Gaussian Process Regression Analysis for Large Functional Data

Jian Qing SHI

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

High dimensional and dependent functional data
Research workshop: 10-12 September 2012, Bristol, UK
Outline

1 Introduction

2 Gaussian process functional regression (GPFR) model
   - Gaussian process prior for a single curve
   - Models for repeated curves (batch data)
   - Model learning
   - Numerical studies

3 GPR: variable selection
   - Penalized GPR
   - Selection of Grouped Variables - ‘NET’ PGPRs
   - Examples
   - Asymptotic Theory
   - Classification

4 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.
- **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?).

- **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 $y(t)$. 

![Diagram of human body with annotations for accelerations and joint angles]
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

- If $Q$ is small, e.g. $Q = 1$ or 2, most of conventional methods can be used (e.g. Spline smoothing, local polynomial models).
- If $Q$ is large, the conventional methods suffer from curse of dimensionality. Alternative methods include
  - Additive model (Breiman and Friedman, 1985; Hastie and Tibshirani, 1990).
  - Varying coefficient model (Hastie and Tibshirani, 1993; Fan and Zhang, 1999).
  - Dimension reduction methods: projection pursuit, sliced inverse regression, single index model.
  - Neural Network model (Cheng and Titterington, 1994, Neal 1996);
  - Gaussian process regression (GPR) model
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
  - It provides a flexible nonlinear model;
  - \( x \) could be large-dimensional;
  - Need to select a parametric covariance kernel, for example the following covariance function (squared exponential + linear).

\[
K(x, x'; \theta) = v_1 \exp \left( -\frac{1}{2} \sum_{q=1}^{Q} w_q (x_q - x_q')^2 \right) + \sum_{q=1}^{Q} a_q x_q x_q'.
\]

where \( \theta = (v_1, w_1, \ldots, w_Q, a_1, \ldots, a_Q) \) – hyper-parameters or tuning parameters.
GPR for a single curve: inference

• How to choose the values of hyper-parameters $\theta$?
  ▶ GCV (only if the dimension of $\theta$ is very small)
  ▶ Empirical Bayesian approach: MAP
  ▶ Fully Bayesian: assume a hyper-prior for $\theta$ and then use MCMC.

• A GPR model is **generally formulated** as

$$y_i | f_i \sim g(f_i) \quad \text{and}$$

$$(f_1, \ldots, f_n) \sim GP(0, k(\cdot, \cdot; \theta)).$$

• If

$$y_i | f_i \sim N(f_i, \sigma_c^2),$$

the marginal distribution of $y_i$ is still a normal distribution.

• In general,

$$p(y | \theta) = \int p(y | f) p(f | \theta) df.$$ 

• Implementing/computing issues: [http://www.gaussianprocess.org/](http://www.gaussianprocess.org/)
Theorem

(Choi, 2005) Let $P_0$ denote the joint conditional distribution of $\{Y_n\}_{n=1}^\infty$ given the covariate assuming that $f_0$ is the true response function. Suppose that the values of the covariate in $[0, 1]$ are fixed, i.e., known ahead of time. Then for every $\epsilon > 0$,

$$\Pi \left\{ f \in W_{\epsilon,n}^C | \mathcal{D} \right\} \rightarrow 0 \text{ a.s. } [P_0].$$

The neighbourhood is defined as

$$W_{\epsilon,n} = \left\{ (f, \sigma) : \int |f(x) - f_0(x)| dQ_n(x) < \epsilon, \left| \frac{\sigma}{\sigma_0} - 1 \right| < \epsilon \right\}.$$
GPR: asymptotic results – information consistency

- K-L distance: \( D[p\|q] = \int (\log p - \log q) dP. \)
- Lower bound of \( D[P(y_1, \ldots, y_n|f_0)\|P_{bs}(y_1, \ldots, y_n)] \),

\[
D[P(y_1, \ldots, y_n|f)\|P_{bs}(y_1, \ldots, y_n)] \leq \frac{1}{2} \|f\|_K^2 + \frac{1}{2} \log |I_n + cK|, \quad (2)
\]

- \( \|f\|_K \) is the RKHS norm of \( f \), and \( c \) is a certain constant.
- \( P_{bs}(y_1, \ldots, y_n) \) – a Bayesian GP prediction strategy based on \( n \) observations.

- \( P_{bs}(y^*|\mathcal{D}) = \int p_f(y^*)d\Pi(f|\mathcal{D}) \), here \( y^* \) is a future observation.
- Thus the expected KL divergence between \( P_{bs}(y^*|\mathcal{D}) \) and \( P_{bs}(y^*|f_0) \) converges to zero as the sample size increases (Seeger, et al. 2008).
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 \( C(t, t') = \text{Cov}(y(t), y(t')) \). Note that \( t \) is one-dimensional.

- Gaussian process functional regression (GPFR) model (Shi et al. 2007):
  \[ f_m(x, t) = \mu_m(t) + \tau_m(x). \]
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.
- If we take \( \mu_m(t) = u_m' \beta(t) \), then \( y_m(t, x) \) can be decomposed by

\[ y_m(x, t) = u_m' \beta(t) + \sum_j \phi_j(x) \gamma_j + \epsilon_m \]

where \( \phi_j(x) \) is the eigenfunction for covariance function \( K(\cdot, \cdot) \) and \( \gamma_j \sim N(0, \lambda_j) \).
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.
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 $x^* = x(t^*)$.

The prediction and the predictive variance of $y^*$ are

\[
\hat{y}_{M+1}^* = \hat{\mu}_{M+1}(t^*) + H'(y_{M+1} - \hat{\mu}_{M+1}(t)), \\
\hat{\sigma}^*_{M+1} = \hat{\sigma}^2_{GP} \left(1 + u'_{M+1}(U'U)^{-1}u_{M+1}\right).
\]
GPFR: prediction for a completely new curve

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

- Using mean model: $\hat{y}_{M+1}^* = \hat{\mu}_{M+1}(t^*)$;
- Using both mean and covariance models:
  - If the new batch is the same as batch $m$, and obtain $\hat{y}_m^*$ and $\hat{\sigma}_m^2$.
  - Assume that $P($the new batch belongs to batch $m) = w_m$,

  \[ \hat{y}^* = \sum_{m=1}^{M} w_m \hat{y}_m^*, \]

  \[ \hat{\sigma}^2 = \sum_{m=1}^{M} w_m \hat{\sigma}_m^2 + \left( \sum_{m=1}^{M} w_m \hat{y}_m^2 - \hat{y}^2 \right). \]

  - $w_m$ may be modelled by a ‘spatially indexed’ model (Shi and Wang 2008).
GPFR models for batch data

- Solid line: common mean
- Dashed line: the real curve for a subject

Features

- The mean structure models the solid line: the structure is learnt by borrowing information from other subjects.
- If no data is collected for the \((M + 1)\)-th subject,

\[
\hat{y}_{M+1}^* = \hat{\mu}_{M+1}(t^*)
\]

- It is a consistent estimator of the common mean (solid line).
GPFR models for batch data

Features

- Usually some data is collected: $\hat{y}_{M+1}^{\ast}$ would be
  $$\hat{\mu}_{M+1}(t^\ast) + H'(y_{M+1} - \hat{\mu}_{M+1}(t)).$$
- When the sample size is sufficiently large, the above prediction is a consistent estimate of $f_{M+1}$ (dashed line).
- Improve the fitting and prediction dramatically.
- It is very useful in applications, e.g., construct individual dose-response curve and thus enable for patient-specific treatment regime.

Solid line: common mean
Dashed line: the real curve for a subject
The true model used to generate the data is
\[ y_m(x) = u_m + \sin(0.5x)^3 + \tau_m, \]
\[ x = x_i \text{ for } i = 1, \ldots, N_m \text{ is generated in } (-4,4); \]
\[ \{\tau_m\} \text{ is a Gaussian process with zero mean and covariance function} \]
\[ C(x_i, x_j) = v_0 \exp \left( -\frac{1}{2} w_0 (x_i - x_j)^2 \right) + \sigma_0 \delta_{ij}, \]

with \( v_0 = 0.1, w_0 = 1.0 \) and \( \sigma_0 = 0.0025; \)
\[ u_m \text{ takes value from } \{-1, 0, 1\}. \]
GPFR: Simulation study for batch data: data

**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.
GPFR: Simulation study–Interpolation

Figure: Training data: 30 curves + 50 data points randomly selected from whole range. Left: GPFR, Middle: Mean model and Right: GPR

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

**Figure:** Training data: 30 curves + 50 data points randomly selected from [-4,0]. Left: GPFR, Middle: Mean model and Right: GPR

- **GPR:** Good when 'close to' training data, **BUT** deteriorated very rapidly when move away.
- **GPFR:** very good when 'close to' training data; performance of GPFR will tend to close to LFR when moving away from the training data.
- GPFR is particular useful in multiple-step-ahead forecasting
GPFR: Simulation study–prediction

**Table:** The average values of rmse and 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>rmse</td>
<td>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 rmse in range [0,4]
2. The rmse in range [0,1]
3. The rmse in range [1,4]
GPFR: Leeds Renal Data –individual dose-response curves

**Figure:** Renal data: Hb response for different dose level (drug D)
Gaussian process regression model for a single curve

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

- \( f(\cdot) \sim GPR(x|k(\cdot, \cdot)); \)
- \( k(\cdot, \cdot; \theta) \) covariance kernel/function, depending on \( x; \)
- \( Q \) – could be large dimensional;
- What if \( Q \) is very large, or even \( Q \gg n? \)
GPR: variable selection

- Choose values of hyper-parameters $\theta$ by empirical Bayesian learning:

$$p(\theta|D) \propto p(y|\theta)p(\theta)$$

  - MAP: choose $\hat{\theta}$ by maximising $p(\theta|D)$.

- Variable selection when $Q$ is very large, for e.g.

$$K(x, x'; \theta) = v_1 \exp \left( -\frac{1}{2} \sum_{q=1}^{Q} w_q (x_q - x'_q)^2 \right).$$

  - Hard threshold or ARD (Automatic Relevance Determination): remove those variables with small 'w' values.
  - Subset selections and PCA (Chen et al., 2007).
  - Penalized techniques (Yi et al. 2011).
Penalized GPR: idea

\[ K(x, x'; \theta) = v_1 \exp \left( -\frac{1}{2} \sum_{q=1}^{Q} w_q (x_q - x'_q)^2 \right). \]

- Empirical Bayesian learning - choose the values of hyper-parameters by maximize the marginal pdf, or

\[ \hat{\theta} = \arg \min_{\theta} [ -l_n(\theta; D) ]. \]

- Penalized GPR: penalize \( w_q \)'s by minimizing

\[ l_p(\theta; D, \lambda_n) = -\frac{1}{n} l_n(\theta) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q). \]
Penalized GPR: LASSO PGPR

LASSO PGPR: to minimize

\[ l_p(\theta; D, \lambda_n) = -\frac{1}{n} l_n(\theta; D) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q) \]

where \( P_{\lambda_n}(w_q) = \lambda_n |w_q| \).

Algorithm

- Given \( \lambda_n \), \( \hat{\theta} = \arg\min_{\theta} \left[ -\frac{1}{n} l_n(\theta; D) + \lambda_n \sum_{q=1}^{Q} |w_q| \right] \).
- Some \( \hat{w}_q \)'s are equal to zero.
- Select the optimal \( \lambda_n \) by GCV.
Penalized GPR: other penalty functions

To minimize

\[ l_p(\theta; D, \lambda_n) = -\frac{1}{n} l_n(\theta; D) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q) \]

- **Ridge penalty**: \( P_{\lambda_n}(w_q) = \lambda_n w_q^2 \).
  - cannot be used for variable selection.
- **Bridge penalty**: \( P_{\lambda_n}(w_q) = \lambda_n w_q^\gamma, (0 < \gamma < 1) \).
  - Need to select two tuning parameters \( \lambda_n \) and \( \gamma \) by GCV.
- **Adaptive LASSO PGPR**: \( p_{\lambda_n}(|w|) = \lambda_n \sum_{q=1}^{Q} \psi_q |w_q| \).
  - Zou (2006) constructs the weight vector as \( \hat{\psi}_q = 1/\hat{w}_q^\gamma \) for \( \gamma > 0 \).
  - There are two tuning parameters: \( \lambda_n \) and \( \gamma \).
Penalized GPR: other penalty functions

To minimize

\[ l_p(\theta; D, \lambda_n) = -\frac{1}{n} l_n(\theta; D) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q) \]

- **SCAD penalty:**

\[
p_{\lambda_n}(|w|) = \begin{cases} 
\lambda |w| & \text{if } |w| \leq \lambda_n, \\
-\frac{|w|^2 - 2a\lambda_n|w| + \lambda_n^2}{2(a-1)} & \text{if } \lambda_n < |w| \leq a\lambda_n, \\
\frac{(a+1)\lambda_n^2}{2} & \text{if } |w| > a\lambda_n.
\end{cases}
\]

where \( a > 1 \).

- There are two tuning parameters: \( \lambda_n \) and \( a \).
Penalized GPR: comparisons

- SCAD, Adaptive LASSO and Bridge PGPR achieve some nice asymptotic properties (e.g. sparsity), but the computation in GCV is very heavy.
- Numerically, SCAD, Adaptive LASSO and ridge PGPR achieved better results than others when the input variables are highly correlated.
Selection of Grouped Variables - Elastic NET PGPR

- To select variables which are naturally grouped (highly correlated) - Elastic NET PGPR:

\[ l_p(\theta; D, \lambda_1, \lambda_2) = -\frac{1}{n} l_n(\theta; D) + \lambda_1 \sum_{q=1}^{Q} |w_q| + \lambda_2 \sum_{q=1}^{Q} w_q^2. \]

- Elastic NET is constructed by adding LASSO and Ridge penalties together.
- Thus can achieve the advantages of both penalties.
- Advantage: select naturally grouped variables.
- Disadvantage: double bias from both Ridge and LASSO penalties.
Selection of Grouped Variables - other NET penalties

- **SCAD net:**

  \[ l_p(\theta; \mathcal{D}, a, \lambda_1, \lambda_2) = -\frac{1}{n} l_n(\theta) + \lambda_1 \sum_{q=1}^{Q} P_{\lambda_1,a}(w_q) + \lambda_2 \sum_{q=1}^{Q} w_q^2. \]

  - Has the properties of both SCAD and Ridge penalties.
  - Select variables which are naturally grouped with less bias than the Elastic NET PGPR.

- **Bridge NET:**

  \[ l_p(\theta; \mathcal{D}, a, \lambda_1, \lambda_2) = -\frac{1}{n} l_n(\theta) + \lambda_1 \sum_{q=1}^{Q} w_q^\gamma + \lambda_2 \sum_{q=1}^{Q} w_q^2. \]

  - Has the properties of both Bridge and Ridge penalties.
  - Select variables which are naturally grouped with less bias than the Elastic NET PGPR.
Examples - Prostate Cancer Data

- Response variable: \( \log(\text{prostate-specific antigen}) \). 8 input variables: age, \( \log(\text{cancer volume}) \) etc.
- Training data: 67 observations. Test data: 30 observations.

<table>
<thead>
<tr>
<th>Methods Used</th>
<th>Tuning Parameter</th>
<th>RMSE-PredVar</th>
<th>Variables Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLE (Linear)</td>
<td>( \lambda_n = 1 )</td>
<td>0.586 (0.184)</td>
<td>All</td>
</tr>
<tr>
<td>Ridge (Linear)</td>
<td>( \lambda_n = 1 )</td>
<td>0.566 (0.188)</td>
<td>All</td>
</tr>
<tr>
<td>Lasso (Linear)</td>
<td>( s = 0.39 )</td>
<td>0.499 (0.161)</td>
<td>(1,2,4,5,8)</td>
</tr>
<tr>
<td>MLE (GPR)</td>
<td>( \lambda_n = 1 )</td>
<td>0.495 (0.073)</td>
<td>All</td>
</tr>
<tr>
<td>Ridge (GPR)</td>
<td>( \lambda_n = 1.7 )</td>
<td>0.471 (0.061)</td>
<td>All</td>
</tr>
<tr>
<td>LASSO (GPR)</td>
<td>( \lambda_n = 0.06 )</td>
<td>0.464 (0.057)</td>
<td>(1,2,3,4,5,7,8)</td>
</tr>
<tr>
<td>Bridge (GPR)</td>
<td>( \gamma = 0.1, \lambda_n = 0.05 )</td>
<td>0.415 (0.025)</td>
<td>(1,2,5)</td>
</tr>
<tr>
<td>SCAD (GPR)</td>
<td>( a = 3.7, \lambda_n = 1.8 )</td>
<td>0.453 (0.034)</td>
<td>(1,2,4,5,8)</td>
</tr>
<tr>
<td>Adap. LASSO (GPR)</td>
<td>( \gamma = 0.8, \lambda_n = 0.18 )</td>
<td>0.413 (0.025)</td>
<td>(1,2,5)</td>
</tr>
</tbody>
</table>
Examples - Meat Fat Data using near infrared spectroscopy (NIRS)

Response variable: fat contents. 100 input variables: measurement of the absorption with different wavelength – highly correlated. training data: 172. Test data: 43.

<table>
<thead>
<tr>
<th>Method</th>
<th>RMSE</th>
<th>Number of Variables Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCR</td>
<td>2.855</td>
<td>All</td>
</tr>
<tr>
<td>PLS</td>
<td>2.560</td>
<td>All</td>
</tr>
<tr>
<td>QPLS</td>
<td>0.995</td>
<td>All</td>
</tr>
<tr>
<td>Neural Network</td>
<td>1.418</td>
<td>All</td>
</tr>
<tr>
<td>10-6-1 Network, early stopping</td>
<td>0.65</td>
<td>10</td>
</tr>
<tr>
<td>10-3-1 Network, Bayesian</td>
<td>0.52</td>
<td>10</td>
</tr>
<tr>
<td>13-X-1 Network, Bayesian ARD</td>
<td>0.36</td>
<td>13</td>
</tr>
<tr>
<td>GPR(MLE)</td>
<td>0.89</td>
<td>All</td>
</tr>
<tr>
<td>GPR(Ridge)</td>
<td>0.711</td>
<td>All</td>
</tr>
<tr>
<td>GPR(LASSO)</td>
<td>0.649</td>
<td>26</td>
</tr>
<tr>
<td>GPR(Bridge)</td>
<td>0.432</td>
<td>4</td>
</tr>
<tr>
<td>GPR(SCAD)</td>
<td>0.5297</td>
<td>15</td>
</tr>
<tr>
<td>GPR(Adaptive LASSO)</td>
<td>0.3901</td>
<td>3</td>
</tr>
</tbody>
</table>
Examples - Paraplegia Standing-up Data

Response variable: vertical trajectory of the body centre of mass. Input variables: 33.

<table>
<thead>
<tr>
<th></th>
<th>RMSE</th>
<th>Pred Var</th>
<th>No. of Var Sel</th>
<th>Tuning Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPR(Hard)</td>
<td>16.3034</td>
<td>46.0874</td>
<td>6</td>
<td>N/A</td>
</tr>
<tr>
<td>GPR(Ridge)</td>
<td>12.5814</td>
<td>32.5563</td>
<td>N/A</td>
<td>$\lambda_n = 0.01$</td>
</tr>
<tr>
<td>GPR(LASSO)</td>
<td>12.1583</td>
<td>46.4524</td>
<td>11</td>
<td>$\lambda_n = 0.00002$</td>
</tr>
<tr>
<td>GPR(Bridge)</td>
<td>9.6093</td>
<td>23.6331</td>
<td>5</td>
<td>$\gamma = 0.01, \lambda_n = 0.8$</td>
</tr>
<tr>
<td>GPR(AdLASSO)</td>
<td>78.8941</td>
<td>36.6152</td>
<td>2</td>
<td>$\gamma = 0.5, \lambda_n = 0.08$</td>
</tr>
</tbody>
</table>
Asymptotic Theories

\[ a_n = \max \left\{ P_{\lambda_n}'(w_q(0)) : q \in \mathcal{A} \right\}, \quad b_n = \max \left\{ P_{\lambda_n}''(w_q(0)) : q \in \mathcal{A} \right\}. \]

Theorem

- Let \( p_{\theta}^{n} \) denote the joint probability density of \( \{(y_i, x_i)\}_{i=1}^{n} \) that satisfies some regularity conditions \((C1)-(C4)\).

- Assume that the penalty function \( P_{\lambda_n} \) satisfies
  1. \( P_{\lambda_n}(w_q) \geq 0 \) and \( P_{\lambda_n}(0) = 0 \) and
  2. \( P_{\lambda_n}(w_q^*) \geq P_{\lambda_n}(w_q) \) if \( |w_q^*| \geq |w_q| \).

- There exists a sequence \( r_n \to \infty \) so that \( \hat{\theta} \) is \( r_n \) consistent.

If \( b_n \) converges to 0, then there exists a local minimizer \( \hat{\theta}_n \) of \( l_p(\theta) \) such that
\[ \|\hat{\theta}_n - \theta\| = O_p(r_n^{-1} + a_n). \]
Asymptotic Theories

Let 
\[ \mathcal{A} = \{ q : w_q^{(0)} \neq 0 \} \quad \text{and} \quad \mathcal{B} = \{ q : w_q^{(0)} = 0 \}, \]

Theorem

(Sparsity) Let \( \hat{\theta}_n = [ \hat{w}'_A, \hat{w}'_B, \hat{v}_0, \hat{\sigma}^2_v ]' \) be the \( r_n \)-consistent local optimizer of \( l_p(\theta) \) in Theorem 1. Assume the same regularity conditions \((C1)–(C4)\) also hold as in Theorem 1. In addition, assume that

1. \( \lim \inf_{n \to \infty} \lim \inf_{\theta \to 0^+} \frac{1}{\lambda_n} \frac{\partial P_{\lambda_n}(\hat{\theta})}{\partial w_q} > 0 \)

2. \( \lambda_n \to 0 \) and \( \frac{n\lambda_n}{r_n} \to \infty \) as \( n \to \infty \).

Therefore, with probability tending to 1, model sparsity can be achieved, i.e.

\[ \lim_{n \to \infty} P(\hat{w}_B = 0) = 1. \]
Penalized Gaussian process classification - PGPC

- $t_i|x_i \sim \text{Bin}(1, \pi_i(x_i))$.
- We use the logistic link function $f(x_i) \triangleq \text{logit}(\pi_i(x_i)) = \log \left( \frac{\pi_i}{1-\pi_i} \right)$.
- $\pi_i = p(t_i = 1|f(x_i)) = \frac{1}{1+\exp(-f(x_i))}$.
- $f(\cdot) \sim \text{GPR}(0, k(\cdot, \cdot)|x)$.
- $k(x_i, x_j; \xi) = v_0 \exp \left( -\frac{1}{2} \sum_{q=1}^{Q} w_q (x_{iq} - x_{jq})^2 \right)$, where $\xi = [w_1, \ldots, w_Q, v_0]$. 
Penalized Gaussian process classification - PGPC

Marginal density:

\[ p(t|X) = \int p(t, f|X) df \]

\[ = \int p(t|X, f)p(f|X) df \]

\[ = \int \prod_{i=1}^{N} \pi_i^{t_i}(1 - \pi_i)^{1-t_i} p(f|X) df \]

\[ = \int \prod_{i=1}^{N} \left( \frac{1}{1 + \exp(-f_i)} \right)^{t_i} \left( 1 - \frac{1}{1 + \exp(-f_i)} \right)^{1-t_i} p(f|X) df \]
Penalized Gaussian process classification - PGPC

- Marginal log-likelihood

\[ l_n(\xi) = \log p(t|X, \xi) = \log \int p(t|X, f, \xi)p(f|X, \xi)df = \log \int \exp(\Phi(f))df. \]

- Laplace approximation

\[ \int \exp(\Phi(f))df \approx \exp \left\{ \Phi(\hat{f}) + \frac{N}{2} \log 2\pi - \frac{1}{2} \log |C^{-1} + K| \right\}, \]

where \( K = \nabla\nabla \log p(t|X, f, \xi) \)

- Penalized likelihood:

\[ l_p(\xi) = -l_n(\xi) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q). \]

J. Q. Shi (Newcastle University, UK)  Gaussian Process Regression Analysis  44 / 56
Penalized Gaussian process classification - Leukaemia Cancer Data

- 2 types of Leukaemia Cancer, Acute Myeloid Leukaemia (AML) and Acute Lymphoblastic Leukaemia (ALL).
- 7129 genes (input variables).
- Training data (38): 27 cases of ALL and 11 cases of AML.
- Test data (34): 20 cases of ALL and 14 cases of AML.
- typical large $p$ small $n$ problem. (here is large $Q$ small $n$.)
Penalized Gaussian process classification - PGPC
Penalized Gaussian process classification - PGPC

Gene Expression Data for Training (30 Genes Selected)
Penalized Gaussian process classification - PGPC

LASSO Selected Gene Expression Data for Test Data

Sample Number

Selected Genes

J. Q. Shi (Newcastle University, UK)  Gaussian Process Regression Analysis
Penalized Gaussian process classification - PGPC

ENET Selected Gene Expression Data for Test Data

Selected Genes

Sample Number
## Penalized Gaussian process classification - PGPC

<table>
<thead>
<tr>
<th>Method</th>
<th>5-fold GCV Error</th>
<th>ClassError</th>
<th>No. of Genes Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Golub</td>
<td>3/38</td>
<td>4/34</td>
<td>50</td>
</tr>
<tr>
<td>ENET Linear</td>
<td>3/38</td>
<td>0/34</td>
<td>45</td>
</tr>
<tr>
<td>LASSO PGPC</td>
<td>4/38</td>
<td>3/34</td>
<td>30</td>
</tr>
<tr>
<td>ENET PGPC</td>
<td>2/38</td>
<td>1/34</td>
<td>22</td>
</tr>
</tbody>
</table>
Suppose that $z_m(t)$ has a distribution from exponential family, a generalized GPFR model (Wang and Shi, 2012) can be defined as

$$E(z_m(t)|\tau_m(t)) = h(\mu_m(t) + \tau_m(t)),$$

$$\tau_m(t) = \tau_m(x_m(t)) \sim GPR(0, k(\cdot, \cdot; \theta)|x_m(t)).$$
A functional linear regression model with a scalar response $y \in \mathbb{R}$ is defined by

$$y = \mu + \int_S \beta(s)(x(s) - \mu_x(s))ds + \epsilon,$$

where $\mu_x(s) = E(x(s))$ and $\epsilon$ is mean-zero noise, $x(s) \in L^2(S)$ where $S$ is a subset of the real line $\mathbb{R}$.

In general, a nonlinear functional model is

$$y = g(x_1(s), \ldots, x_p(s), z_1, \ldots, z_q) + \epsilon = g(x(s), z) + \epsilon,$$
Comments – future work

A nonlinear **GP function-on-function** model may be defined as (in progress)

- If $g(\cdot)$ depends on $x(s)$ only,

  $$g(x(s)) \sim fGPR[\mu, k_f(\theta)|x(s)]$$

  where the covariance kernel depends on two sets of functional input covariates, e.g.

  $$\text{Cov}[g(x_i(s)), g(x_j(s))] = k_f[x_i(s), x_j(s); \theta]$$

  $$= v_0 \exp \left\{ -\frac{1}{2} \sum_{k=1}^{p} w_k \|x_{ik}(s) - x_{jk}(s)\|_f^2 \right\}.$$

  Here $\|x_{ik}(s) - x_{jk}(s)\|_f^2$ is the norm between two functions, for example a $L^2$ norm $\|x_{ik}(s) - x_{jk}(s)\|_f^2 = \int_S (x_{ik}(s) - x_{jk}(s))^2 ds$.

- If $g(\cdot)$ depends on both $x(s)$ and $z$, we may extend the above with a new covariance kernel by multiplication of two covariance kernels:

  $$k[(x_i(s), z_i), (x_j(s), z_j)] = k_f[x_i(s), x_j(s)] \cdot k(z_i, z_j).$$
GPFR model performs very well on prediction and clustering for the repeated functional data with large dimensional functional covariates;

There are still many interesting statistical problems, for example

- Selection of kernel covariance function and the related theory;
- Empirical Bayesian learning and the related theory (e.g. convergence rate);
- Extensions: e.g.
  - Dynamic nonlinear control problems;
  - Nonparametric functional latent variable models;
  - Function-on-function regression model
Comments – penalized technique

- Penalized GPR works well.
- Need to develop an efficient optimization algorithm particularly for classification problem or other problems with categorical functional data.
- More research on group selection, particularly when the input variables is high-dimensional and highly correlated.

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

Thank you ...