Gaussian Process Functional Regression Model for Curve Prediction and Clustering

J.Q. SHI

School of Mathematics and Statistics, University of Newcastle, UK
j.q.shi@ncl.ac.uk
http://www.staff.ncl.ac.uk/j.q.shi

INI, Cambridge

04/03/2008
1 Introduction: two examples

2 Gaussian process regression (GPR) model

3 Gaussian process functional regression (GPFR) model

4 Hierarchical mixtures of GPFR models (mix-GPFR)
   - Curve clustering

5 Comments
Example 1: Dose-response study

- **Background:** Patient with renal failure need to take drug e.g. Darbepoetin Alpha (DA) to control haemoglobin (Hb) level in a certain range.

- **Objective:** how to determine a suitable level of dose and others to control Hb level.
Example 1: Dose-response study

- **Background:** Patient with renal failure need to take drug e.g. Darbepoetin Alpha (DA) to control haemoglobin (Hb) level in a certain range.
- **Objective:** how to determine a suitable level of dose and others to control Hb level.
- **Functional Response** $y(t)$: Hb level, measured at different time points.
- **Two types of covariates:**
  - **Functional covariates** $x(t)$: including e.g. $x_1(t)$–dose level; $x_2(t)$–time taking the drug; $x_3(t)$–iron dose.
  - **Subject based scalar covariates** $u$: including e.g. age, weights, gender.
Example 1: Dose-response study

- Modeling: how to find a functional regression model
  \[ y_m(t) = f_m(x(t), u) + \epsilon_m(t) \] where \( f \) is usually unknown (non-parametric? nonlinear?).
Example 1: Dose-response study

- **Modeling**: how to find a functional regression model
  \[ y_m(t) = f_m(x(t), u) + \epsilon_m(t) \]
  where \( f \) is usually unknown (non-parametric? nonlinear?).

- **Prediction**: based on all the up-to-date information for a particular patient and a given dose level, predict Hb level in the next month—dose-response curve.
Example 1: Dose-response study

- Modeling: how to find a functional regression model
  \[ y_m(t) = f_m(x(t), u) + \epsilon_m(t) \]
  where \( f \) is usually unknown (non-parametric? nonlinear?).

- Prediction: based on all the up-to-date information for a particular patient and a given dose level, predict Hb level in the next month—dose-response curve.

- Patient-specific treatment regime: individual dose-response curve (prediction of Hb level against dose level).

- Data: there are only a few observations (13) for each of many subjects (near 200, can have more...).
Example 2: Standing-up manoeuvre of unilateral amputee
Example 2: Standing-up manoeuvre of unilateral amputee

- **Output** $y(t)$:
  Body state eg Cbd position or joint angles (e.g. ANtk: trunk angle).

- **Input** $x(t)$:
  measurements of accelerations and angular velocities (30 variables).

- **Objective**: Use input variables $x(t)$ to predict $y(t)$. 
Modelling standing-up manoeuvres of unilateral amputee: Output CBD-\(x\)
Modelling standing-up manoeuvres of unilateral amputee: Output CBD-z
Modelling standing-up manoeuvres of unilateral amputee: One of input variables accy5
Introduction: nonparametric functional regression model

To find $f$ such that

$$y_m(t) = f_m(x_1(t), x_2(t), \cdots, x_Q(t); u) + \epsilon_m(t)$$

Possible methods for modelling and prediction

- Parametric approaches: the application is limited.
- Nonparametric approaches:
  - Difficulties
    - Large number of functional input variables;
    - Heterogeneity among different curves: the data is collected from different patients, and the patient may use different techniques in different standings-up.
Introduction: nonparametric functional regression model

\[ y_m(t) = f_m(x_1(t), x_2(t), \cdots, x_Q(t); u) + \epsilon_m(t) \]

- Neural Network model (Kamnik et al, 1999);
- Gaussian process regression (GPR) model
  - GPFR for repeated curves (batch data): modeling mean structure and covariance structure simultaneously
    Hierarchical GPR (Shi et al., 2005), GPFR and mixture GPFRs (Shi et al., 2005, 2007 and 2008).
Gaussian process prior for a single curve

\[ y = f(x) + \epsilon. \]

- \( f(\cdot) \) – mapping \( x \in \mathcal{R}^Q \) to \( y \in \mathcal{R} \). It is unknown.
- Define a Gaussian process prior for \( f(\cdot) \):
  - The prior of \( f(\cdot) \) is a Gaussian process with zero mean and kernel covariance \( K(\cdot, \cdot) \).
  - Covariance structure: \( \text{Cov}(f, f') = K(x, x') \).
- Features
  - \( x \) could be large-dimensional.
  - It is easy to calculate prediction (conditional mean).
Gaussian process prior for a single curve

\[ y = f(x) + \epsilon. \]

- \( f(\cdot) \) – mapping \( x \in \mathcal{R}^Q \) to \( y \in \mathcal{R} \). It is unknown.
- Define a Gaussian process prior for \( f(\cdot) \):
  - The prior of \( f(\cdot) \) is a Gaussian process with zero mean and kernel covariance \( K(\cdot, \cdot) \).
  - Covariance structure: \( \text{Cov}(f, f') = K(x, x') \).
- Features
  - \( x \) could be large-dimensional.
  - It is easy to calculate prediction (conditional mean).
  - It is a non-parametric (at least, semi-parametric) model. We can select different kernel covariance function (MacKay, 1997): one example is

\[
K(x, x') = v_1 \exp \left( -\frac{1}{2} \sum_{q=1}^{Q} w_q (x_q - x'_q)^2 \right) + \sum_{q=1}^{Q} a_q x_q x'_q.
\]
“...the selection of a smoothing criterion corresponds to the specification of a prior probability measure over a function space” (Kimerldorf and Wahba, 1970).
Gaussian process prior for a single curve

“...the selection of a smoothing criterion corresponds to the specification of a prior probability measure over a function space” (Kimerldorf and Wahba, 1970).

Suppose that $K(\cdot, \cdot)$ continuous and has a finite trace, then $f(x)$ has a representation

$$f(x) = \sum_{s=1}^{\infty} \theta_s \phi_s(x) = \sum_{s=1}^{S} \theta_s \phi_s(x) + b^{1/2}z(x) \quad (1)$$

where $\lambda_1 \geq \lambda_2 \ldots$, and $\phi_s$ is the eigen-function of $K(\cdot, \cdot)$. $\theta = (\theta_1, \ldots, \theta_S)^t \sim N(0, D^*)$ with $D^* = \text{diag}(\lambda_1, \ldots, \lambda_S)$, and

$$E(z(x)z(x')) = K_1(x, x') = \sum_{s=S+1}^{\infty} \frac{\lambda_s}{b} \phi_s(x)\phi_s(x').$$

We may take $b = \lambda_{S+1}$. $z(x)$ is also a GP with kernel $K_1$. 
Gaussian process prior for a single curve

We therefore have RKHS

$$\mathcal{H}_K = \mathcal{H}_0 \oplus \mathcal{H}_1,$$  \hspace{1cm} (2)

where $\mathcal{H}_0$ is the span of $\phi_1, \cdots, \phi_S$ (null space) and $\mathcal{H}_1$ is the RKHS for $K_1$. 
Gaussian process prior for a single curve

We therefore have RKHS

\[ \mathcal{H}_K = \mathcal{H}_0 \oplus \mathcal{H}_1, \]  

(2)

where \( \mathcal{H}_0 \) is the span of \( \phi_1, \cdots, \phi_S \) (null space) and \( \mathcal{H}_1 \) is the RKHS for \( K_1 \).

Let \( P_1 \) be the orthogonal projection operator in \( \mathcal{H}_K \) onto \( \mathcal{H}_1 \), and \( f_{n,\lambda} \) be the minimiser in \( \mathcal{H}_K \) of the regularised risk functional:

\[ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \| P_1 f \|_K, \]  

(3)
Gaussian process prior for a single curve

We therefore have RKHS

\[ \mathcal{H}_K = \mathcal{H}_0 \oplus \mathcal{H}_1, \]  

(2)

where \( \mathcal{H}_0 \) is the span of \( \phi_1, \ldots, \phi_S \) (null space) and \( \mathcal{H}_1 \) is the RKHS for \( K_1 \).

Let \( P_1 \) be the orthogonal projection operator in \( \mathcal{H}_K \) onto \( \mathcal{H}_1 \), and \( f_{n, \lambda} \) be the minimiser in \( \mathcal{H}_K \) of the regularised risk functional:

\[ \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \| P_1 f \|_K, \]  

(3)

Theorem. Let \( \hat{f}_{GP}(x) = E(f(x)|y_1, \ldots, y_n) \), then

\[ \lim_{D \to \infty} \hat{f}_{GP}(x) = f_{n, \lambda}(x), \]  

(4)

where \( \lambda = \frac{\sigma^2}{nb} \) and \( D = \text{diag}(\lambda_1/b, \ldots, \lambda_S/b) \). \( \lim_{D \to \infty} \) means that each element tends to infinity.
Gaussian process prior for a single curve

- Posterior consistency (Choi, 2005) and information consistency (Seeger, et al. 2008).
Gaussian process prior for a single curve

- Inference:
  - Select a proper kernel covariance function $k(x, x')$;
  - Estimate all the unknown parameters involved;
  - Predict $y^*$ by its posterior mean $E(y^*|\mathcal{D})$. 
Gaussian process prior for a single curve

**Inference:**
- Select a proper kernel covariance function \( k(x, x') \);
- Estimate all the unknown parameters involved;
- Predict \( y^* \) by its posterior mean \( E(y^*|D) \).

**One major problem:**
- Need to calculate \( N \times N \) covariance matrix of \((y_1, \ldots, y_N)\) and its inverse.
- Complexity grows at rate \( O(N^3) \).
Gaussian process prior for a single curve

- **Inference:**
  - Select a proper kernel covariance function \( k(x, x') \);
  - Estimate all the unknown parameters involved;
  - Predict \( y^* \) by its posterior mean \( E(y^*|\mathcal{D}) \).

- **One major problem:**
  - Need to calculate \( N \times N \) covariance matrix of \( (y_1, \ldots, y_N) \) and its inverse.
  - Complexity grows at rate \( O(N^3) \).

- **Possible solutions**
  - Approximate methods: Subset of Regression method (Poggio and Girosi 1990), the Bayesian committee machine (Tresp, 2000), and Selection Mechanisms (Seeger et al., 2003).
  - Filtered GP: (Shi et al., 2005b).
  - Batch data: Using hierarchical GPR (Shi et al., 2005a).
Models for repeated curves (batch data)

\[ y_m(x, t) = f_m(x, t, u) + \epsilon_m(t), \quad m = 1, \ldots, M \]

- If (input) covariates are scalar, a linear functional regression model (Ramsay and Silverman, 1997) is defined as
  \[ f_m(t) = \mu_m(t) = u_m' \beta(t). \]
Models for repeated curves (batch data)

\[ y_m(x, t) = f_m(x, t, u) + \epsilon_m(t), \quad m = 1, \ldots, M \]

- If (input) covariates are scalar, a linear functional regression model (Ramsay and Silverman, 1997) is defined as

\[ f_m(t) = \mu_m(t) = u_m' \beta(t). \]

- Model both mean and covariance structure (Rice and Silverman, 1991)

\[ f_m(t) = \mu_m(t) + \tau_m(t), \]

\( \tau_m(t) \) is a stochastic process with zero mean and covariance function \( C(t, t') = \text{Cov}(y(t), y(t')) \).
Models for repeated curves (batch data)

\[ y_m(x, t) = f_m(x, t, u) + \epsilon_m(t), \ m = 1, \ldots, M \]

- If (input) covariates are scalar, a linear functional regression model (Ramsay and Silverman, 1997) is defined as

\[ f_m(t) = \mu_m(t) = u_m' \beta(t). \]

- Model both mean and covariance structure (Rice and Silverman, 1991)

\[ f_m(t) = \mu_m(t) + \tau_m(t), \]

\( \tau_m(t) \) is a stochastic process with zero mean and covariance function

\[ C(t, t') = \text{Cov}(y(t), y(t')). \]

- Idea of Gaussian process functional regression (GPFR) model:

\[ f_m(x, t) = \mu_m(t) + \tau_m(x). \]
We define a Gaussian Process Functional Regression model as follows:

\[ y_m(x, t) = \mu_m(t) + \tau_m(x) + \epsilon_m, \quad m = 1, \ldots, M, \]

where

- \( \tau_m(x) \sim GP(0, k(x, x'|\theta)) \),
- \( x(t) \) is functional, giving the values of input at each data point.
GPFR models for batch data

We define a Gaussian Process Functional Regression model as follows:

\[ y_m(x, t) = \mu_m(t) + \tau_m(x) + \epsilon_m, \quad m = 1, \ldots, M, \]

where

- \( \tau_m(x) \sim GP(0, k(x, x'|\theta)) \),
  - \( x(t) \) is functional, giving the values of input at each data point.
- \( \mu_m(t) = u_m'\beta(t) \),
  - \( u_m \) is NOT functional, giving the information for each batch (curve).
- \( \beta(t) \) is a set of functional coefficients.
GPFR models for batch data

We define a Gaussian Process Functional Regression model as follows:

\[ y_m(x, t) = \mu_m(t) + \tau_m(x) + \epsilon_m, \quad m = 1, \ldots, M, \]

where

- \( \tau_m(x) \sim GP(0, k(x, x'|\theta)) \),
  \( x(t) \) is functional, giving the values of input at each data point.
- \( \mu_m(t) = u_m'\beta(t) \),
  \( u_m \) is NOT functional, giving the information for each batch (curve).
- \( \beta(t) \) is a set of functional coefficients.
- \( y_m(t, x) \) can be decomposed by

\[ y_m(x, t) = u_m'\beta(t) + \sum_s \phi_s(x)\theta_s + \epsilon_m \]

where \( \phi_s(x) \) is the eigenfunction for kernel covariance function \( K(\cdot, \cdot) \).
Features

- The mean structure models the solid line: the structure is learnt by borrowing information from other subjects.

- Solid line: common mean
- Dashed line: observed curve (common mean + dependent error)
- Dotted line: common mean + independent error
GPFR models for batch data

Features

- The mean structure models the solid line: the structure is learnt by borrowing information from other subjects.
- The covariance structure models the difference between solid line and dashed line: based on the information obtained from the subject.

- Solid line: common mean
- Dashed line: observered curve (common mean + dependent error)
- Dotted line: common mean + independent error
GPFR models for batch data

Features

- The mean structure models the solid line: the structure is learnt by borrowing information from other subjects.
- The covariance structure models the difference between solid line and dashed line: based on the information obtained from the subject.
- Improve the fitting and prediction dramatically

Solid line: common mean
Dashed line: observed curve (common mean + dependent error)
Dotted line: common mean + independent error
GPFR: estimation

\[ y_m(t, x) = u_m' \beta(t) + \tau_m(x) + \epsilon_m \]

- \( \beta(t) \): B-spline approximation:
  \[ \beta(t) = B\Phi(t). \]

- Estimate the unknown parameters \( B \) involved in mean structure and \( \theta \) involved in covariance structure:
  - MLE (or MAP): an iterative procedure is used to update \( B \) and \( \theta \) respectively at each iteration.
  - A simple two-stage method:
    - Stage one: Use least square to estimate \( B \) without assuming any covariance structure.
    - Stage two: Use MLE to estimate \( \theta \) using the mean estimated in Stage one.

- MCMC.
Type I Prediction: interpolation and extrapolation

- Training data $\mathcal{D}$ includes observations in the first $M$ batches and $N$ observations in the $(M + 1)$-th batch $\{y_{M+1,i}, i = 1, \ldots, N\}$.
- To predict $y^*$ at a new test data point $t^*$ in the $(M + 1)$-th batch with the test inputs $x^* = x(t^*)$. 
GPFR: prediction

- **Type I Prediction:** interpolation and extrapolation
  - Training data $\mathcal{D}$ includes observations in the first $M$ batches and $N$ observations in the $(M + 1)$-th batch $\{y_{M+1,i}, i = 1, \ldots, N\}$.
  - To predict $y^*$ at a new test data point $t^*$ in the $(M + 1)$-th batch with the test inputs $x^* = x(t^*)$.

- **Type II Prediction:** a complete new curve
  - Training data $\mathcal{D}$ includes observations in the first $M$ batches;
  - To predict $y^*$ at a new test data point $t^*$ in a complete new $(M + 1)$-th batch.
The prediction and the predictive variance of $y^*$ are

\[
\hat{y}_{M+1} = \hat{\mu}_{M+1}(t^*) + H'(y_{M+1} - \hat{\mu}_{M+1}(t)),
\]
\[
\hat{\sigma}_{M+1}^* = \hat{\sigma}_{GP}^* \left( 1 + u_{M+1}'(U'U)^{-1}u_{M+1} \right).
\]

The proof is given in Shi et al (2007).
GPFR: type II prediction

Predict $y^*$ for a new test input $x^*$ at $t^*$ in a new batch

- Assume that the new batch is batch $m$, and obtain $\hat{y}^*_m$ and $\hat{\sigma}^*_m$. 
GPFR: type II prediction

Predict $y^*$ for a new test input $x^*$ at $t^*$ in a new batch

- Assume that the new batch is batch $m$, and obtain $\hat{y}^*_m$ and $\hat{\sigma}^*_m$.
- Assume that the empirical probability that the new batch is batch $m$ is $1/M$. 
GPFR: type II prediction

Predict $y^*$ for a new test input $x^*$ at $t^*$ in a new batch

- Assume that the new batch is batch $m$, and obtain $\hat{y}^*_m$ and $\hat{\sigma}^*_2$.
- Assume that the empirical probability that the new batch is batch $m$ is $1/M$.
- The prediction is

$$\hat{y}^* = \frac{1}{M} \sum_{m=1}^{M} \hat{y}^*_m,$$

- The predictive variance is

$$\hat{\sigma}^* = \frac{1}{M} \sum_{m=1}^{M} \hat{\sigma}^*_2 / M + \left( \frac{1}{M} \sum_{m=1}^{M} \hat{y}^*_m / M - \hat{y}^* \right)^2.$$
The true model used to generate the data is
\[ y_m(x) = u_m + \sin(0.5x^3) + \tau_m, \]

\[ x = x_i \text{ for } i = 1, \ldots, N_m \] is generated in (-4,4);
\{\tau_m\} is a Gaussian process with zero mean and covariance function
\[ C(x_i, x_j) = v_0 \exp \left( -\frac{1}{2} w_0 (x_i - x_j)^2 \right) + \sigma_0 \delta_{ij}, \]

with \( v_0 = 0.1, \ w_0 = 1.0 \) and \( \sigma_0 = 0.0025; \)
\( u_m \) takes value from \( \{-1, 0, 1\}. \)
Figure: 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.
Training data: 30 curves + 50 data points randomly selected from whole range.

- Both GPFR and GPR give very precise results
GPFR: Simulation study–Extrapolation

Training data: 30 curves + 50 data points randomly selected from [-4,0]

(d) GPFR(.107)  (e) LFR(.415)  (f) GPR(.668)

- **GPFR** always gives very good predictions;
- **GPR**: Good when ’close to’ training data, **BUT** deteriorated very rapidly when move away.
- **Far away** to training data: performance of GPFR and LFR will tend to close.
Table: The average values of \textit{rmse} and \textit{r} between true and predicted responses from simulation study

<table>
<thead>
<tr>
<th>Model</th>
<th>Interpolation</th>
<th>Extrapolation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\textit{rmse}</td>
<td>\textit{r}</td>
</tr>
<tr>
<td>GPFR</td>
<td>0.0588</td>
<td>0.9954</td>
</tr>
<tr>
<td>LFR</td>
<td>0.3244</td>
<td>0.9068</td>
</tr>
<tr>
<td>GPR</td>
<td>0.0830</td>
<td>0.9911</td>
</tr>
</tbody>
</table>

1. The overall \textit{rmse} in range [0,4]
2. The \textit{rmse} in range [0,1]
3. The \textit{rmse} in range [1,4]
GPFR is particularly useful in multiple-step-ahead forecasting.
Hierarchical mixtures of GPFR models

- Define a latent variable $z_m$ for subject $m$;
- Hierarchical mixture models
  - The lower-level model: $y_m(x) | z_m = k \sim GPFR_k(x; \Theta_k)$;
  - The upper-level model: $z_m \sim H$, e.g. a logistic allocation model, Potts model (Green & Richardson, 2000), GP classification model (Shi et al., 2003).
Mix-GPFR: allocation model

We used a logistic allocation model:

\[
P(z_m = k) = \pi_{mk} = \frac{\exp\{v_m' \gamma_k\}}{1 + \sum_{j=1}^{K-1} \exp\{v_m' \gamma_j\}}
\]

for \( k = 1, \ldots, K - 1 \) and

\[
P(z_m = K) = \pi_{mK} = 1 - \sum_{j=1}^{K-1} \pi_{mj},
\]

where \( \gamma_k \)'s are unknown.

Some other allocation models can be used, for example

- Dirichlet mixture
- Nonparametric allocation model
Mix-GPFR: estimation

The $k$-th component of the mixture model is

$$\mu_{mk}(t) = u_m' \beta_k(t), \quad \text{and} \quad \tau_{mk}(x) \sim GP_k(x; \theta_k).$$

Main idea:
- We use a B-spline approximation:
  $$\mu_{mk}(t) = u_m' \beta_k(t) = u_m' B_k' \Phi(t).$$
- The log-likelihood for $\Theta = (B, \theta, \gamma)$ is
  $$L(B, \theta, \gamma) = \sum_{m=1}^{M} \log \left\{ \sum_{k=1}^{K} \pi_{mk} p(y_m | B_k, \theta_k, x_m) \right\},$$
  - Treat the indicator $Z$ as missing, and use an EM algorithm.
Mix-GPFR: Prediction

To predict a new $y^*$ in a batch which we have obtained observations.

- Given $z_{M+1} = k$, the prediction is

$$
\hat{y}_k^* = E(y^*|D, z_{M+1} = k) = \hat{\mu}_{M+1,k}(t^*) + H'_k(y_{M+1} - \hat{\mu}_{M+1,k}(t)),$$

$$\hat{\sigma}^*_k = Var(y^*|D, z_{M+1} = k) = [C_k(x^*, x^*) - H'_k C_k H_k](1 + u_{M+1}'(U'U)^{-1}u_{M+1}),$$

- The overall predictive mean is

$$E(y^*|D) = E_z[E(y^*|D, z_{M+1})]$$

- The overall predictive variance is

$$Var(y^*|D) = E_z[Var(y^*|D, z_{M+1})] + Var_z[E(y^*|D, z_{M+1})]$$
Mix-GPFR: Prediction

The prediction and its variance are given by

\[
\hat{y}^* = E(y^*|D) = \sum_{k=1}^{K} \hat{\alpha}_{M+1,k} \hat{y}_k^*,
\]

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

where

\[
\hat{\alpha}_{M+1,k} = P(z_{M+1} = k|D) = \frac{\hat{\pi}_{M+1,k} p(y_{M+1}|\hat{B}_k, \hat{\theta}_k, x_{M+1})}{\sum_{j=1}^{K} [\hat{\pi}_{M+1,j} p(y_{M+1}|\hat{B}_j, \hat{\theta}_j, x_{M+1})]}.
\]
Mix-GPFR: Prediction

To predict a new $y^*$ in a complete new batch.

- Assume that the observed $M$ batches provide an empirical distribution for the new batch:

$$P(y^* \text{ belongs to the } m\text{-th batch}) = w_m.$$
Mix-GPFR: Prediction

To predict a new \( y^* \) in a complete new batch.

- Assume that the observed \( M \) batches provide an empirical distribution for the new batch:

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

- Calculate Kullback-Leibler divergence by

\[
KL(z_{M+1}, z_m) = \sum_{k=1}^{K} \hat{\pi}_{M+1,k} \log \frac{\hat{\pi}_{M+1,k}}{\hat{\pi}_{mk}}
\]

- Define \( w_m \propto 1/KL(z_{M+1}, z_m) \).
Mix-GPFR: Prediction

To predict a new $y^*$ in a complete new batch.

- Assume that the observed $M$ batches provide an empirical distribution for the new batch:
  \[ P(y^* \text{ belongs to the } m\text{-th batch}) = w_m. \]

- Calculate Kullback-Leibler divergence by
  \[ KL(z_{M+1}, z_m) = \sum_{k=1}^{K} \hat{\pi}_{M+1,k} \log \frac{\hat{\pi}_{M+1,k}}{\hat{\pi}_{mk}}. \]

- Define $w_m \propto 1/KL(z_{M+1}, z_m)$.
- The overall prediction for $y^*$ is
  \[ \hat{y}^* = E(y^*|D) = \sum_{m=1}^{M} w_m y_m^*. \]

  where $y_m^*$ is the predictive mean if we assume $y^*$ belongs to $m$-th batch.
Model based clustering: based on the shape of response curves, i.e. the curve shape of $y(t)$.

If $y(t)$ depends on a set of covariate curves, i.e. $y = y(x)$, how to do clustering? based on the surface shape $y(x)$.

Basic idea: calculate the posterior probability

$$P(z_m = k | D).$$

The curve (surface) $y_m(x)$ is classed as belonging to the $k^*$-th cluster if $P(z_m = k^* | D)$ takes its maximum value at $k = k^*$. 
Figure: 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.
Simulation study: Prediction

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

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Interpolation</th>
<th>Extrapolation</th>
<th>New curve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$rmse$</td>
<td>$r$</td>
<td>$rmse$</td>
</tr>
<tr>
<td>mix-gpfr</td>
<td>0.0595</td>
<td>0.9871</td>
<td>0.2492</td>
</tr>
<tr>
<td>gpfr</td>
<td>0.1229</td>
<td>0.9425</td>
<td>0.4124</td>
</tr>
<tr>
<td>mix-mean</td>
<td>0.3809</td>
<td>0.6331</td>
<td>0.3786</td>
</tr>
<tr>
<td>mix-gp</td>
<td>0.1926</td>
<td>0.8564</td>
<td>0.4729</td>
</tr>
</tbody>
</table>

- Mixture model is better than a single model
- Interpolation: GPFR and GPR perform much better than Mean model.
- Extrapolation: Close to training data: GPFR still performs very good, GPR and Mean models are okay. Moving away from the training data: GPFR and Mean models are okay, but GPR failed.
Simulation study: multi-step-ahead forecasting

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

<table>
<thead>
<tr>
<th></th>
<th>1-step</th>
<th></th>
<th>3-step</th>
<th></th>
<th>6-step</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rmse</td>
<td>$r$</td>
<td>rmse</td>
<td>$r$</td>
<td>rmse</td>
<td>$r$</td>
</tr>
<tr>
<td>mix-gpfr</td>
<td>0.0712</td>
<td>0.9844</td>
<td>0.1081</td>
<td>0.9589</td>
<td>0.2128</td>
<td>0.8735</td>
</tr>
<tr>
<td>gpfr</td>
<td>0.1080</td>
<td>0.9648</td>
<td>0.2355</td>
<td>0.7873</td>
<td>0.3799</td>
<td>0.5567</td>
</tr>
<tr>
<td>mix-mean</td>
<td>0.4291</td>
<td>0.5483</td>
<td>0.4106</td>
<td>0.6388</td>
<td>0.4568</td>
<td>0.6727</td>
</tr>
<tr>
<td>mix-gp</td>
<td>0.1288</td>
<td>0.9504</td>
<td>0.3744</td>
<td>0.2722</td>
<td>0.4411</td>
<td>0.2281</td>
</tr>
</tbody>
</table>

*The conclusion is similar to 'extrapolation'.
Figure: Error rate: 2% based on surface shape $y(x)$, and 13.5% based on curve shape $y(t)$. 
Simulation study: Clustering

(a) 90 sample curves

(b) BIC

Figure: error rate 0.5% based on surface shape $y(x)$, and 35.5% based on curve shape $y(t)$. 
Real data: standing-up manoeuvres

Table: 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 |
Real data: standing-up manoeuvres

Figure: 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.
Real data: standing-up manoeuvres

**Figure:** Paraplegia data. (a) The values of BIC. (b) Height and weight for each patient.
GPFR model performs really well on prediction and clustering for the repeated functional data with large dimensional functional covariates; there are many interesting problems, for example:

- Selection of kernel covariance function;
- Model selection (the problem of $Q > n$, number of clusters);
- Categorial functional data;
- Dynamic control problem.
Statistical properties:

- Given a random sample of \( n \) observations, the covariance kernel \( K \) can be expanded into a feature space of dimension \( n \) as

\[
K_n(x, x') = \sum_{i=1}^{n} \lambda_i \phi_s(x) \phi_s(x').
\]

In GPFR model, there are \( N_m \) observations in the \( m \)-th batch, what are the sample errors and abstract approximation errors here (Cucker and Smale, 2001)?

- Posterior or information consistency: two types of information—from \( N_m \) observation in each subject and from \( M \) subjects.
Acknowledgements

- Dr. Roman Kamnik (Slovenia)
- Prof. Rod Murray-Smith (Glasgow)
- Prof. Mike Titterington (Glasgow)
- Dr. Bo Wang (York)
- Dr. Robert West (Leeds)
Thank you...

http://www.staff.ncl.ac.uk/j.q.shi