. We will use a similar idea here but only for the Gaussian process component $\tau_{M+1}(\cdot) = y_{M+1}(\cdot) - \hat{\mu}_{M+1}(\cdot)$. We assume that, for $m = 1, \ldots, M$,

$$P(\tau^* \text{ belongs to } m\text{-th batch}) = 1/M.$$

To say that τ^* belongs to the m-th batch means that

$$(\tau_{m,1}, \ldots, \tau_{m,N_m}, \tau^*) \sim N(0, \boldsymbol{\Omega}_m),$$

where $\boldsymbol{\Omega}_m$ is an $(N_m + 1) \times (N_m + 1)$ covariance matrix which is defined similarly to (13). We can therefore calculate $\hat{y}^*_m$ and $\hat{\sigma}^{*2}_m$ from (16) and (17) respectively, as if the test data belongs to the mth batch. Since the empirical distribution is relevant to the Gaussian process component only, $\hat{y}^*_m$ is given by

$$\hat{y}^*_m = \hat{\mu}_{M+1}(t^*) + \boldsymbol{H}^{'}(\boldsymbol{y}_m - \hat{\boldsymbol{\mu}}_m(\boldsymbol{t})). \tag{18}$$

The value of $\hat{\sigma}^{*2}_m$ is given by (17) and (15), but the related covariance matrices are calculated at $\boldsymbol{x}^*$ and $(\boldsymbol{x}_{m,1}, \ldots, \boldsymbol{x}_{m,N_m})$.

Based on the above empirical assumption, the prediction for the response associated with a test input $\boldsymbol{x}^*$ at t^* in a completely new batch is

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

and the predictive variance is

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

3 Applications

3.1 Learning and prediction for large batch data-sets

We first consider an example with simulated data. The true model used to generate the data is $y_{mi}(x) = u_m + \sin(0.5x_{mi})^3 + \tau_{mi}$, where, for each m, $x_{mi} \in (-4, 4)$ and $\{\tau_{mi}\}$ is a Gaussian process with zero mean and covariance function

$$C(x_{mi}, x_{mj}) = v_0 \exp\Big(-\frac{1}{2}w_0(x_{mi} - x_{mj})^2\Big) + \sigma_0\delta_{ij},$$

with $v_0 = 0.1$, $w_0 = 1.0$ and $\sigma_0 = 0.0025$. In this example, x_{mi} is the same as t_{mi}. Thirty independent curves are generated and are presented in Figure 2(b), where $u_m = 0$ for batches 1 to 10, $u_m = -1$ for batches 11 to 20, and $u_m = 1$ for the remaining 10 batches. In each batch, 100 data points are generated. We randomly select half of the data points as training data.

In this example, only one discrete covariate u_m is used in the mean structure model (9). The model is such that $\mu_m(t) = \beta_1(t)$ when $u_m = -1$, $\mu_m(t) = \beta_2(t)$ when $u_m = 0$, and $\mu_m(t) = \beta_3(t)$ when $u_m = 1$. We use B-spline smoothing and the method discussed in Section 2.3 to estimate $\beta_i(t)$, for $i = 1, 2, 3$. We then apply a GPR model to $\hat{\tau}_m(t) = y_m(t) - \hat{\mu}_m(t)$ with covariance function (4), depending on the scalar input variable x as discussed in Section 2.3. Prediction can then be carried out by the method discussed in Section 2.4.

To assess the performance of the method, we calculate predictions for a new batch and compare them with the actual output values of the test data. One hundred data points are generated in a new batch with $u_m = 1$. Half of the data are selected as part of the training data and are used to predict the rest. Two types of training data are considered. Type I data are selected randomly from the whole set of 100 points. In Type II data, data points corresponding to $x \in (-4, 0)$ are used as training data and the rest are used as test data. Thus, Type I prediction is essentially interpolation, while Type II prediction involves extrapolation, which is obviously harder. For comparison, GPR and LFR are also applied to the same data-set.

The predictions obtained by the different methods, along with the actual values of the test data, are plotted in Figure 3 and presented in Table 1, in which *rmse* is the root mean squared error between the predictions and the true test values, and r is the related correlation coefficient. The left-hand panels in Figure 3 report the results for Type I prediction, i.e., interpolation. Both GPFR and GPR predict the output very precisely. It is not surprising that the result using LFR is less good, since LFR models the common mean structure only. The results for Type II prediction are given in the right-hand panels in Figure 3. GPFR still gives very precise predictions. LFR predicts the common mean structure, so that its performance is quite similar for both types of test data. GPR fails to predict this type of test data well except for a few points close to 0. GPR models the covariance structure only, and does not reliably predict the test data if they are 'far away' from the training data. When test data are 'close' to training data, the GPFR model uses both the mean structure and the covariance structure to calculate predictions, and gives a very precise result; when test data are distant from the training data, GPFR would rely mainly on the mean structure to predict test data and could still give a reasonable good result. This is confirmed by the simulation study, based on 50 replications, reported in Table 1. The overall performance of GPFR for Type II prediction is better than LFR and GPR. In the range [0,1], which is close to the training data, GPFR ($rmse = 0.1321$) is much better than LFR ($rmse = 0.2847$) and

GPR ($rmse = 0.2271$). In the range [1,4], which is further away from the training data, GPFR ($rmse = 0.3116$) is slightly better than LFR ($rmse = 0.3352$), as expected, and GPR fails in the prediction of output in this range ($rmse = 0.6843$, see also Figure 3(f)).

3.2 Multiple-step-ahead forecasting

If, in a discrete-time dynamic system, we have available data up to time t, we may wish to predict the output at time $t + k$. This is called k-step-ahead forecasting. Many dynamic systems are periodic in real-life problems, and data collected in each period form a batch. We may refer to the data collected in previous periods as historical data. In the current period, we use the data up to time t as well as the historical data to train the model and forecast the output at time $t + k$.

We use the artificial model used in Section 3.1 to generate a set of batch data. Values for 30 curves are generated, each at 50 equally-spaced points. For the k-step-ahead forecasting problem, we assume that there are two functional input variables associated with each output y_i, namely x_i and y_{i-k}. We first use the data for those 30 curves and the related input variables to train the model, and then do k-step-ahead forecasting after one third of the data in a new batch has been observed. For comparison, the GPFR, LFR and GPR methods are applied to the same data-set. A simulation study with 50 replications was conducted. The results from the simulation study and the result from one typical replication, for $1-$, $3-$ and 6-step-ahead forecasts, are reported in Table 2. The results for the single replication are also presented in Figure 4. The values of $rmse$ and r are calculated by comparing the forecast values with the actual simulated values. The GPFR model performs very well in all the cases and becomes more and more accurate as the end of the period is approached, by which time more observations have been obtained. The LFR model gives reasonably good predictions in all the cases. The precision achieved does not vary much with k. As expect, LFR does not give very accurate predictions since only the common mean structure is modelled. When k is small, GPR gives good predictions; see for example the 1-step-ahead results reported in Table 2. However, the precision decreases very rapidly as k increases. It fails when k is larger than six in this example. Of course, the precision of the forecasting obtained by GPFR also decreases as k increases, but the rate of decrease is much slower than for the GPR model. When k is very large, the forecast obtained by GPFR will be close to that given by LFR, i.e. it will give a forecast based on a common mean structure. This is still a useful forecast.

3.3 Modelling of standing-up manoeuvres by paraplegic patients

Our application, as introduced in Section 1, involves the analysis of the standing-up manoeuvre in the context of paraplegia; for operational details see Kamnik *et al.* (1999, 2005). Here we model the vertical trajectory of the body COM as output, and select 14 input variables. In one standing-up, output and inputs are recorded for a few hundred time points. The experiment was repeated several times for each patient. The vertical trajectories of the body COM are presented in Figure 1 for 40 standings-up, 5 for each of 8 patients.

For modeling the mean structure in the GPFR model, we use three covariates for $\boldsymbol{u}_m$, namely the patient's height, weight and sex. These are natural covariates for the mean structure model. Figure 1 indicates that the time scale may be different for different standings-up. Some registration methods are used before estimating the mean structure; see Ramsay and Silverman (1997) and Ramsay and Li (1998). We then estimate the mean structure and covariance structure by the method discussed in Section 2 with the covariance function (4), and use this model to predict a new standing-up. For comparison, results from GPR models are also obtained.

We consider two important types of prediction here. Type A uses the data which have already been observed for one patient to predict a new standing-up for the same patient, whereas Type B predicts a standing-up for a new patient. For Type A prediction, we use the training data from the same patient, which should have similar mean structure, so that GPR should predict well. However, it may be less satisfactory for Type B prediction. This is confirmed by the numerical results. Figures 5 (a) and (b) present the Type A prediction by GPFR and GPR respectively, both giving very good results. Figures 5 (c)-(f) show Type B predictions. Figures 5 (c) and (d) present the prediction for a single standing-up for a new patient by using the GPFR and GPR models respectively. The values of *rmse* are 18.6180 and 47.8120 respectively, which shows that GPFR performs much better than GPR. Figures 5 (e) and (f) show the predictions for all five standings-up for this new patient. The average value of *rmse* is 14.6661 for GPFR and 40.95550 for GPR, showing that GPFR performs consistently better than GPR.

We selected one of the eight patients as the new patient in turn, and used the data obtained from the rest as the training data. The results are very similar to those reported in Figure 5 except for one patient. However, that patient is atypical, being a thin (59kg) and very tall (178cm) woman. She is an outlier in terms of the mean structure model, and thus the GPFR model fails to improve much on the result obtaining by GPR. For this patient, the average value of *rmse* for five standings-up is 62.8 for the GPFR and 51.74 for the GPR model.

4 Discussion

Nonparametric and nonlinear regression analysis for batch data (functional data or longitudinal data) is a difficult problem with a large literature. However, most methods are limited to scenarios with one- or two-dimensional covariates (Lin and Carroll, 2000; Yao *et al.*, 2005), or the nonparametric linear regression model (Fan et al., 2003; Ramsay and Silverman, 1997). However, for our motivating example discussed in Section 1 and Section 3.3 and many other applications, one needs a nonparametric nonlinear model that copes with high-dimensional input functional covariates.

Rice and Silverman (1991) treat the output curve (batch) as a stochastic process and then estimate both the mean and covariance structure: $y_m(t) = \mu_m(t) + \tau_m(t)$, where $\mu_m(t) = \mathrm{E}(y_m(t))$ and $\tau_m(t)$ is a stochastic process with zero mean and kernel covariance function $C(t, t') = \mathrm{Cov}(y(t), y(t'))$. If there is an orthogonal expansion (in the L^2 sense) $C(t, t') = \sum_v \lambda_v \phi_v(t) \phi_v(t')$ in terms of eigen-functions $\phi_v(t)$ and non-increasing eigenvalues λ_v, $\tau_m(t)$ can be expanded as $\tau_m(t) = \sum_v \xi_{mv} \phi_v(t)$ and thus

$$y_m(t) = \mu_m(t) + \sum_v \xi_{mv} \phi_v(t), \tag{21}$$

where the ξ_{mv} are uncorrelated random variables with zero means and variances λ_v. The first term on the right-hand side of the above equation is the overall mean for the different curves (batches), and the second term describes the covariance structure for the different data points in the same curve. In our method the covariance kernel C for the second term is a function of input variables $\boldsymbol{x}$, i.e. $\mathrm{Cov}(y(t), y(t')) = C(\boldsymbol{x}, \boldsymbol{x}^{'})$, as in model (2). Similarly to (21), model (2) can be expanded as

$$y_m(t) = \mu_m(t) + \tau_m(\boldsymbol{x}) = \mu_m(t) + \sum_v \xi_{mv}\phi_v(\boldsymbol{x}), \tag{22}$$

where $\phi_v(\boldsymbol{x})$ are the eigen-functions of $C(\boldsymbol{x}, \boldsymbol{x}^{'})$. Thus, model (2) or (22) gives a nonparametric nonlinear regression model for $y(t)$ in terms of a set of functional covariates $\boldsymbol{x}(t) = (x_1(t), \ldots, x_Q(t))'$ and t also, through the eigen-functions $\{\phi_v(\boldsymbol{x})\}$. If we assume that $p(\boldsymbol{x})$ is the density function of the input vector $\boldsymbol{x}$, we have the following results, with the subscript m omitted. The covariance kernel function can be expanded as $C(\boldsymbol{x}, \boldsymbol{x}^{'}) = \sum_v \lambda_v \phi_v(\boldsymbol{x})\phi_v(\boldsymbol{x}^{'})$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ are the eigenvalues and $\phi_1, \phi_2, \ldots$ are the corresponding eigen-functions of the operator whose kernel is $C(\boldsymbol{x}, \boldsymbol{x}^{'})$, so that

$$\int C(\boldsymbol{x}, \boldsymbol{x}^{'})\phi_v(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \lambda_v \phi_v(\boldsymbol{x}^{'}).$$

The eigen-functions are p-orthogonal, and ξ_v is given by

$$\xi_v = \int \phi_v(\boldsymbol{x})\tau(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}.$$

Yao *et al.* (2005) approximated the covariance matrix $C(t, t')$ by a smooth surface estimate and then estimated ξ_{mv} and $\phi_v(t)$ in (21). It is very difficult to extend their method to model (22) which involves $C(\boldsymbol{x}, \boldsymbol{x}')$ when the dimension of $\boldsymbol{x}$ is larger than one.

For the problem of curve fitting and prediction with high-dimensional input variables, neural network models are another popular approach; see Cheng and Titterington (1994) for a review. Shi *et al.* (2005) and Kamnik *et al.* (2005) showed that the GPR model gives a better fit for the paraplegia data than did the neural network models. However, the GPR model models the covariance structure only.

In future work it would be worthwhile to explore a unified Bayesian approach to the GPFR model; see for example DiMatteo *et al.* (2001) and Shi *et al.* (2005).

Acknowledgements

Dr. Wang is grateful for support of the UK Engineering and Physical Sciences Research Council for grant EPSRC GR/T29253/01. 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. We are grateful to the Associate Editor and the reviewers for very helpful comments.

Appendix I. Derivation of equation (17)

From (14), we have

$$\begin{aligned}\mathrm{Var}[\mathrm{E}(y^*|\mathcal{D},\boldsymbol{\mu})] &= \mathrm{Var}[\mu_{M+1}(t^*)|\mathcal{D}] + \boldsymbol{H}'\mathrm{Var}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t})|\mathcal{D}]\boldsymbol{H} \\ &\quad -2\boldsymbol{H}'\mathrm{Cov}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t}),\ \mu_{M+1}(t^*)|\mathcal{D}]. \qquad (23)\end{aligned}$$

From the results in Section 2.3, the estimator of the functional mean at a single data point t is given by

$$\begin{aligned}\hat{\mu}_{M+1}(t) &= \boldsymbol{u}_{M+1}{}'\hat{\boldsymbol{B}}\boldsymbol{\Phi}(t) = \boldsymbol{u}_{M+1}{}'(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{U}'\boldsymbol{A}\boldsymbol{\Phi}(t) \\ &= \boldsymbol{u}_{M+1}{}'(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{U}'\boldsymbol{Y}(t), \qquad (24)\end{aligned}$$

where $\boldsymbol{Y}(t) = (y_1(t), \ldots, y_M(t))'$, and $y_m(t)$ is the output response variable at data point t in the m-th batch, for $m = 1, \ldots, M$. Since the data in different batches are independent, it follows that

$$\mathrm{Var}[\boldsymbol{Y}(t)] = \mathrm{Var}[(y_1(t), \ldots, y_M(t))'] = C(\boldsymbol{x}, \boldsymbol{x})\boldsymbol{I}_M,$$

where $\boldsymbol{I}_M$ is an $M \times M$ identity matrix, and $C(\boldsymbol{x}, \boldsymbol{x})$ is the variance of $y(t)$ at $\boldsymbol{x}(t) = \boldsymbol{x}$. Applying the above equation to (24) gives the variance of $\hat{\mu}_{M+1}(t)$ as

$$\mathrm{Var}[\hat{\mu}_{M+1}(t)|\mathcal{D}] = C(\boldsymbol{x}, \boldsymbol{x})\gamma, \quad \text{with } \ \gamma = \boldsymbol{u}_{M+1}{}'(\boldsymbol{U}'\boldsymbol{U})^{-1}\boldsymbol{u}_{M+1}. \tag{25}$$

Similarly, the covariance between the estimated means at two data points t and t^* is given by

$$\mathrm{Cov}[\hat{\mu}_{M+1}(t^*), \hat{\mu}_{M+1}(t)|\mathcal{D}] = C(\boldsymbol{x}^*, \boldsymbol{x})\gamma. \tag{26}$$

Applying equations (25) and (26) to data points $(t_1, \ldots, t_N)$ and t^* and using the notation around (14) and (15), we have

$$\begin{aligned} \boldsymbol{H}^{'}\mathrm{Var}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t})|\mathcal{D}]\boldsymbol{H} &= \gamma\boldsymbol{H}^{'}\boldsymbol{C}\boldsymbol{H}, \\ \boldsymbol{H}^{'}\mathrm{Cov}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t}),\ \mu_{M+1}(t^*)|\mathcal{D}] &= \gamma\boldsymbol{H}^{'}\boldsymbol{C}(\boldsymbol{x}^*, \boldsymbol{x}) = \gamma\boldsymbol{H}^{'}\boldsymbol{C}\boldsymbol{H}. \end{aligned}$$

Substituting the related items in (23) by the above equations results in (17).

REFERENCES

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

DiMatteo, I., Genovese, C. R. and Kass, R. E. (2001). Bayesian curve fitting with free-knot splines. *Biometrika* **88**, 1055-1071.

Eubank, R. L. (2003). Inference for (prediction using?) functional regression models. Department of Statistics, Texas A & M University. IMS Mini-Meeting on Functional Data Analysis. Florida.

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 Michigan.

Girard, A. and Murray-Smith, R. (2005). Gaussian Processes: Prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series. In *Switching and Learning in Feedback Systems*, R. Murray-Smith and R. Shorten (eds.), Lecture Notes in Computing Science, Vol. **3355**, 158-184, New York: Springer-Verlag.

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 modeling 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.

Lin, X. and Carroll, R. J. (2000). nonparametric function estimation for clustered data when the predictor is measured without/with error. *Journal of the American Statistical Association* **95**, 520-534.

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. (1999). Regression and classification using Gaussian process priors (with discussion). In *Bayesian Statistics 6*, J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (eds.), 475-501. Oxford University Press.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society, Series B* **40**, 1-42.

Ramsay, J. O. (1998). Principal differential analysis: Data reduction by differential operators. *Journal of the Royal Statistical Society, Series B* **58**, 495-508.

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

Ramsay, J. O. and Li, X. (1998) Curve registration. *Journal of the Royal Statistical Society, Series B* **60**, 351-363.

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

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning.* Cambridge, MA: The MIT Press.

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.

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

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. Cambridge, MA: The MIT Press.

Yao, F., Müller, H. G. and Wang, J. L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association* **100**, 577-590.

Table 1: The values of *rmse* and *r* between true and predicted responses

Model	Type I		Type II			
Results corresponding to Figure 3						
	rmse	r	rmse	r		
GPFR	0.0900	0.9917	0.1073	0.9894		
LFR	0.4570	0.7770	0.4150	0.9838		
GPR	0.0820	0.9934	0.6675	-0.0149		
Average values from simulation study						
	rmse	r	rmse[1]	r	rmse[2]	rmse[3]
GPFR	0.0588	0.9954	0.2802	0.9270	0.1321	0.3116
LFR	0.3244	0.9068	0.3318	0.9143	0.2874	0.3352
GPR	0.0830	0.9911	0.6044	0.1246	0.2271	0.6843

[1] The overall *rmse* comparing true and predicted responses in range [0,4]

[2] The *rmse* comparing true and predicted responses in range [0,1]

[3] The *rmse* comparing true and predicted responses in range [1,4]

Table 2: Numerical results of *rmse* and *r* for *k*-step-ahead forecasting

model	1 step		3 step		6 step	
	rmse	*r*	*rmse*	*r*	*rmse*	*r*
Results corresponding to Figure 4						
GPFR	0.0893	0.9911	0.1448	0.9766	0.3084	0.9057
FDA	0.3368	0.8660	0.3937	0.8351	0.4324	0.8122
GPR	0.1621	0.9723	0.4498	0.7568	0.6424	0.2706
Average values from simulation study						
GPFR	0.0785	0.9896	0.1229	0.9769	0.2169	0.9360
FDA	0.3013	0.8989	0.3097	0.8820	0.3242	0.8172
GPR	0.1487	0.9670	0.4375	0.6640	0.5994	0.0021

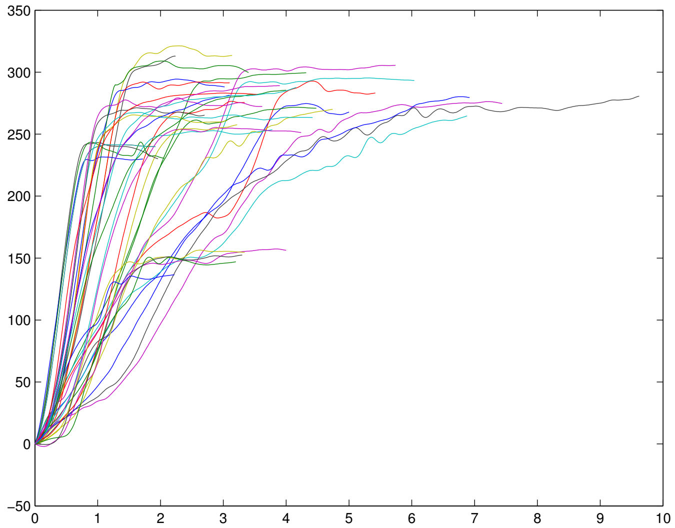


Figure 1: Paraplegia Data for 8 patients: vertical trajectory of the body COM *comz* coordinate (Y-axis, in mm) against time (X-axis, in seconds). Each curve corresponds to one standing-up.

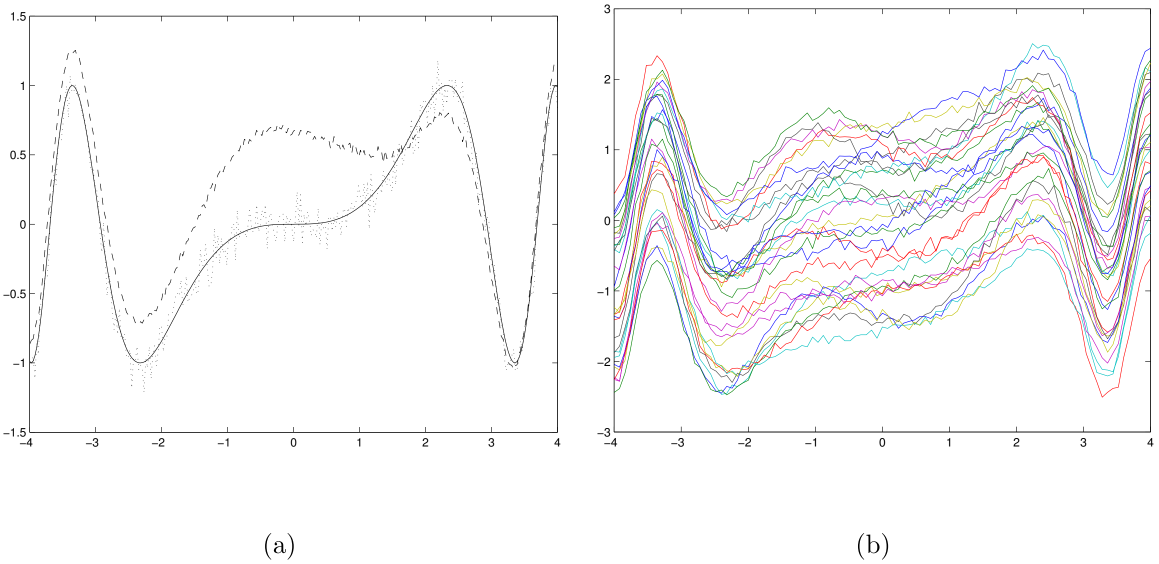
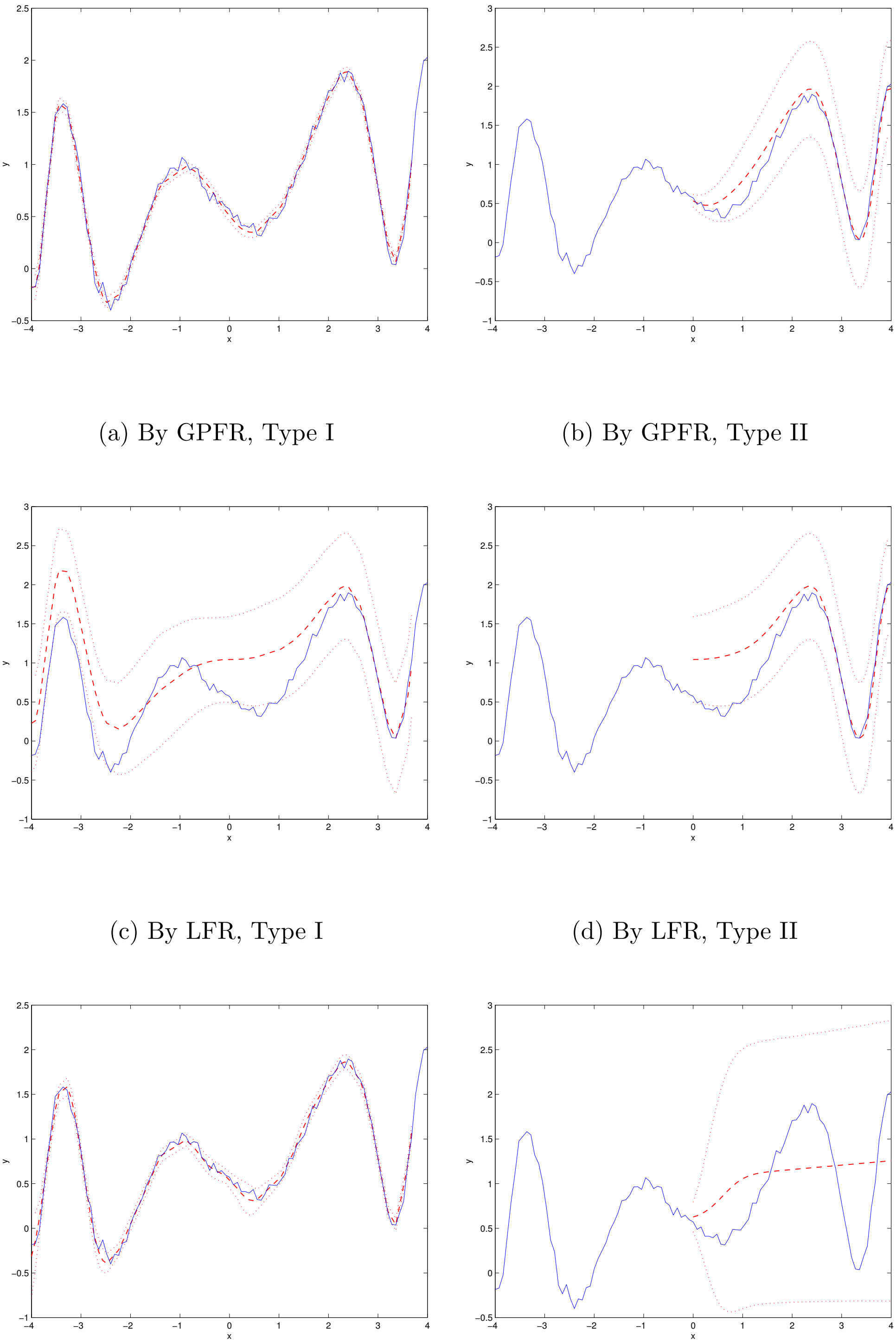


Figure 2: The sample curves. (a) Solid line—the true mean curve; dotted line—the curve with random errors; dashed line—the curve with errors having GP covariance structure depending on x. (b) 30 sample curves with GP errors.



(a) By GPFR, Type I

(b) By GPFR, Type II

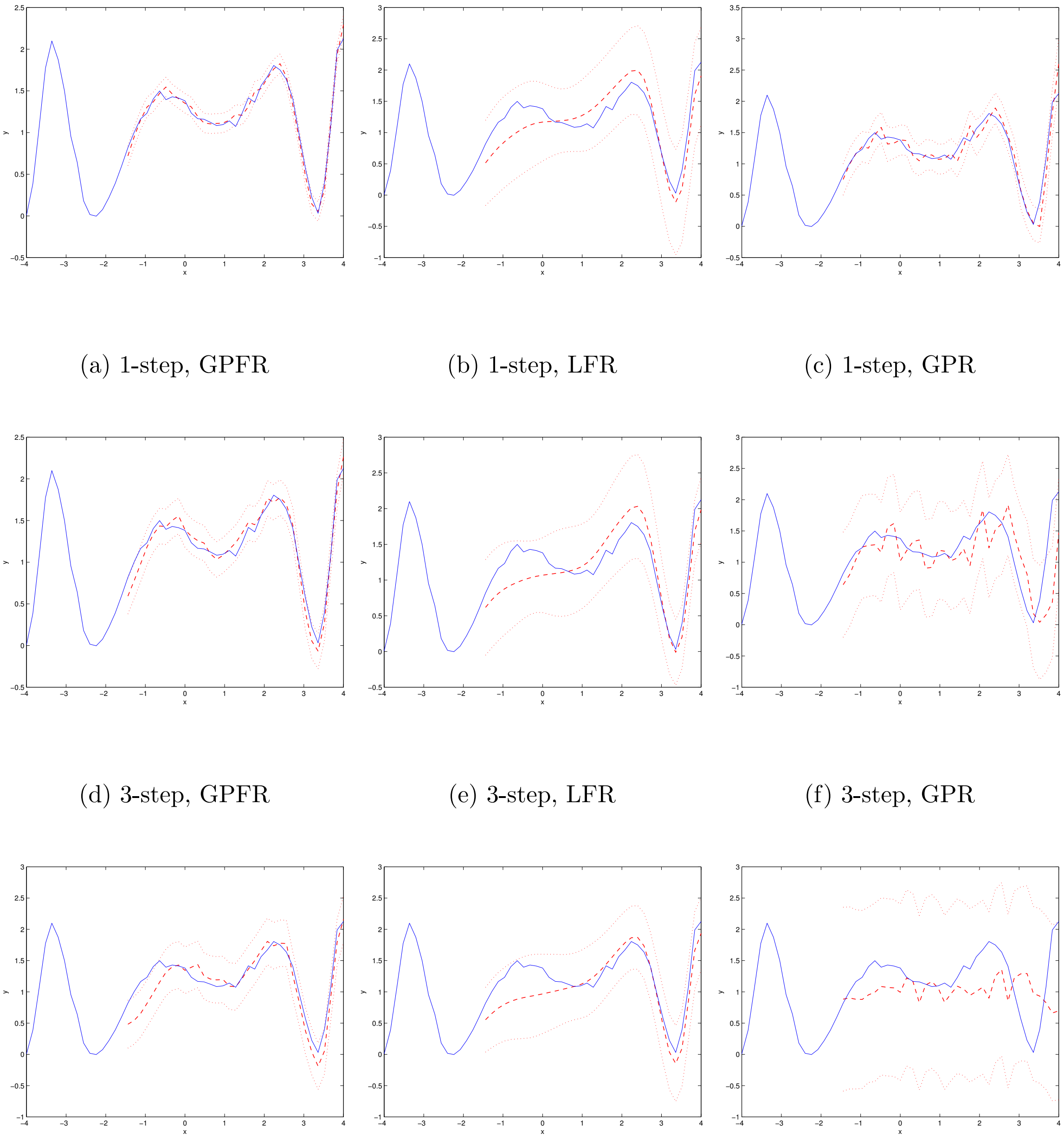
(c) By LFR, Type I

(d) By LFR, Type II

(e) By GPR, Type I

(f) By GPR, Type II

Figure 3: The predictions obtained by the different methods, where the solid line represents the true curve, the dashed line represents the predictions, and the dotted lines represent 95% prediction intervals.



(a) 1-step, GPFR

(b) 1-step, LFR

(c) 1-step, GPR

(d) 3-step, GPFR

(e) 3-step, LFR

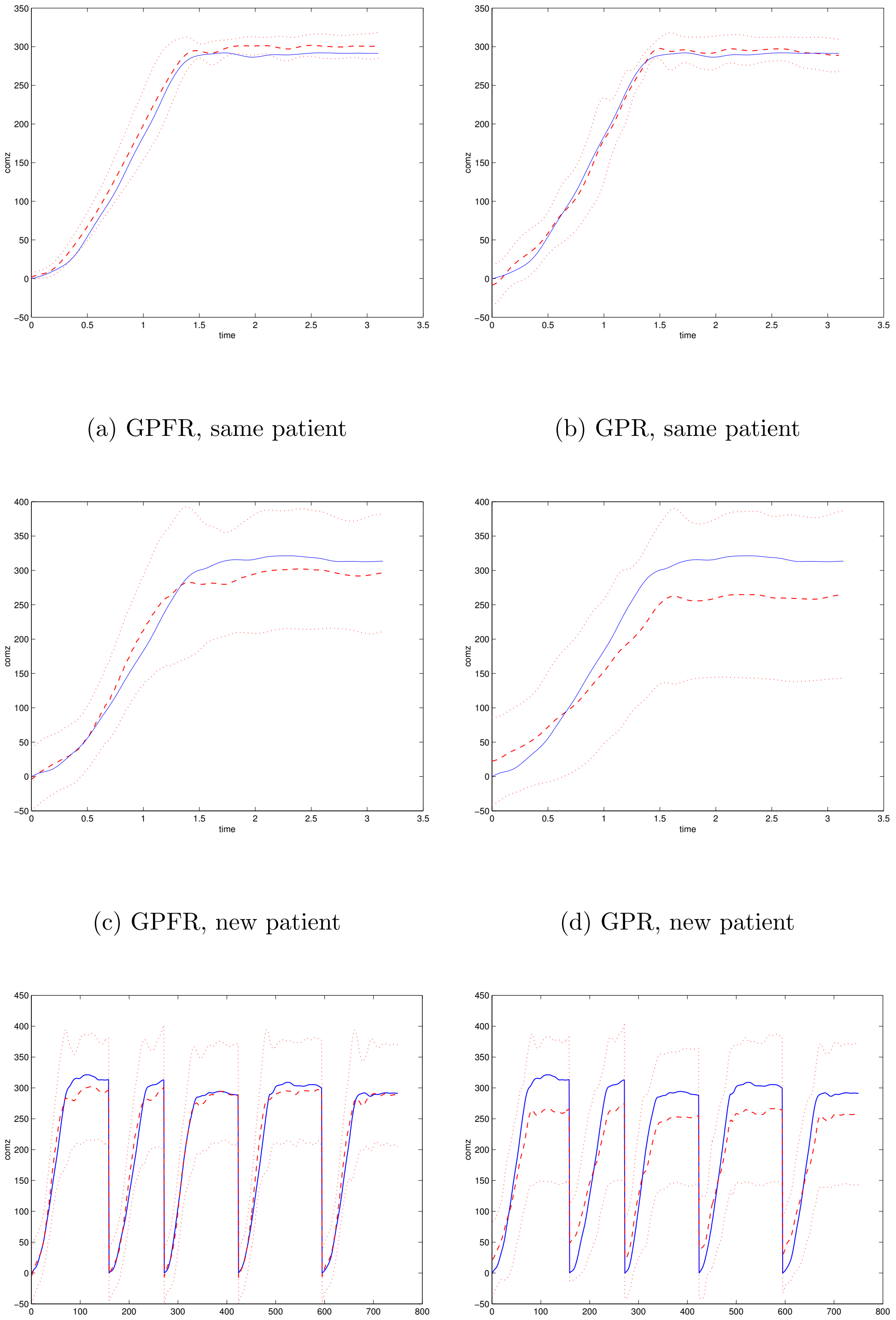
(f) 3-step, GPR

(g) 6-step, GPFR

(h) 6-step, LFR

(i) 6-step, GPR

Figure 4: The k-step-ahead forecasts and the associated 95% prediction intervals



(a) GPFR, same patient

(b) GPR, same patient

(c) GPFR, new patient

(d) GPR, new patient

(e) GPFR, new patient

(f) GPR, new patient

Figure 5: Paraplegia data: the true test data (solid line), the prediction (dashed line) and the 95% prediction intervals (dotted line). (a)-(b) Prediction of a new standing-up as for the same patient. (c)-(d) Prediction of a new standing-up as for a new patient. (e)-(f) Prediction of 5 new standings-up for a new patient.