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

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 $\mathbf{x}(t)$: including e.g. $x_1(t)$–dose level; $x_2(t)$–time taking the drug; $x_3(t)$–iron dose.
  - Subject based scalar covariates $\mathbf{u}$: including e.g. age, weights, gender.

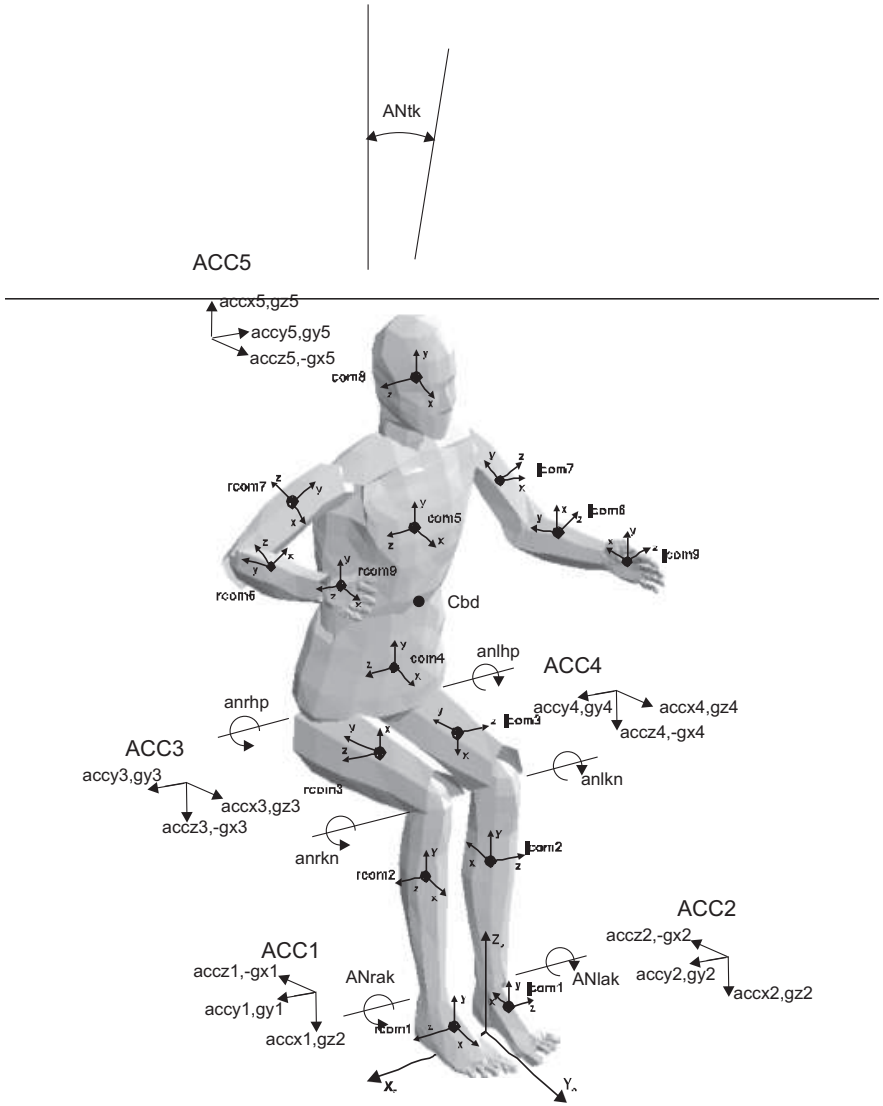
Example 1: Dose-response study

- Modeling: how to find a functional regression model $y_m(t) = f_m(\mathbf{x}(t), \mathbf{u}) + \epsilon_m(t)$ 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...).

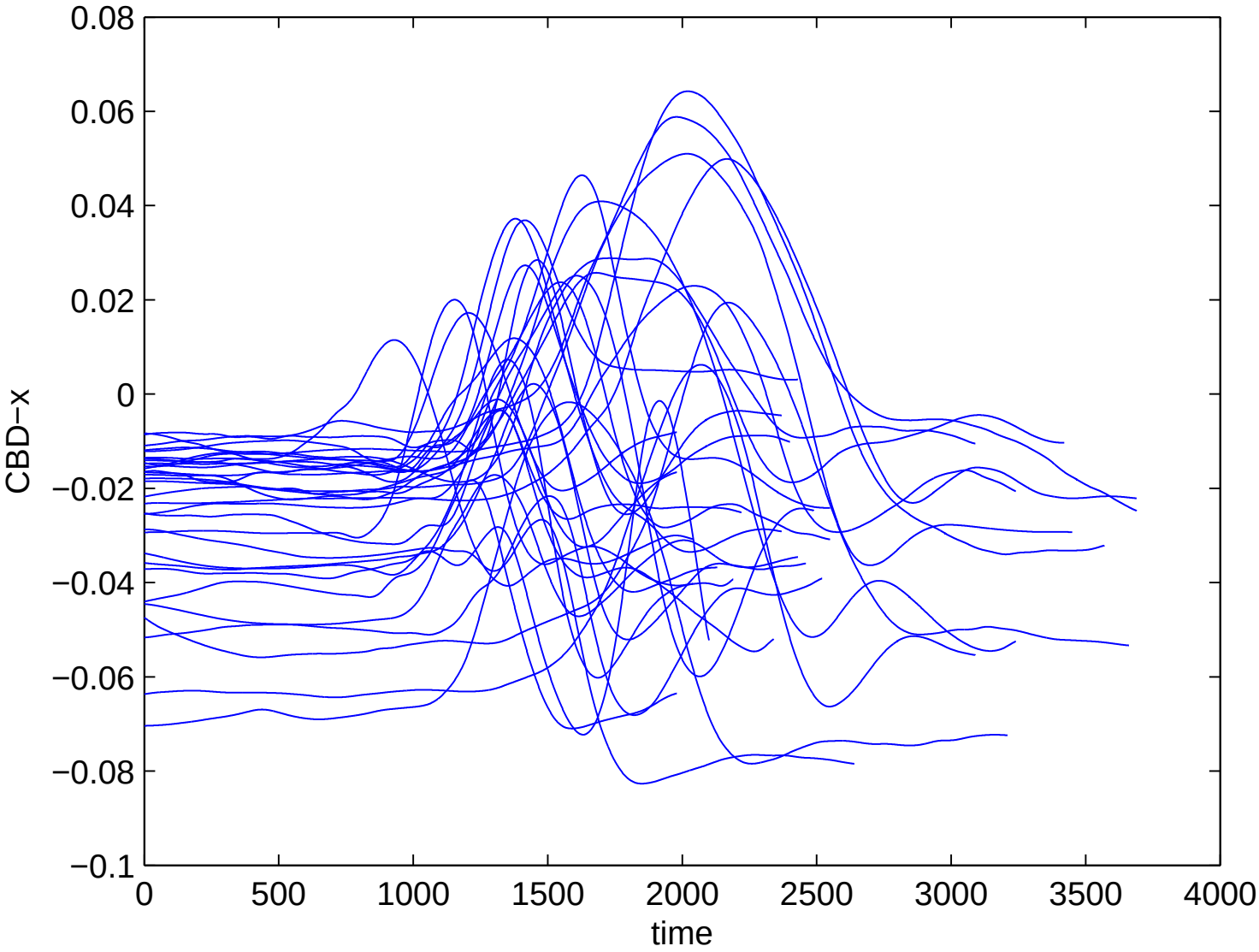
Example 2: Standing-up manoeuvre of unilateral amputee

Example 2: Standing-up manoeuvre of unilateral amputee

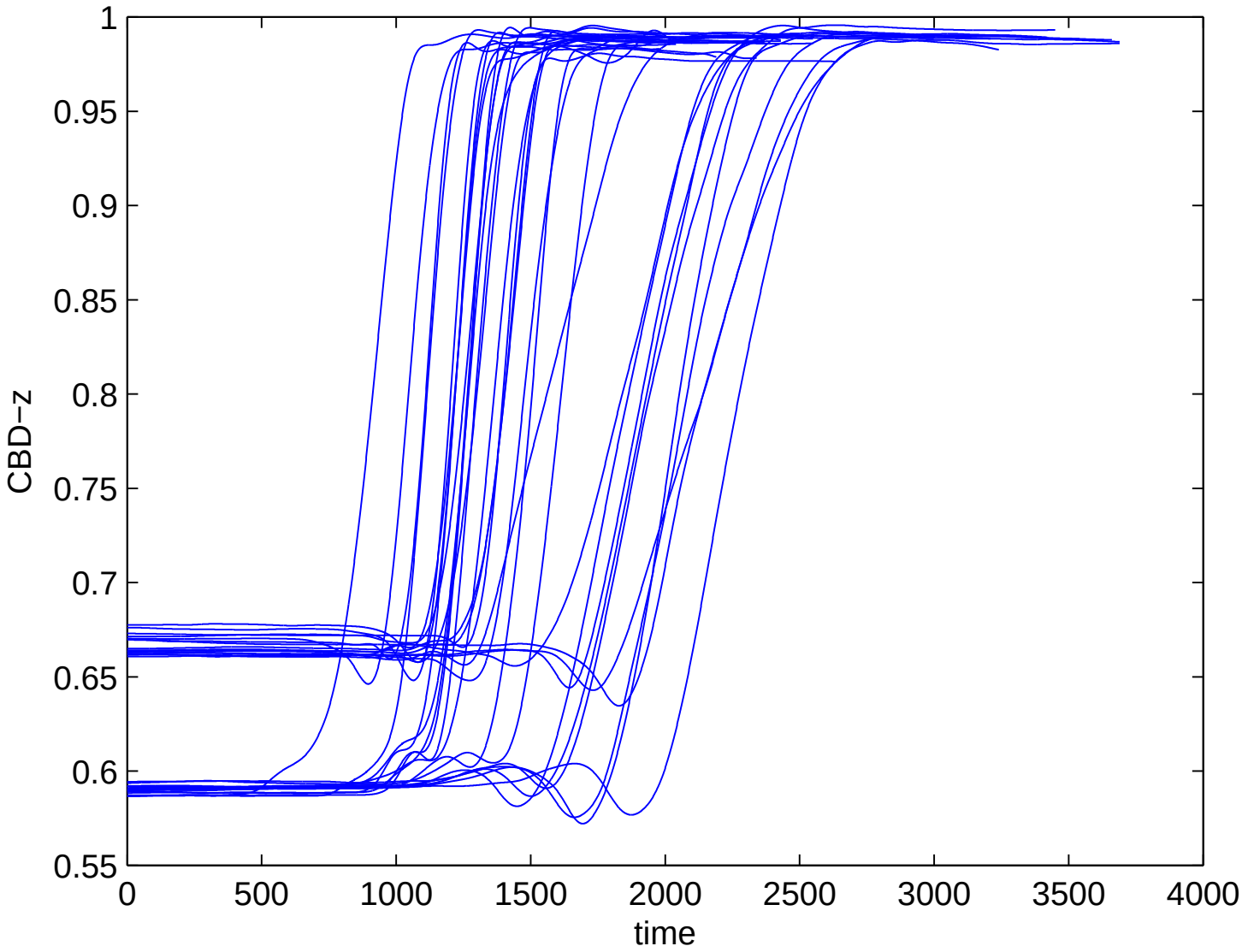
- Output $y(t)$: Body state eg Cbd position or joint angles (e.g. ANtk: trunk angle).
- Input $\mathbf{x}(t)$: measurements of accelerations and angular velocities (30 variables).
- Objective: Use input variables $\mathbf{x}(t)$ to predict $y(t)$.



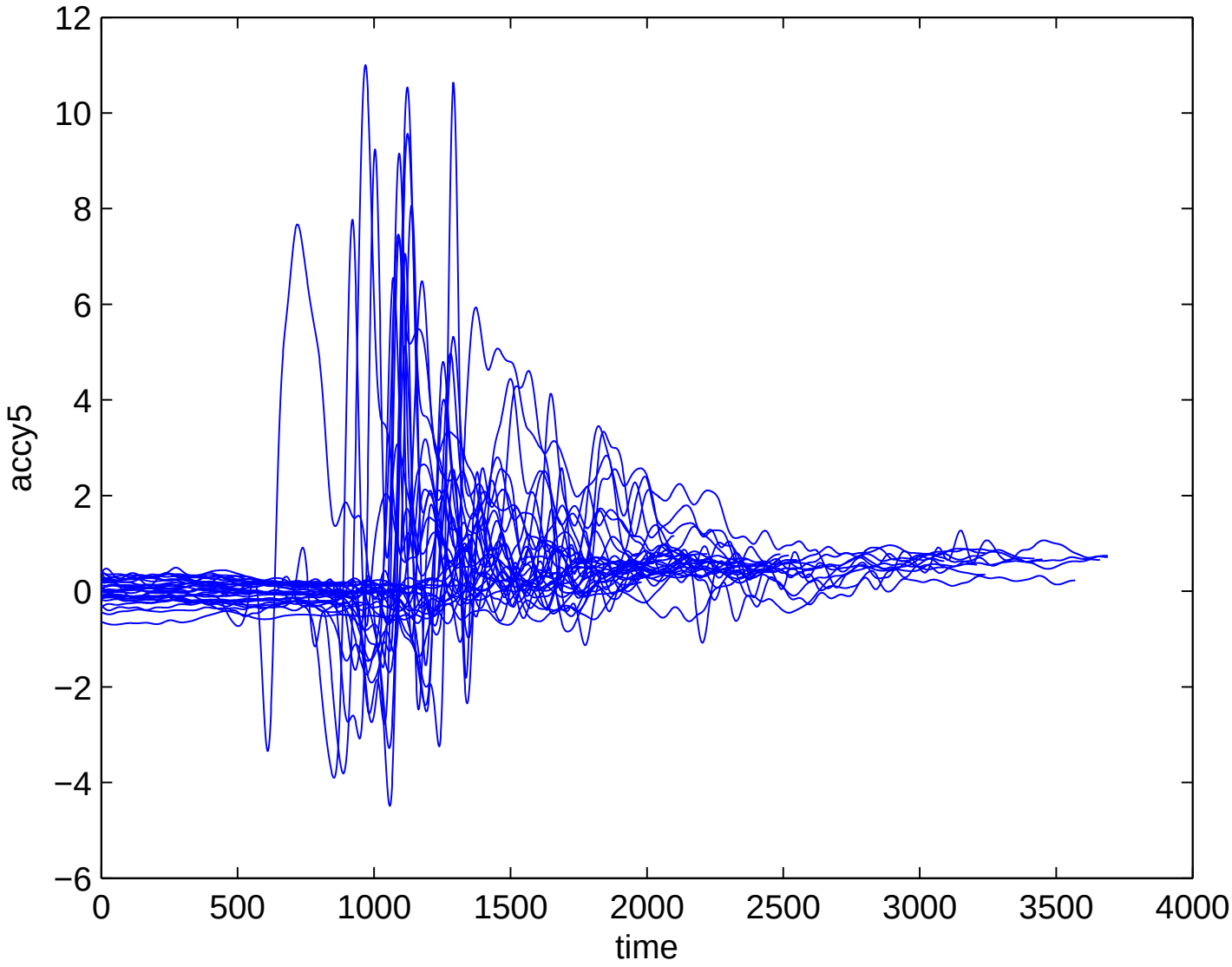
Modelling standing-up manoeuvres of unilateral amputee: Output CBD-x



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

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

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

$$y = f(\mathbf{x}) + \epsilon.$$

- $f(\cdot)$ – mapping $\mathbf{x} \in \mathcal{R}^Q$ 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: $Cov(f, f') = K(\mathbf{x}, \mathbf{x}')$.
- Features
  - It provides a flexible nonlinear model;
  - $\mathbf{x}$ could be large-dimensional;
  - Need to select a parametric covariance kernel, for example the following covariance function (squared exponential + linear).

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

where $\boldsymbol{\theta} = (v_1, w_1, \ldots, w_Q, a_1, \ldots, a_Q)$ – hyper-parameters or tuning parameters.

GPR for a single curve: inference

- How to choose the values of hyper-parameters $\boldsymbol{\theta}$?
  - GCV (only if the dimension of $\boldsymbol{\theta}$ is very small)
  - Empirical Bayesian approach: MAP
  - Fully Bayesian: assume a hyper-prior for $\boldsymbol{\theta}$ and then use MCMC.
- A GPR model is generally formulated as

$$\begin{aligned} y_i|f_i &\overset{\text{ind}}{\sim} g(f_i) \quad \text{and} \\ (f_1, \ldots, f_n) &\sim GP(\mathbf{0}, k(\cdot, \cdot; \boldsymbol{\theta})), \end{aligned}$$

- If

$$y_i|f_i \overset{\text{ind}}{\sim} N(f_i, \sigma_\epsilon^2),$$

  the marginal distribution of y_i is still a normal distribution.
- In general,

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\boldsymbol{\theta}) d\mathbf{f}.$$

- Implementing/computing issues: http://www.gaussianprocess.org/

GPR: asymptotic results – posterior consistency

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 $\epsilon > 0$,

$$\Pi\left\{f \in W_{\epsilon,n}^{C} | \mathcal{D}\right\} \to 0 \ a.s.\ [P_0]. \tag{1}$$

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 $D[P(y_1, \ldots, y_n|f_0)\|P_{bs}(y_1, \ldots, y_n)]$,

$$D[P(y_1, \ldots, y_n|f)\|P_{bs}(y_1, \ldots, y_n)] \leq \frac{1}{2}\|f\|^2_{\mathbf{K}} + \frac{1}{2}\log|\mathbf{I}_n + c\mathbf{K}|, \quad (2)$$

  - $\|f\|_{\mathbf{K}}$ 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)

$$y_m(\mathbf{x}, t) = f_m(\mathbf{x}, t, \mathbf{u}) + \epsilon_m(t), m = 1, \ldots, M$$

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

$$f_m(t) = \mu_m(t) = \mathbf{u}_m{}'\boldsymbol{\beta}(t).$$

- 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 $C(t, t') = Cov(y(t), y(t'))$. Note that t is one-dimensional.
- Gaussian process functional regression (GPFR) model (Shi et al. 2007):

$$f_m(\mathbf{x}, t) = \mu_m(t) + \tau_m(\mathbf{x}).$$

GPFR models for batch data

We define a Gaussian Process Functional Regression model as follows:

$$y_m(\mathbf{x}, t) = \mu_m(t) + \tau_m(\mathbf{x}) + \epsilon_m, \quad m = 1, \ldots, M,$$

where

- $\tau_m(\mathbf{x}) \sim GP\,(0, k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\theta}))$,
  $\mathbf{x}(t)$ is functional, giving the values of input at each data point.
- If we take $\mu_m(t) = \mathbf{u}_m{}'\boldsymbol{\beta}(t)$, then $y_m(t, \mathbf{x})$ can be decomposited by

$$y_m(\mathbf{x}, t) = \mathbf{u}_m{}'\boldsymbol{\beta}(t) + \sum_j \phi_j(\mathbf{x})\gamma_j + \epsilon_m$$

  where $\phi_j(\mathbf{x})$ is the eigenfunction for covariance function $K(\cdot, \cdot)$ and $\gamma_j \sim N(0, \lambda_j)$.

GPFR: estimation

$$y_m(t, \mathbf{x}) = \mathbf{u}_m{}'\beta(t) + \tau_m(\mathbf{x}) + \epsilon_m$$

- $\beta(t)$: B-spline approximation:

$$\beta(t) = \mathbf{B}\mathbf{\Phi}(t).$$

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

GPFR: 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 $\mathbf{x}^* = \mathbf{x}(t^*)$.
- The prediction and the predictive variance of y^* are

$$\begin{aligned} \hat{y}^*_{M+1} &= \hat{\mu}_{M+1}(t^*) + \mathbf{H}'(\mathbf{y}_{M+1} - \hat{\boldsymbol{\mu}}_{M+1}(\mathbf{t})), \\ \hat{\sigma}^{*2}_{M+1} &= \hat{\sigma}^{*2}_{GP}\left(1 + \mathbf{u}'_{M+1}(\mathbf{U}'\mathbf{U})^{-1}\mathbf{u}_{M+1}\right). \end{aligned}$$

GPFR: prediction for a completely new curve

Predict y^* for a new test input $\mathbf{x}^*$ 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 $\hat{\sigma}^{*2}_m$.
  - Assume that

$$P(\text{the new batch belongs to batch } m) = w_m,$$

    - ★ the prediction can be calculated

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

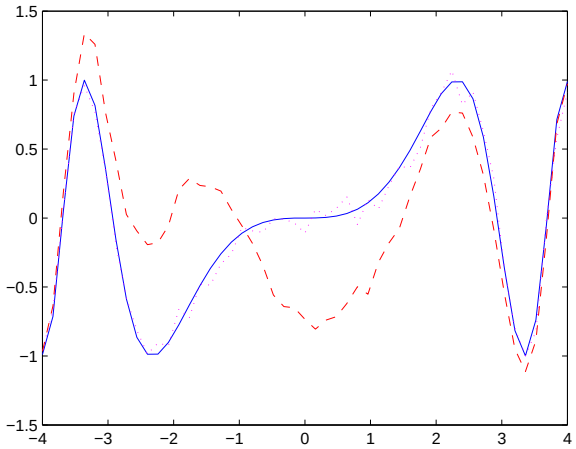
    - ★ The predictive variance is

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

  - w_m may be modelled by a 'spatially indexed' model (Shi and Wang 2008).

GPFR models for batch data

I



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

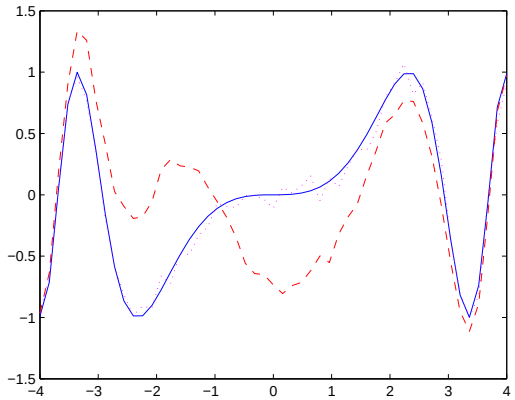
Features

- 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



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

Features

- Usually some data is collected: $\hat{y}^*_{M+1}$ would be

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

- 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.

GPFR: Simulation study for batch data

- The true model used to generate the data is $y_m(x) = u_m + \sin(0.5x)^3 + \tau_m$,
- $x = x_i$ 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 $\{-1, 0, 1\}$.

GPFR: Simulation study for batch data: data

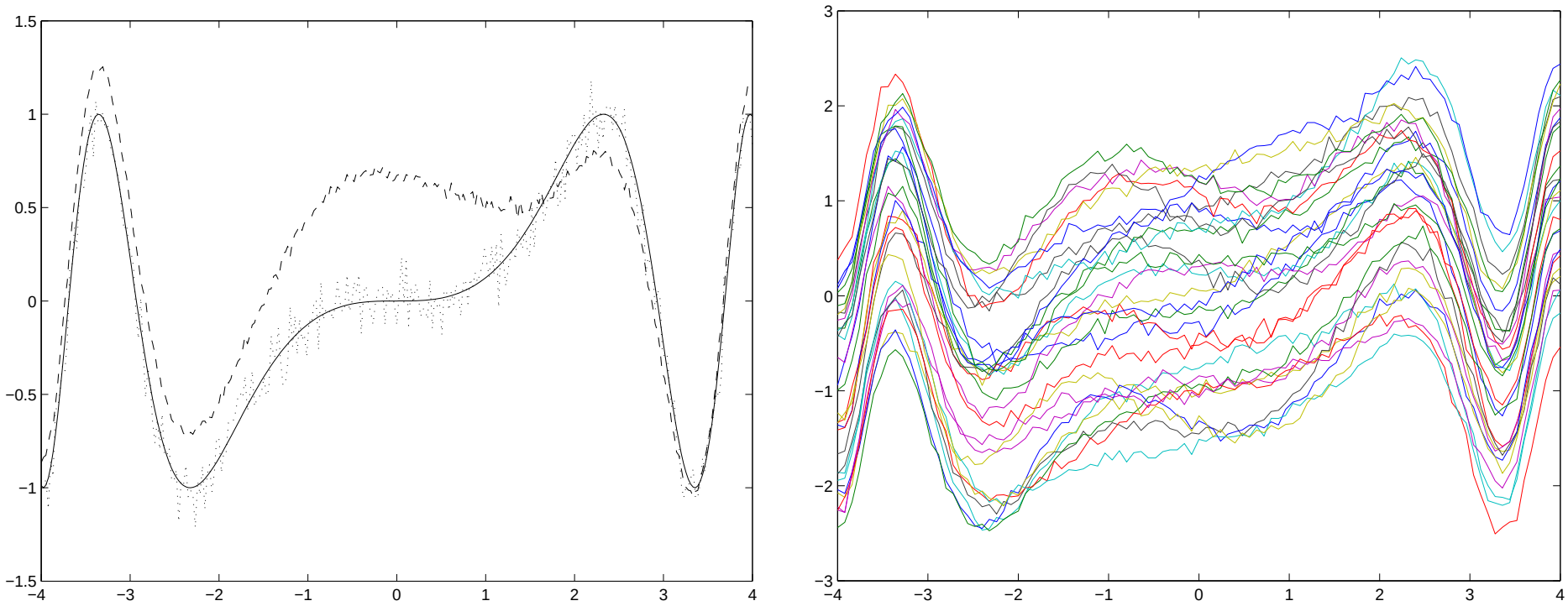


Figure: The sample curves. (a) Solid line—the true mean curve; dotted line—the curve with random errors; dashed line—the curve with errors having GP covariance structure depending on x. (b) 30 sample curves with GP errors.

GPFR: Simulation study–Interpolation

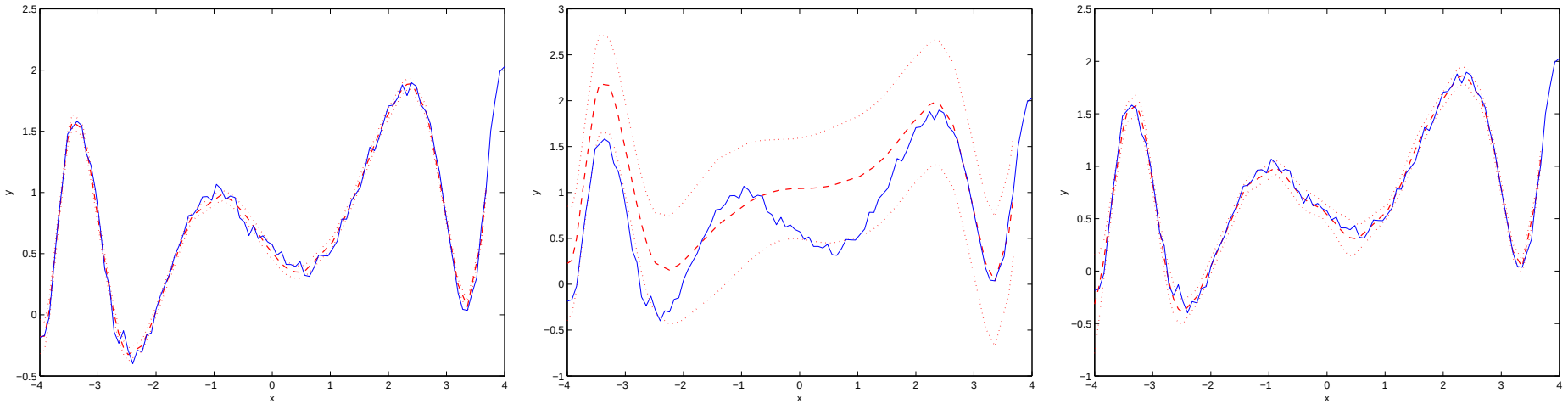


Figure: Training data: 30 curves + 50 data points randomly selected from whole range. Left: GPFR, Middle: Mean model and Right: GPR

- Both GPFR and GPR give very precise results

GPFR: Simulation study–Extrapolation

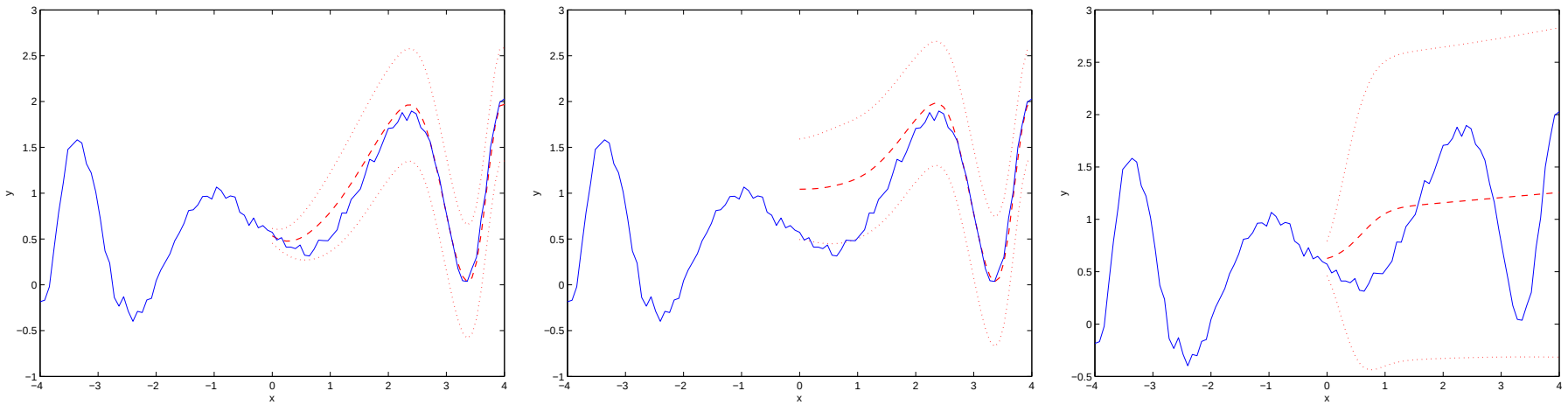


Figure: Training data: 30 curves + 50 data points randomly selected from [-4,0]. Left: GPFR, Middle: Mean model and Right: GPR

- 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

Model	Interpolation		Extrapolation			
	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* 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

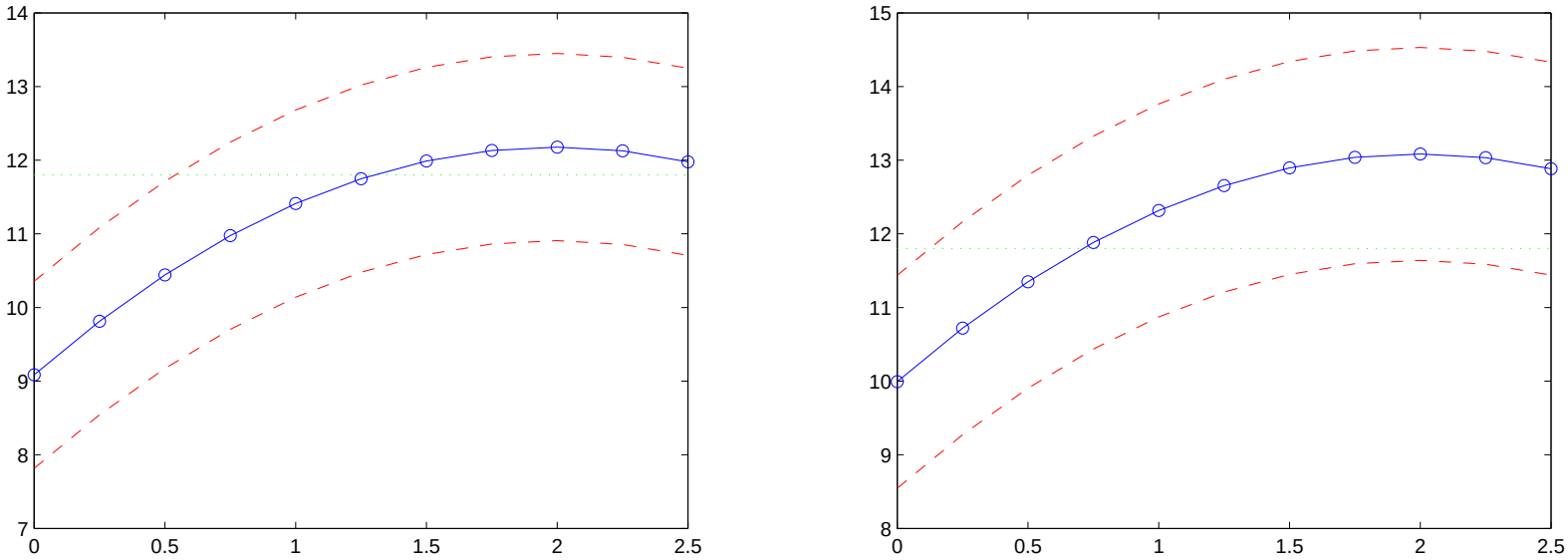


Figure: Renal data: Hb response for different dose level (drug D)

Gaussian process regression model for a single curve

$$y = f(\mathbf{x}) + \epsilon.$$

- $f(\cdot) \sim GPR(\mathbf{x}|k(\cdot,\cdot);$
- $k(\cdot,\cdot;\boldsymbol{\theta})$ covariance kernel/function, depending on $\mathbf{x}$;
- Q – could be large dimensional;
- What if Q is very large, or even $Q >> n$?

GPR: variable selection

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

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

  - MAP: choose $\hat{\boldsymbol{\theta}}$ by maximising $p(\boldsymbol{\theta}|\mathcal{D})$.
- Variable selection when Q is very large, for e.g.

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

  - 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

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

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

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left[-l_n(\boldsymbol{\theta}; \mathcal{D})\right].$$

- Penalized GPR: penalize w_q's by minimizing

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

Penalized GPR: LASSO PGPR

LASSO PGPR: to minimize

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

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

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

Penalized GPR: other penalty functions

To minimize

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

- 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 λ_n and γ 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: λ_n and γ.

Penalized GPR: other penalty functions

To minimize

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

- 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: λ_n and a.

Penalized GPR: comparisons

- SCAD, Adaptive LASSO and Bridge PGPR achieve some nice asypmptotic 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:

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

- 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:

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

  - 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:

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

.

  - 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(prostate-specific antigen). 8 input variables: age, log(cancer volumn) etc.
- Training data: 67 observations. Test data: 30 observations.

Methods Used	Tuning Parameter	RMSE-PredVar	Variables Selected
OLE (Linear)		0.586 (0.184)	All
Ridge (Linear)	$\lambda_n = 1$	0.566 (0.188)	All
Lasso (Linear)	$s = 0.39$	0.499 (0.161)	(1,2,4,5,8)
MLE (GPR)		0.495 (0.073)	All
Ridge (GPR)	$\lambda_n = 1.7$	0.471 (0.061)	All
LASSO (GPR)	$\lambda_n = 0.06$	0.464 (0.057)	(1,2,3,4,5,7,8)
Bridge (GPR)	$\gamma = 0.1, \lambda_n = 0.05$	0.415 (0.025)	(1,2,5)
SCAD (GPR)	$a = 3.7, \lambda_n = 1.8$	0.453 (0.034)	(1,2,4,5,8)
Adap. LASSO (GPR)	$\gamma = 0.8, \lambda_n = 0.18$	0.413 (0.025)	(1,2,5)

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.

	RMSE	Number of Variables Selected
PCR	2.855	All
PLS	2.560	All
QPLS	0.995	All
Neural Network	1.418	All
10-6-1 Network,early stopping	0.65	10
10-3-1 Network, Bayesian	0.52	10
13-X-1 Network, Bayesian ARD	0.36	13
GPR(MLE)	0.89	All
GPR(Ridge)	0.711	All
GPR(LASSO)	0.649	26
GPR(Bridge)	0.432	4
GPR(SCAD)	0.5297	15
GPR(Adaptive LASSO)	0.3901	3

Examples - Paraplegia Standing-up Data

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

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

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 $p^n_{\boldsymbol{\theta}}$ denote the joint probability density of $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ that satisfies some regularity conditions $(C1)$–$(C4)$.*
- *Assume that the penalty function P_{λ_n} satisfies*
  (i) $P_{\lambda_n}(w_q) \geq 0$ and $P_{\lambda_n}(0) = 0$ and
  (ii) $P_{\lambda_n}(w_q^*) \geq P_{\lambda_n}(w_q)$ if $|w_q^*| \geq |w_q|$.
- *There exists a sequence $r_n \to \infty$ so that $\hat{\boldsymbol{\theta}}$ is r_n consistent.*

If b_n converges to 0, then there exists a local minmizer $\hat{\boldsymbol{\theta}}_n$ of $l_p(\boldsymbol{\theta})$ such that $\|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\| = \mathcal{O}_p(r_n^{-1} + a_n)$.

Asymptotic Theories

Let

$$\mathcal{A} = \{q : w_q^{(0)} \neq 0\} \text{ and } \mathcal{B} = \{q : w_q^{(0)} = 0\},$$

Theorem

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

(1) $\liminf_{n\to\infty} \liminf_{\theta\to 0_+} \frac{1}{\lambda_n} \frac{\partial P_{\lambda_n}(\hat{\boldsymbol{\theta}})}{\partial w_q} > 0$

(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.

$$\lim_{n\to\infty} P(\hat{\mathbf{w}}_{\mathcal{B}} = \mathbf{0}) = 1. \tag{3}$$

Penalized Gaussian process classification - PGPC

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

Penalized Gaussian process classification - PGPC

- Marginal density:

$$
\begin{aligned}
p(\mathbf{t}|\mathbf{X}) &= \int p(\mathbf{t},\mathbf{f}|\mathbf{X})d\mathbf{f} \\
&= \int p(\mathbf{t}|\mathbf{X},\mathbf{f})p(\mathbf{f}|\mathbf{X})d\mathbf{f} \\
&= \int \prod_{i=1}^{N} \pi_i^{t_i}(1-\pi_i)^{1-t_i} p(\mathbf{f}|\mathbf{X})d\mathbf{f} \\
&= \int \prod_{i=1}^{N} \left(\frac{1}{1+\exp(-f_i)}\right)^{t_i} \left(1-\frac{1}{1+\exp(-f_i)}\right)^{1-t_i} p(\mathbf{f}|\mathbf{X})d\mathbf{f}
\end{aligned}
$$

Penalized Gaussian process classification - PGPC

- Marginal log-likelihood

$$\begin{aligned} l_n(\boldsymbol{\xi}) &= \log p(\mathbf{t}|\mathbf{X}, \boldsymbol{\xi}) \\ &= \log \int p(\mathbf{t}|\mathbf{X}, \mathbf{f}, \boldsymbol{\xi}) p(\mathbf{f}|\mathbf{X}, \boldsymbol{\xi}) \mathrm{d}\mathbf{f} \\ &= \log \int \exp(\Phi(\mathbf{f})) d\mathbf{f}. \end{aligned}$$

- Laplace approximation

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

  where $K = \nabla\nabla \log p(\mathbf{t}|\mathbf{X}, \mathbf{f}, \boldsymbol{\xi})$

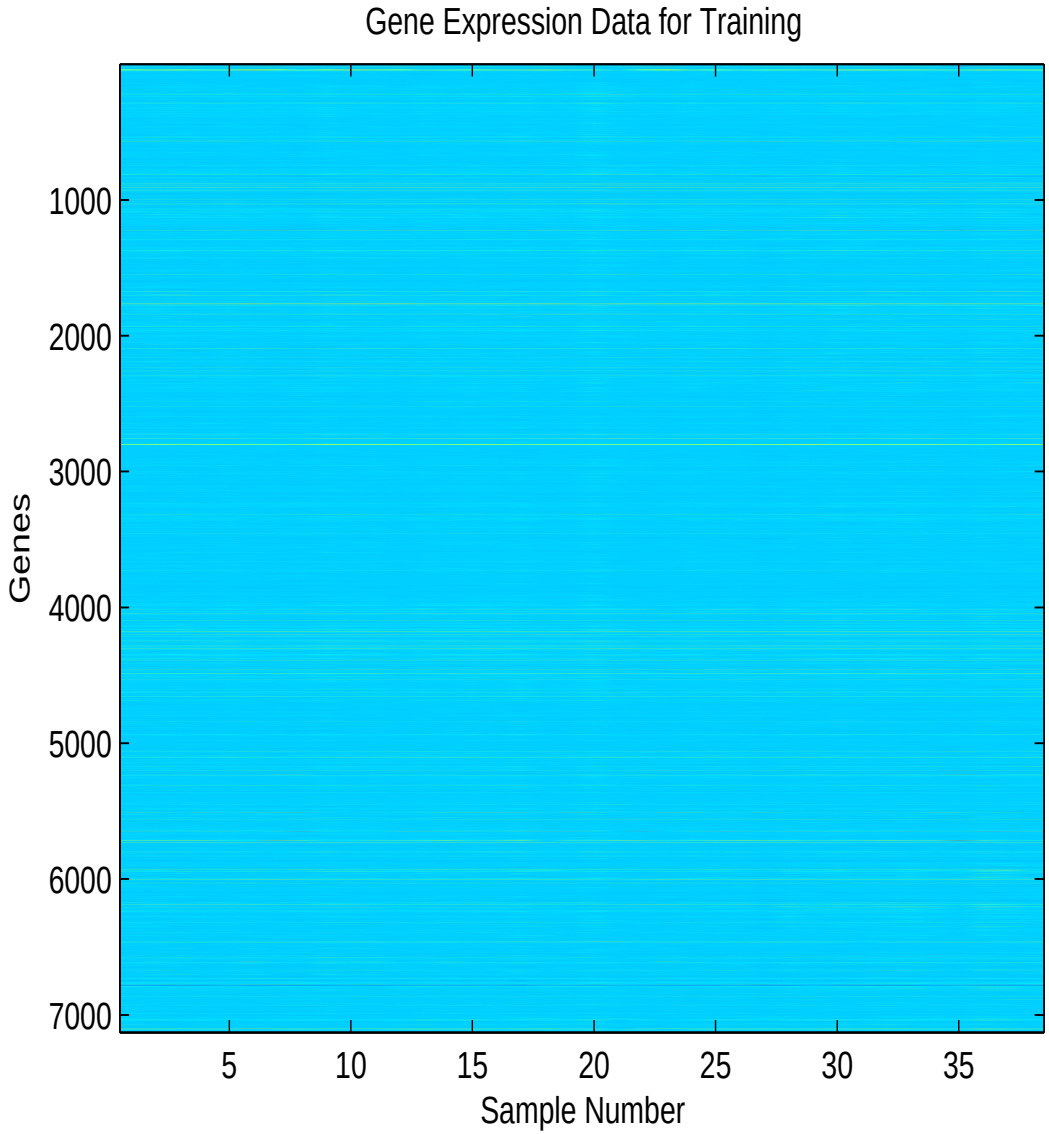
- Penalized likelihood:

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

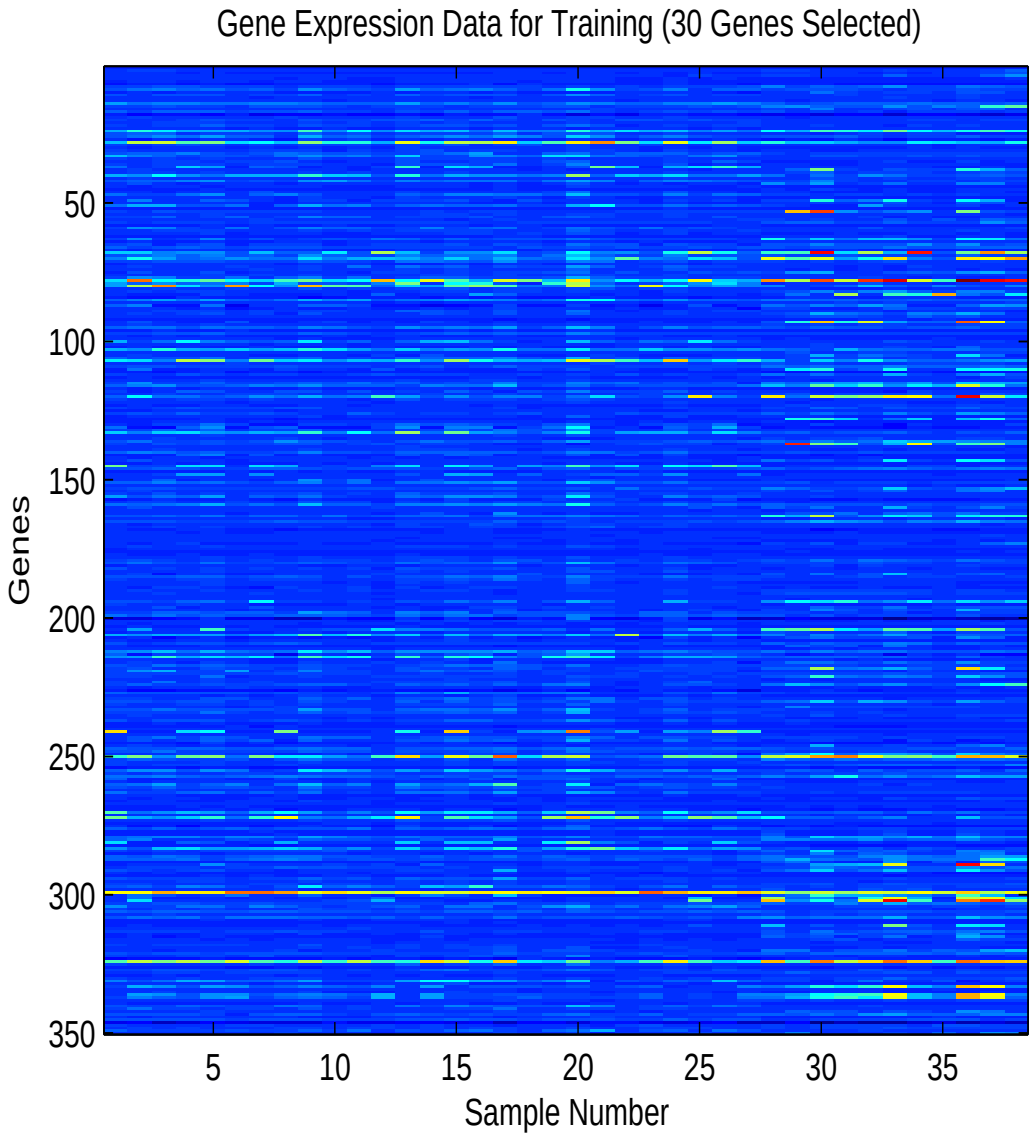
Penalized Gaussian process classification - Leukaemia Cancer Data

- 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.)

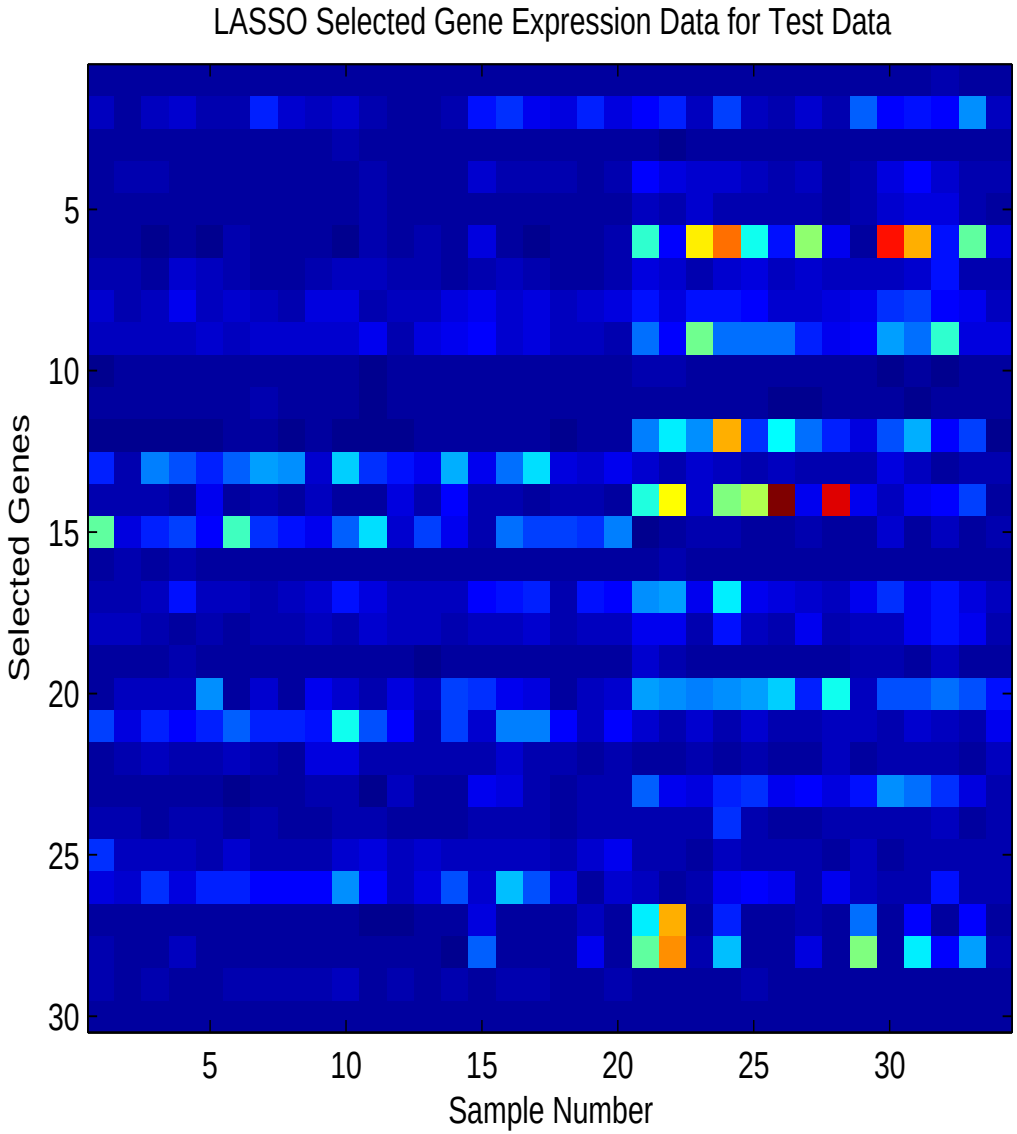
Penalized Gaussian process classification - PGPC



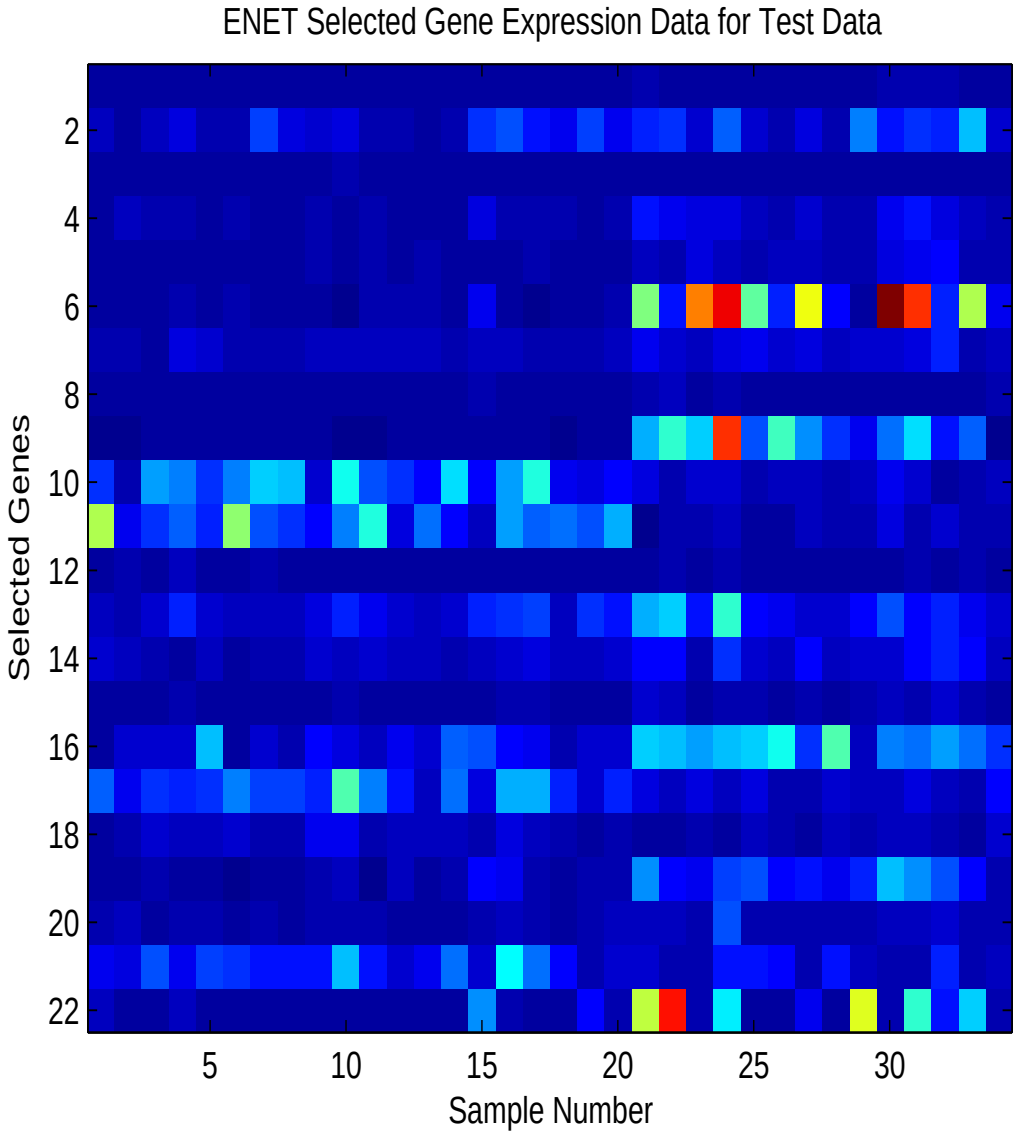
Penalized Gaussian process classification - PGPC



Penalized Gaussian process classification - PGPC



Penalized Gaussian process classification - PGPC



Penalized Gaussian process classification - PGPC

Method	5-fold GCV Error	ClassError	No. of Genes Selected
Golub	3/38	4/34	50
ENET Linear	3/38	0/34	45
LASSO PGPC	4/38	3/34	30
ENET PGPC	2/38	1/34	22

Comments – Generalized GPFR model

Suppose that $z_m(t)$ has a distribution from exponential family, a generalized GPFR model (Wang and Shi, 2012) can be defined as

$$\begin{aligned} \mathrm{E}(z_m(t)|\tau_m(t)) &= h(\mu_m(t) + \tau_m(t)), \\ \tau_m(t) = \tau_m(\mathbf{x}_m(t)) &\sim GPR(0, k(\cdot,\cdot;\boldsymbol{\theta})|\mathbf{x}_m(t)). \end{aligned}$$

Comments – future work

- A functional linear regression model with a scalar response $y \in \mathbb{R}$ is defined by

$$y = \mu + \int_{\mathcal{S}} \beta(s)(x(s) - \mu_x(s))ds + \epsilon,$$

  where $\mu_x(s) = \mathrm{E}(x(s))$ and ϵ is mean-zero noise, $x(s) \in L^2(\mathcal{S})$ where $\mathcal{S}$ is a subset of the real line $\mathbb{R}$.
- In general, a nonlinear functional model is

$$y = g(x_1(s), \ldots, x_p(s), z_1, \ldots, z_q) + \epsilon = g(\mathbf{x}(s), \mathbf{z}) + \epsilon,$$

Comments – future work

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

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

$$g(\mathbf{x}(s)) \sim \text{fGPR}[\mu, k_f(\boldsymbol{\theta})|\mathbf{x}(s)]$$

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

$$\begin{aligned} \text{Cov}[g(\mathbf{x}_i(s)), g(\mathbf{x}_j(s))] &= k_f[\mathbf{x}_i(s), \mathbf{x}_j(s); \boldsymbol{\theta}] \\ &= v_0 \exp\left\{-\frac{1}{2}\sum_{k=1}^{p} w_k ||x_{ik}(s) - x_{jk}(s)||_f^2\right\}. \end{aligned}$$

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

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

$$k[(\mathbf{x}_i(s), \mathbf{z}_i), (\mathbf{x}_j(s), \mathbf{z}_j)] = k_f[\mathbf{x}_i(s), \mathbf{x}_j(s)] \cdot k(\mathbf{z}_i, \mathbf{z}_j).$$

Comments

- 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.

Shi, J. Q and Choi, T. (2011) *Gaussian Process Regression Analysis for Functional Data*. Chapman & Hall/CRC.

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

Thank you ...