New techniques for functional data analysis

Matthew Avery

North Carolina State University

3/13/12

1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

General model for functional regression

A general model for functional linear regression is

$$Y_i = \beta_0 + \int_{\mathcal{T}} X_i(t)\beta(t)dt + \epsilon_i, \quad i = 1, \dots, n.$$

- Y_i is a scalar response variable with functional predictor $X_i(t)$
- $\beta(t)$ is the coefficient function
- $\mathcal{T}$ is the domain for the functional predictor, often scaled to the interval $[0, 1]$
- ϵ_i is the random error

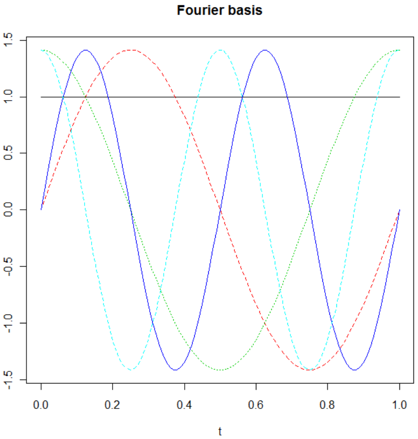
Basis function method for functional linear regression

- One standard method for analyzing functional data is to decompose $\beta(t)$ using some basis, $\mathbf{B}(t) = [b_1(t), b_2(t), \ldots, b_p(t)]^T$

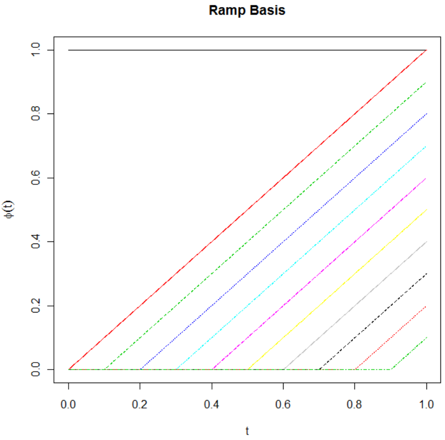
$$\beta(t) = \mathbf{B}(t)^T \boldsymbol{\eta} \qquad x_{ij} = \int_{\mathcal{T}} X_i(t) b_j(t) dt \qquad Y_i = \beta_0 + \mathbf{x}_i \boldsymbol{\eta} + \epsilon_i$$

- $\mathbf{B}(t)$ is a vector of basis functions, $\{b_j\}$
- $\mathbf{x}_i$ is the ith row of the design matrix created from $\mathbf{B}(t)$
- $\boldsymbol{\eta}$ is a vector of regression coefficients corresponding to $\mathbf{B}(t)$

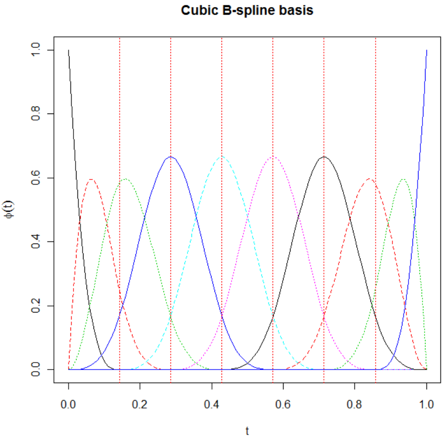
Fourier basis



Ramp basis



Cubic b-spline basis



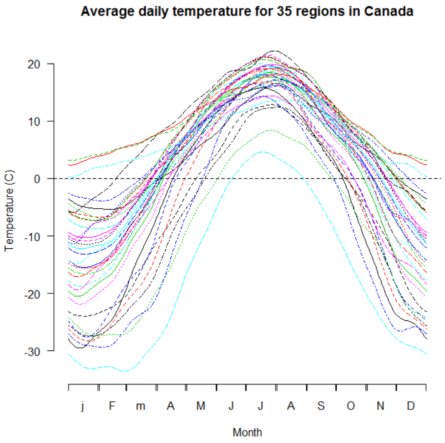
Sparse regression with functional data

It is possible that only a proper subset of $\mathcal{T}$ is useful for modeling Y with $X(t)$. Then for some subset, $\mathcal{T}' \subset \mathcal{T}$, $\beta(t) = 0 \ \ \forall t \in \mathcal{T}'$, where $\mathcal{T}'$ may be broken into one or a few contiguous subintervals.

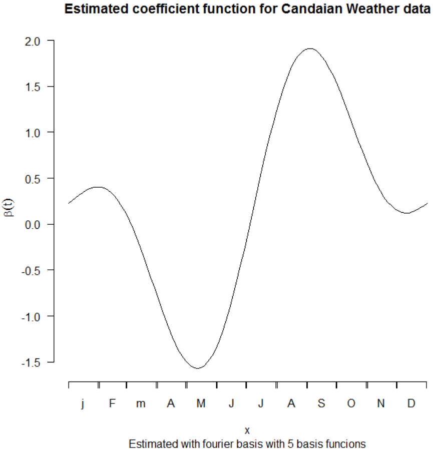
- Want to identify $\mathcal{T}'$
- Find estimate of $\beta(t)$ such that $\widehat{\beta}(t) = 0 \ \ \forall t \in \mathcal{T}'$
- Rather than sparse parameter space, we identify "sparse" region on domain that is relevant for modeling Y

Canadian Weather Data

Canadian Weather data from Ramsay and Silverman [2005]

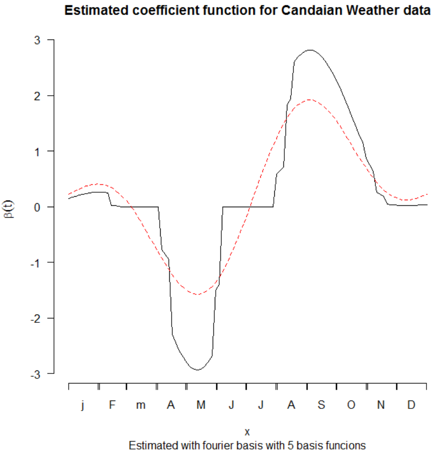


Canadian Weather Data



Objective: Identify important months for predicting rainfall

Sparse fit for Canadian Weather Data

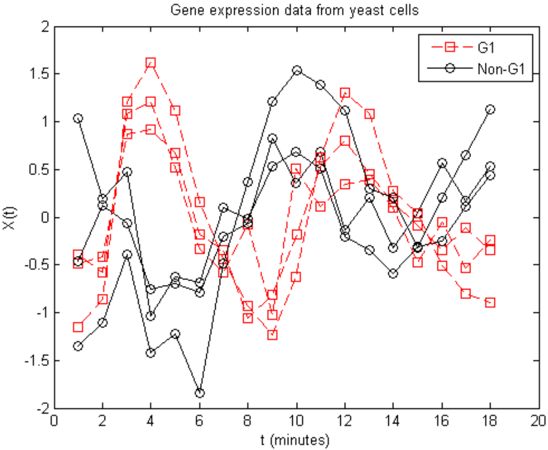


Classification for functional data

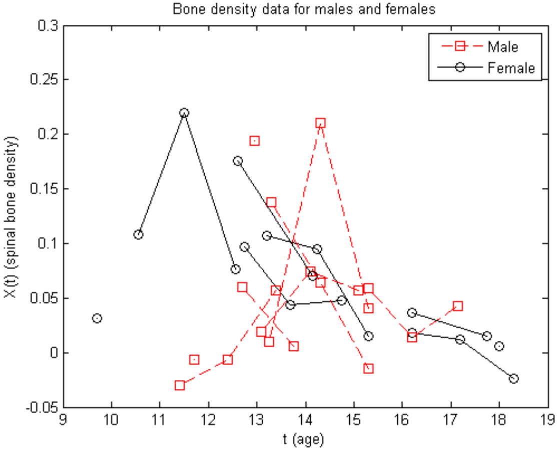
Let $Y \in \{1, 2, ..., K\}$ be a categorical response variable with associated functional predictor, $X(t)$. We use $X(t)$ to classify Y.

- Often, $K = 2$. For example, Leng and Müller [2006] discuss the problem of identifying the cellular process certain genes are involved in.
- Data is densely sampled at regular intervals.
- Spinal bone mineral density data has been discussed by James et al. [2000], where response is gender.
- Sparsely sampled at irregular time intervals
- When $K > 2$, the problem becomes more complex. James and Hastie [2001] use bone mineral density data with ethnicity as response.

Gene expression in yeast



Bone density in adolescents



Outline

1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

Model Assumptions

The goal is to identify regions of sparsity in $\beta(t)$ while accurately modeling the data

- Use basis function approach discussed above
- Assume $\beta(t) = 0$ over some region, $\mathcal{T}'$
- $\mathcal{T}'$ can be broken into a few, contiguous subsets, $\mathcal{T}'_m$
- Want to find an estimator, $\widehat{\beta}(t)$ that correctly identifies these sparse regions and also gives a consistent estimate for non-zero regions.

Existing Methods

One method in the literature for comparison with our own is Functional Linear Regression That's Interpretable (FLiRTI). James et al. [2009]

- Objective is to create interpretable estimates for $\beta(t)$
- Restricts certain derivatives (0th, 1st, 2nd, etc.) of $\widehat{\beta}(t)$ to be sparse
- Fits restricted model using LASSO

Valdés-Sosa et al. [2005] explores a variety of variable selection methods for modeling brain connectivity with longitudinal brain image data

Outline

1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

General approach for sparsity

$$L_{\lambda_1}(\mathbf{y}, \mathbf{X}, \boldsymbol{\eta}) = .5(\mathbf{y} - \mathbf{X}\boldsymbol{\eta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\eta}) + \lambda_1 \sum_{j=1}^{p} q(|\eta_j|). \qquad (1)$$

The choice of basis, $\mathbf{B}(t) = [b_1(t), b_2(t), \ldots, b_p(t)]^T$, is important, since not all basis functions allow us to easily select a sparse subset of $\mathcal{T}$ through penalized regression

- Low-order B-splines are a natural option, since they are identically 0 over a large subset
- LASSO is a common method for finding sparse fits in scalar regression
- The Fused LASSO takes advantage of sequential structure of B-splines

Fused LASSO

The LASSO model [Tibshirani, 1994]

$$L_{\lambda_1}(\mathbf{y}, \mathbf{X}, \eta) = .5(\mathbf{y} - \mathbf{X}\eta)^T(\mathbf{y} - \mathbf{X}\eta) + \lambda_1 \sum_{j=1}^{p} |\eta_j|. \tag{2}$$

- Will result in sparsity but sparse regions won't necessarily be adjacent or contiguous

Fused LASSO

The LASSO model [Tibshirani, 1994]

$$L_{\lambda_1}(\mathbf{y}, \mathbf{X}, \boldsymbol{\eta}) = .5(\mathbf{y} - \mathbf{X}\boldsymbol{\eta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\eta}) + \lambda_1 \sum_{j=1}^{p} |\eta_j|. \tag{2}$$

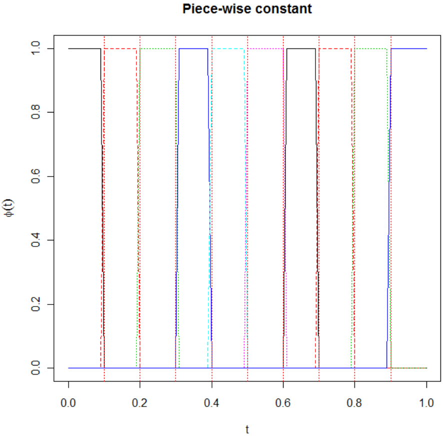
- Will result in sparsity but sparse regions won't necessarily be adjacent or contiguous

The Fused LASSO model [Tibshirani et al., 2005]

$$L_{\lambda_1,\lambda_2}(\mathbf{y}, \mathbf{X}, \boldsymbol{\eta}) = .5(\mathbf{y} - \mathbf{X}\boldsymbol{\eta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\eta}) + \lambda_1 \sum_{j=1}^{p} |\eta_j| + \lambda_2 \sum_{j=2}^{p} |\eta_j - \eta_{j-1}|. \tag{3}$$

- Ensures interpretable, contiguous zero-regions
- Quadratic programming problem allows for quick computation

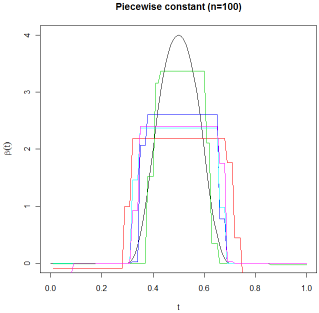
Piecewise constant basis



Piecewise constant basis with Fused LASSO

$$\widehat{\beta}(t) = \sum_{j=1}^{J} \widehat{\eta}_j \mathbf{1}_{t \in [t^*_{j-1}, t^*_j]} \tag{4}$$

where $\widehat{\boldsymbol{\eta}}$ solves Equation 3



Alternative basis for a better fit

A more flexible basis may provide a better fit but won't necessarily result in sparse regions when combined with the Fused LASSO

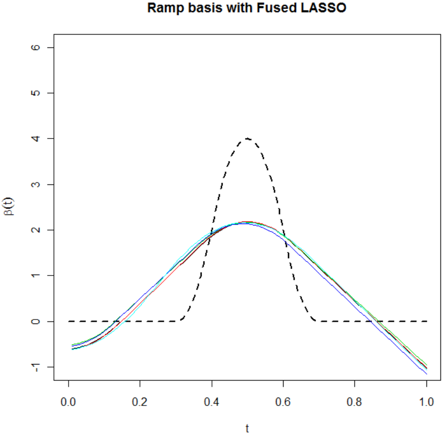
- "Ramp" basis: linear with new slope at each break point

$$b_j(t) = t\mathbf{1}_{t>t_j}$$

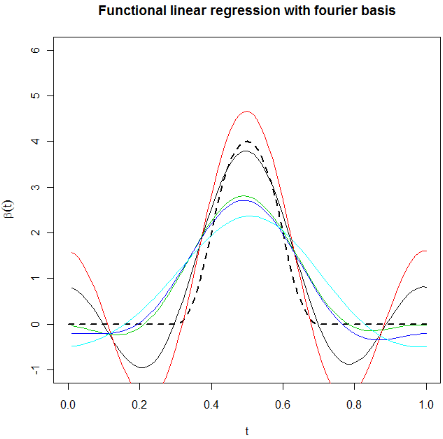
- Fourier basis without sparsity constraint

$$b_{2j}(t) = sin(j\pi t) \quad b_{2j-1}(t) = cos(j\pi t)$$

"Ramp" basis with Fused LASSO



Fourier basis fit (no sparsity constraint)



Two-stage approach

We can combine the sparsity properties of the Fused LASSO/piecewise constant basis with the better fits from alternative methods via a two-stage approach

- The first stage aims to model the coefficient function accurately using one of the methods from the previous slide (or any method that produces a good estimate for $\beta(t)$).
- The second stage adds sparsity

Second Stage

Let $\widetilde{\beta}(t)$ be first stage estimate for $\beta(t)$. Define

$$x^*_{i,j} = \int X_i(t)\widetilde{\beta}(t)b_j(t)dt. \tag{5}$$

Now minimize

$$L_{\lambda_1,\lambda_2}(\mathbf{y}, \mathbf{X}^*, \theta) = .5(\mathbf{y} - \mathbf{X}^*\theta)^T(\mathbf{y} - \mathbf{X}^*\theta) + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \lambda_2 \sum_{j=2}^{p} |\theta_j - \theta_{(j-1)}|. \tag{6}$$

where $\{b_j(t)\}$ is the piecewise constant basis and the penalty terms λ_1 and λ_2 are chosen via cross-validation.

Second Stage

Then

$$\widehat{\beta}(t) = \widetilde{\beta}(t) * \widetilde{\theta}(t), \tag{7}$$

where $\widetilde{\theta}(t) = \sum_{j=1}^{p} \widetilde{\theta}_j b_j(t)$.

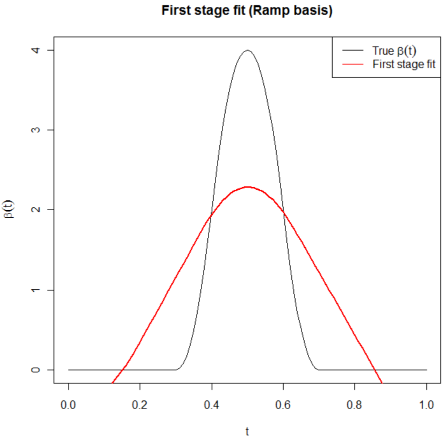
- $\mathbf{X}^*$ contains all information used to model y in our first stage
- First stage fit projected onto piecewise constant basis
- Second stage fit with Fused LASSO identifies which regions, $\mathcal{T}_j^* = [t_{j-1}^*, t_j^*]$, are not relevant for modeling Y and sets $\theta_j = 0$

Summary of algorithm for sparse estimate of $\beta(t)$

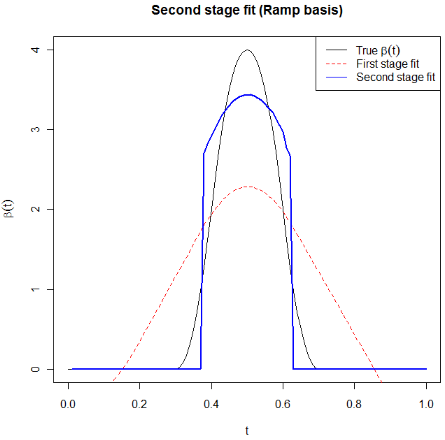
To summarize the above,

- Step 1: Obtain good estimate for $\beta(t)$ (can be done using any method desired, sparsity it not a concern). Denote this $\widetilde{\beta}(t)$
- Step 2: Create $\mathbf{X}^*$ and estimate sparsity function, $\widetilde{\theta}(t)$
- Step 3: Combine $\widetilde{\theta}(t)$ and $\widetilde{\beta}(t)$ to obtain final, sparse estimate, $\widehat{\beta}(t)$.

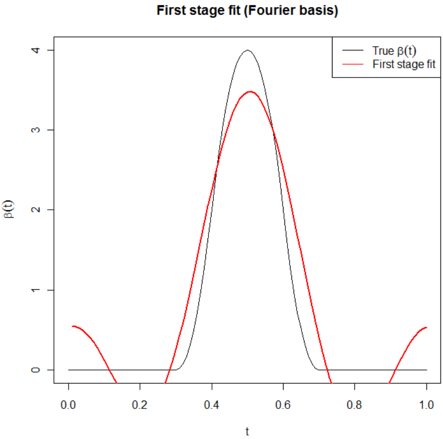
Stage 1 Ramp basis



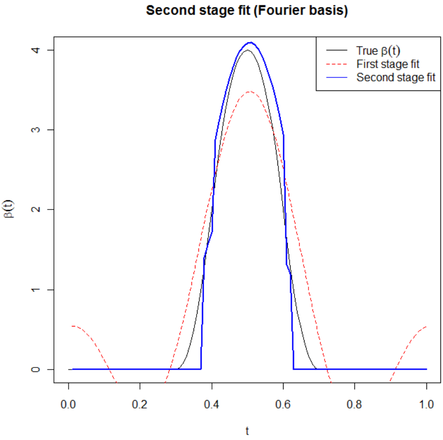
Stage 2 Ramp basis



Stage 1 Fourier basis



Stage 2 Fourier basis



Assessing quality of fit

To assess how well our model is performing, we use a separate test data set to compare our estimate is to the true coefficient function.

- First, we look at how close our estimated coefficient function is to the true coefficient function. Denote
$$MISE_1 = \int_{\mathcal{T}} \left(\beta(t) - \widehat{\beta}(t) \right)^2 dt \tag{8}$$
- The next value is similar to $MISE_1$ but weights by the data and is closer to a measure of prediction error.
$$MISE_2 = N^{-1} \sum_{i=1}^{N} \left(\int_{\mathcal{T}} \beta(t) X_i(t) dt - \int_{\mathcal{T}} \widehat{\beta}(t) X_i(t) dt \right)^2 . \tag{9}$$

Model selection error

To assess how well our model identifies sparse regions, we use the Type I and Type II error rates.

- Sparsity of the model, assessed by selection error

$$\text{Type I Error} = p_1^{-1} \sum_{i \in \Omega_1} (1 - \mathbf{1}_{\widehat{\beta}(t_i)=0}) \tag{10}$$

$$\text{Type II Error} = p_2^{-1} \sum_{i \in \Omega_2} (1 - \mathbf{1}_{\widehat{\beta}(t_i)\neq 0}), \tag{11}$$

where Ω_1 and Ω_2 are the set of indices where $\beta(t_i) = 0$ and $\beta(t_i) \neq 0$ respectively, and p_1 and p_2 are the total time points observed in each of these groups.

Model Tuning

Three parameters in the model require tuning. : The two penalty terms, λ_1 and λ_2, as well as the number of basis functions, p, for sparsity step.

- The two penalty terms are tuned simultaneously using *optim* in *R*.
- Both cross-validation and EBIC performed well in simulation. ($EBIC = nlog(\widehat{\sigma}^2) + (log(n) + log(p))df$) For the examples discussed later, we tuned using EBIC.
- For high values of p, there was some instability, and computation times increased dramatically
- $p = 35$ performed best in simulation, so we use that for our examples

Computational issues

The Fused LASSO model can be fit using quadratic programming, which solves

$$\arg\min(.5d^T Db - d^T b) \quad s.t.\ A^T b \geq b_0. \tag{12}$$

We can put the Fused LASSO in this form:

$$\mathbf{D} = \begin{pmatrix} \mathbf{X}^T\mathbf{X} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; d^T = \begin{pmatrix} \mathbf{Y}^T\mathbf{X} \\ -\lambda_1\mathbf{1} \\ -\lambda_2\mathbf{1} \end{pmatrix}; \mathbf{A}^T = \begin{pmatrix} -\mathbf{I} & \mathbf{I} & 0 \\ \mathbf{I} & \mathbf{I} & 0 \\ \mathbf{Q} & 0 & \mathbf{I} \\ -\mathbf{Q} & 0 & \mathbf{I} \end{pmatrix} \tag{13}$$

where

$$b = \mathbf{0}, \qquad \mathbf{Q} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & \dots & 1 & -1 \end{pmatrix}. \tag{14}$$

Outline

1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

Simulation Specifications

Data was simulated using the following sinusoidal basis

$$\phi_0(t) = 1 \tag{15}$$
$$\phi_1(t) = \sin(\pi t) \tag{16}$$
$$\phi_2(t) = \cos(\pi t) \tag{17}$$
$$\phi_3(t) = \sin(2\pi t) \tag{18}$$
$$\phi_4(t) = \cos(2\pi t), \tag{19}$$

where $X_i(t) = \sum_{k=0}^{4} \xi_{ik}\phi_k(t) + \epsilon_i(t)$. The ξ_{ik} are normally distributed with mean 0 and variance 16, and the error process has variance 1.

Plot of data

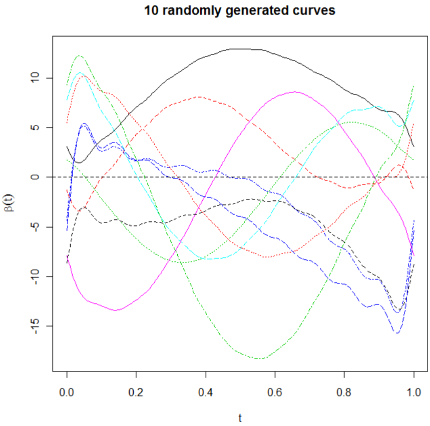
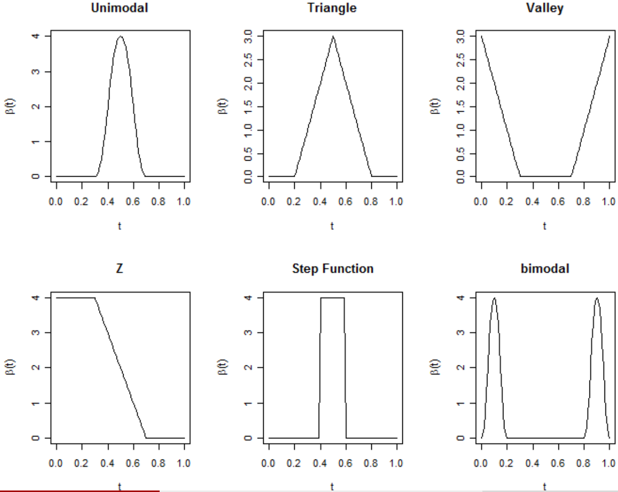


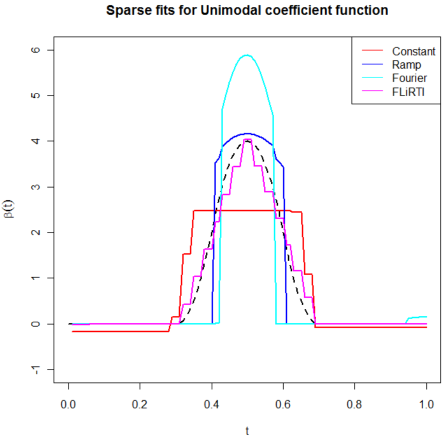
Figure: Ten randomly generated curves with observation error

Coefficient functions

We tested the performance of our method against a variety of coefficient functions



Sparse fits for unimodal coefficient function



Results for Unimodal coefficient function

Table: Simulation results for all methods with $\beta(t)$ Unimodal, $n = 50$, $p = 35$

Coefficient Function	MISE 1	MISE 2	Type I error	Type II error
Constant	0.51	0.09	0.12	0.12
Ramp 1	0.99	2.03	0.63	0.00
Ramp 2	0.33	0.08	0.16	0.18
Fourier 1	0.80	0.11	0.43	0.02
Fourier 2	0.61	0.09	0.21	0.19
Flrt 1	0.85	0.11	0.08	0.35
Flrt 2	0.59	0.11	0.18	0.26
Median SE	0.048	0.008	0.019	0.016

Results for Unimodal coefficient function

Table: Simulation results unimodal coefficient function for different values of n with EBIC tuning. ($p = 35$)

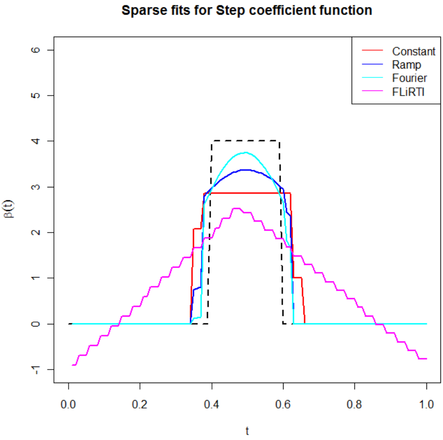
Error Type	Fit	n=50	100	200	500
MISE 1	Piecewise Constant	0.51	0.43	0.41	0.30
	Ramp	0.33	0.31	0.24	0.20
	Fourier	0.61	0.38	0.31	0.16
	FLiRTI ($d = 2$)	0.59	0.41	0.27	0.21
	Median SE	0.044	0.037	0.035	0.023
MISE 2	Piecewise Constant	0.09	0.05	0.02	0.01
	Ramp	0.08	0.04	0.02	0.01
	Fourier	0.09	0.05	0.02	0.01
	FLiRTI ($d = 2$)	0.11	0.05	0.03	0.01
	Median SE	0.007	0.004	0.002	0.001

Results for Unimodal coefficient function

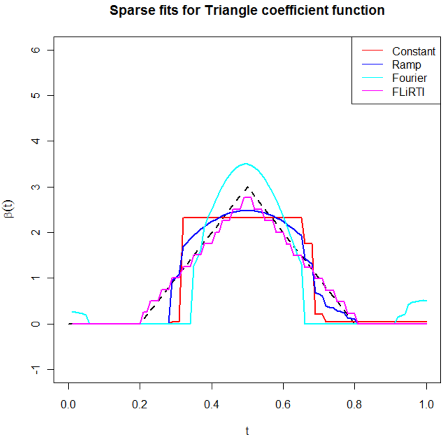
Table: Simulation results unimodal coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	500
Type I error	Piecewise Constant	0.12	0.07	0.07	0.07
	Ramp	0.16	0.19	0.18	0.17
	Fourier	0.21	0.21	0.16	0.14
	FLiRTI ($d = 2$)	0.18	0.11	0.11	0.12
	Median SE	0.020	0.017	0.018	0.016
Type II error	Piecewise Constant	0.12	0.16	0.15	0.19
	Ramp	0.18	0.20	0.23	0.22
	Fourier	0.19	0.17	0.22	0.21
	FLiRTI ($d = 2$)	0.26	0.22	0.18	0.17
	Median SE	0.018	0.017	0.015	0.015

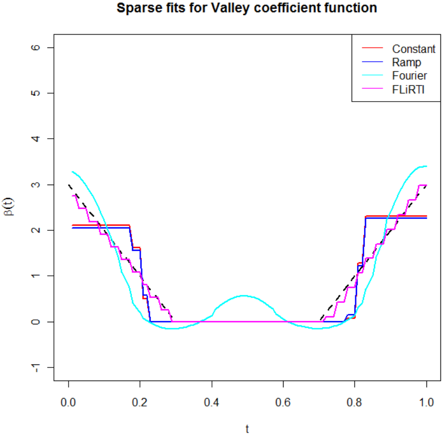
Sparse fits for step coefficient function



Sparse fits for triangle coefficient function



Sparse fits for valley coefficient function



Results for Valley coefficient function

Table: Simulation results for Valley coefficient function for different values of n with EBIC tuning. $(p = 35)$

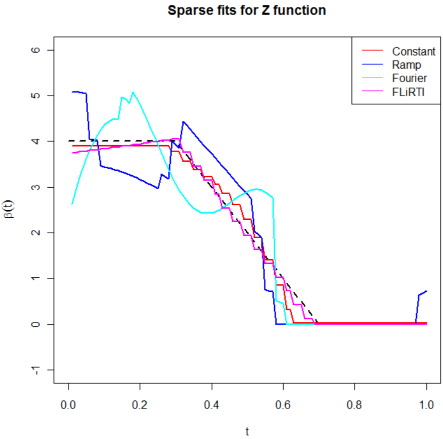
Error Type	Fit	n=50	100	200	500
MISE 1	Piecewise Constant	0.19	0.18	0.17	0.17
	Ramp	0.17	0.15	0.12	0.11
	Fourier	0.50	0.25	0.15	0.08
	FLiRTI $(d = 2)$	0.34	0.26	0.32	0.22
	Median SE	0.036	0.024	0.012	0.008
MISE 2	Piecewise Constant	0.11	0.05	0.02	0.01
	Ramp	0.42	0.07	0.03	0.01
	Fourier	0.11	0.06	0.03	0.01
	FLiRTI $(d = 2)$	0.12	0.06	0.03	0.01
	Median SE	0.009	0.005	0.002	0.001

Results for Valley coefficient function

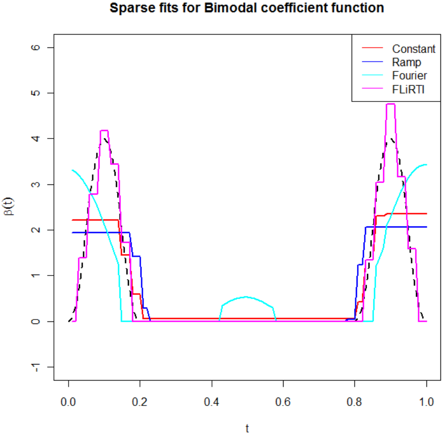
Table: Simulation results Valley coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	500
	Piecewise Constant	0.15	0.27	0.42	0.70
Type I	Ramp	0.09	0.23	0.27	0.44
error	Fourier	0.29	0.31	0.28	0.30
	FLiRTI ($d = 2$)	0.07	0.05	0.03	0.03
	Median SE	0.024	0.033	0.034	0.036
	Piecewise Constant	0.19	0.17	0.12	0.06
Type II	Ramp	0.20	0.15	0.13	0.08
error	Fourier	0.22	0.22	0.20	0.16
	FLiRTI ($d = 2$)	0.15	0.12	0.12	0.10
	Median SE	0.015	0.015	0.014	0.010

Sparse fits for Z coefficient function



Sparse fits for bimodal coefficient function



Results for Bimodal coefficient function

Table: Simulation results for bimodal coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	1000
	Piecewise Constant	0.88	0.85	0.81	0.84
MISE 1	Ramp	0.94	0.86	0.84	0.78
	Fourier	1.42	1.04	0.92	0.89
	FLiRTI ($d = 2$)	0.86	0.75	0.63	0.61
	Median SE	0.038	0.033	0.023	0.021
	Piecewise Constant	0.11	0.05	0.02	0.01
MISE 2	Ramp	0.14	0.08	0.03	0.01
	Fourier	0.12	0.06	0.02	0.01
	FLiRTI ($d = 2$)	0.12	0.06	0.03	0.01
	Median SE	0.009	0.005	0.002	0.001

Results for Bimodal coefficient function

Table: Simulation results bimodal coefficient function for different values of n with EBIC tuning. ($p = 35$)

Error Type	Fit	n=50	100	200	500
Type I error	Piecewise Constant	0.14	0.16	0.13	0.18
	Ramp	0.18	0.21	0.16	0.15
	Fourier	0.28	0.23	0.27	0.23
	FLiRTI ($d = 2$)	0.22	0.19	0.13	0.09
	Median SE	0.018	0.024	0.022	0.021
Type II error	Piecewise Constant	0.01	0.01	0.00	0.00
	Ramp	0.01	0.01	0.01	0.02
	Fourier	0.06	0.03	0.01	0.00
	FLiRTI ($d = 2$)	0.06	0.09	0.14	0.21
	Median SE	0.007	0.005	0.004	0.005

Outline

1 Introduction

2 Sparse functional linear regression
- Model assumptions and existing methods
- New Methodology
- Simulations
- Real Data

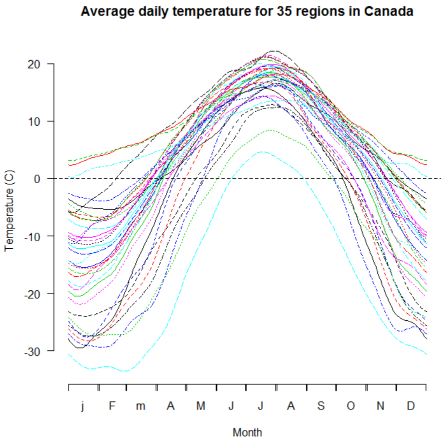
3 Classification for functional data
- Motivation and Goals
- Existing methods
- New Method
- Simulations
- Real Data

4 Future Work

5 References

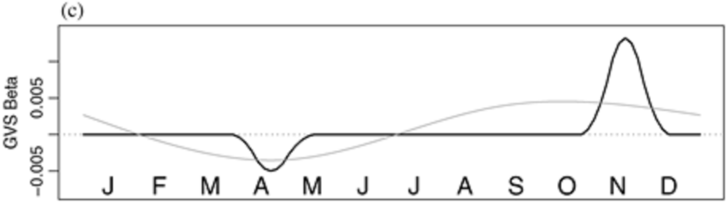
Recall: Canadian Weather Data

Canadian Weather data from Ramsay and Silverman [2005]

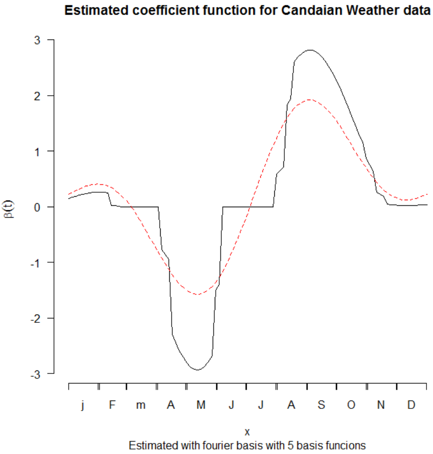


Observations of average daily temperatures from 1960-1994 from 35 locations in Canada.

- Response is average annual precipitation
- Research suggests summer temperatures may not be important for predicting rainfall
- FLiRTI result for average *daily* rainfall



Canadian Weather Data Revisited



Summary of Results

- Flexible method for finding sparse fits and often improves fit overall
- Applicable to any initial fit, $\widetilde{\beta}(t)$
- Results depend on performance of initial fit
- Best performance when $\beta(t)$ has large sparse region(s)
- Better performance for n small relative to FLiRTI

Outline

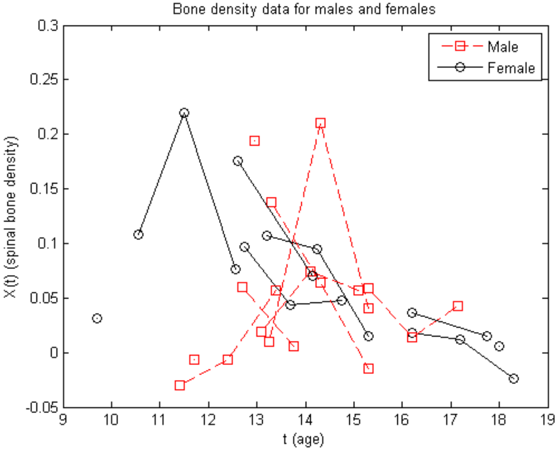
1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

Classification for Functional data

Classification problems with functional predictors are becoming more common in the literatures. One popular data set to look at in this setting is the spinal bone mineral density discussed in Bachrach et al. [1999], James et al. [2000], and James and Hastie [2001].

- The data was collected longitudinally from individuals ages 9-25, with recruitment beginning in 1992.
- The subset of the data used consists of 261 subjects, observed between 1 and 4 times each for a total of 485 observations, sampled at irregular intervals
- We classify based on gender, with 116 males and 145 females

Bone density in adolescents

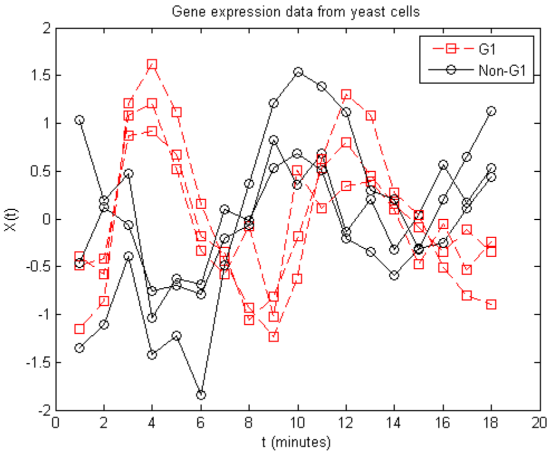


Timecourse gene expression data

In genetics, it can be useful to determine which cellular process particular genes are involved in. This can be accomplished using gene expression data over a time period, where known genes are used as training data.

- Leng and Müller [2006] discuss data from yeast cells, classifying them on whether they are related to a process known as G1 phase regulation.
- We use a data set taken from the same original, consisting of 92 cells observed at 18 fixed time points.
- Leng and Müller [2006] and Park et al. [2008] use timecourse gene expression data from the *Dictyostelium* amoeba to classify which type of cell (prestalk or prespore) a given gene is linked to

Gene expression in yeast



Outline

1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

FLDA

Functional Linear Discriminant Analysis, introduced by James et al. [2000], mentioned above, this was one of the earliest methods to address the problem of classification for functional data

- Uses cubic b-splines to decompose functional predictor, then applies LDA to the scores
- Scores are projected into a low-rank space
- Likelihood maximized via E-M algorithm
- Can be applied to very sparse data (even curves with a single observation)

Logistic Regression and SVM

Logistic regression Introduced by Leng and Müller [2006]

- Decompose curves using principal components using PACE [Yao et al., 2005]
- Apply logistic regression to the scores

Support Vector Machines used by Park et al. [2008] on time-course gene expression data

- Decompose curves using Fourier or B-spline basis
- Perform standard SVM analysis on the coefficients

Outline

1. Introduction
2. Sparse functional linear regression
    - Model assumptions and existing methods
    - New Methodology
    - Simulations
    - Real Data
3. Classification for functional data
    - Motivation and Goals
    - Existing methods
    - New Method
    - Simulations
    - Real Data
4. Future Work
5. References

Discriminant analysis with functional data

In the standard multivariate case, a popular method for classification is discriminant analysis.

- Two-class problem: ith observation from Class k, where $k = 0, 1$
- We observe $\mathbf{x}$ from probability density $f_k(\cdot)$ with mean vector μ_k. The classification rule is then

$$Y = \arg\max_k \frac{f_k(\mathbf{x})\pi_k}{\sum_l f_l(\mathbf{x})\pi_l}, \tag{20}$$

  where π_k is the prior probability that $Y = k$
- Linear discriminant analysis (LDA) uses the above model with the additional assumptions that the f_k are normal with common covariance, Σ. In this case, Equation 20 simplifies to choosing k to maximize

$$log(\pi_k) + (x)'\Sigma^{-1}\mu_k - \mu_k'\Sigma^{-1}\mu_k. \tag{21}$$

LDA with functional data

In the functional case, we observe curves $X_i(t)$ at time points $\{T_{ij}\}_{j=1}^{n_i}$ coming from two classes, $Y_i = 0, 1$, such that

$$X(T_{ij}) = \mu_0(T_{ij}) + \epsilon_0(T_{ij}); \qquad Y = 0 \qquad (22)$$

$$X(T_{ij}) = \mu_1(T_{ij}) + \epsilon_1(T_{ij}); \qquad Y = 1, \qquad (23)$$

- Assume that ϵ_k are independent mean 0 Gaussian processes
- $cov(\epsilon_k(s), \epsilon_k(t)) = V(s, t)$, independent of k
- We assume we can write $V(s, t)$ using an orthogonal eigenfunction expansion, so,

$$V(s, t) = \sum_{j=1}^{\infty} \lambda_j \phi_j(t) \phi_j(s) \qquad (24)$$

Estimating $V(s,t)$

Observations made with error, so we observe $U_{ij} = X_i(T_{ij}) + \delta_{ij}$

- Define $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^n$, where $\mathcal{D}_i = \{(T_{i1}, U_{ij}), \ldots, (T_{in_i}, U_{in_i})\}_{i=1}^n$
- For kth group, estimate a mean function, $\mu_k(t)$ using a local linear smoothing [Fan and Gijbels, 1996, Yao et al., 2005]
- "Center" data by subtracting $\mu_k(t)$ from each observation. Let

$$C_{ij} = U_{ij} - \widehat{\mu}_0(T_{ij})\mathbf{1}_{Y_i=0} - \widehat{\mu}_1(T_{ij})\mathbf{1}_{Y_i=1}. \tag{25}$$

- Use centered data to estimate $V(s,t) = \sum_{j=1}^J \lambda_j \phi_j(t)\phi_j(s)$ by local smoothing
- Can show that $\widehat{V}(s,t)$ is uniformly consistent [Yao et al., 2005]

Consistency of $\widehat{V}(s,t)$

Proposition 1: $\widehat{V}(s,t)$ is uniformly consistent for $V(s,t)$. Define the aggregated error for curve i at time T_{ij},

$$\Delta_{ij} = \Delta(T_{ij}) = \delta_{ij} + \epsilon_i(T_{ij}) + (\mu_k(T_{ij}) - \widehat{\mu}_k(T_{ij})) \tag{26}$$

Therefore,

$$E[\Delta_{ij}|Y_i = k] = 0 \quad \text{and} \quad var(\Delta_{ij}) < \infty. \tag{27}$$

Under some regularity conditions,

$$\sup_{t\in\mathcal{T}} |\widehat{V}(s,t) - V(s,t)| = O_p\left(\frac{1}{\sqrt{n}h_{Vk}}\right) \tag{28}$$

follows from Theorem 1 from Yao et al. [2005].

Decomposing the curves

For each k, a new curve, $X^*(t)$ can be written using the estimated basis functions from $\widehat{V}(s,t)$ as

$$\widehat{X}_k^*(t) = \widehat{\mu}_k(t) + \sum_{l=1}^{J} \widehat{\xi}_{kl}^* \widehat{\phi}_l(t), \tag{29}$$

where ξ_{kl}^* has a Normal distribution with $E(\xi_{kl}^*) = 0$ and $Var(\xi_{kl}^*) = \lambda_l$ if $X^*(t)$ is a member of Class k. For class k, the associated likelihood is

$$L_k = \prod_{l=1}^{J} f_{\widehat{\lambda}_l}(\widehat{\xi}_{kl}^*), \tag{30}$$

where $f_{\widehat{\lambda}_l}$ is the pdf for a $Normal(0, \widehat{\lambda}_l)$.

Classification Rule

Since

$$\frac{L_0}{L_1} \propto \frac{Exp\{-.5\sum_{j=1}^{J}\frac{\widehat{\xi}_{0j}^{*2}}{\widehat{\lambda}_j}\}}{Exp\{-.5\sum_{j=1}^{J}\frac{\widehat{\xi}_{1j}^{*2}}{\widehat{\lambda}_j}\}}, \tag{31}$$

we classify $X^*(t)$ as being in Class 0 if $\left(\frac{\pi_0 L_0}{\pi_1 L_1}\right) > 1$, where π_k is the prior probability of a curve being a member of class k

- $\widehat{\xi}_{kl}^{*2}$ is the estimate for the jth PC score for X_i^* corresponding to class k.
- Normality of ξ_{ik}'s makes for easy computation
- Analogous to LDA classification rule for multivariate data

Multiclass problem

It is relatively straightforward to generalize this approach for the case where $K > 2$. We follow the same steps, centering the data, estimating $\widehat{V}(s,t)$ using centered data from all classes, and then estimating $\widehat{\xi}^*_{kl}$ for each class. By analogy to FDA, note that Equation 31 from the previous slide is equivalent to solving

$$\arg\max_k \delta_k(X(t)), \tag{32}$$

where $\delta_k(X(t)) = Exp\{-.5\sum_{j=1}^{J}\frac{\widehat{\xi}^{*2}_{kj}}{\widehat{\lambda}_j}\} + log(\pi_k)$.

- When k takes more than 2 values, we still choose k with highest associated likelihood
- Identical to the approach from LDA for multivariate data

An alternative approach

An unstated assumption of the method just proposed is that the mean functions, $\mu_k(t)$, are in the space spanned by $\{\phi_j(t)\}_{j=1}^{J}$. Suppose $\mu_0(t)$ is orthogonal to $\phi_j(t)$ for all $j = 1, ..., J$. For a new observation where $Y^* = 1$, estimates of $\widehat{\xi}_{0j}^*$ will be unstable.

- Alternative approach for this case is to not center the data
- Difference of mean functions $\mu_0(t) - \mu_1(t) = \mu_d(t)$ will play role of additional eigenvalue
- Use multivariate LDA on PC scores for classification
- "Uncentered" method should perform well for those cases where one or both of the mean functions is not in the span of $\{\phi_j(t)\}_{j=1}^{J}$.

Outline

1 Introduction

2 Sparse functional linear regression
- Model assumptions and existing methods
- New Methodology
- Simulations
- Real Data

3 Classification for functional data
- Motivation and Goals
- Existing methods
- New Method
- Simulations
- Real Data

4 Future Work

5 References

Data generation

$$X_i(t) = \beta_k + \sum_{l=1}^{3} \xi_{il}\phi_l(t) + \epsilon_i(t) \tag{33}$$

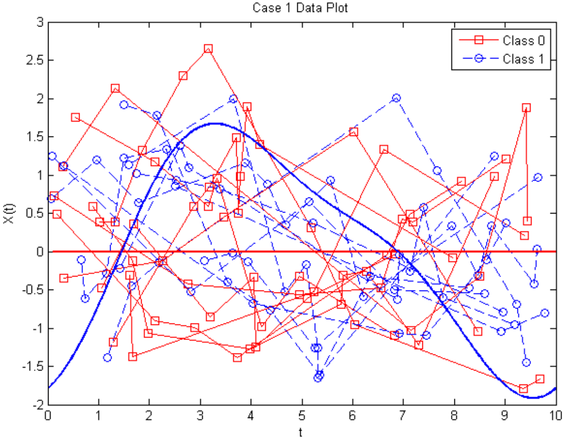
- β_k is a constant vertical shift parameter
- $\phi_l(t)$ are sinusoidal basis functions
- ξ_{il} are random coefficients, normally distributed around μ_{kl} with variance λ_l
- $\epsilon_i(t)$ is random error process with variance $\sigma^2 = 0.25$
- Curves from the kth group centered around $\mu_k = \beta_k + \sum_{l=1}^{3} \mu_{kl}\phi_l(t)$

Case 1

For Case 1, $K = 2$, $\beta_0 = \beta_1 = 0$, which means the difference in the mean functions, $\mu_0(t) - \mu_1(t)$, is in the space spanned by the basis functions, $\{\phi_l(t)\}$.

- $\boldsymbol{\mu}_0 = [0\,0\,0]$ and $\boldsymbol{\mu}_1 = \frac{1}{c}[3\,2\,1]$
- Basis functions should be able to model the two mean functions
- "Centered" method should perform well

Case 1 data


Case 1 Data Plot
Class 0
Class 1
X(t)
t
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
1
2
3
4
5
6
7
8
9
10

Results for Case 1

A subset of the results from simulations for Case 1 are given in the table below.

Table: Classification errors for Case 1

N	n	c	Centered	Uncentered	Logistic	FLDA
10	50	1	0.175	0.177	0.185	0.187
10	100	1	0.171	0.170	0.169	0.172
20	50	1	0.168	0.165	0.169	0.175
20	100	1	0.162	0.160	0.160	0.177
10	50	2	0.336	0.343	0.345	0.351
10	100	2	0.329	0.330	0.330	0.333
20	50	2	0.312	0.319	0.328	0.333
20	100	2	0.306	0.308	0.306	0.313
	Std Dev		0.024	0.018	0.020	0.021

Results for Case 1

All of the methods are close, FLDA is slightly inferior to the other three

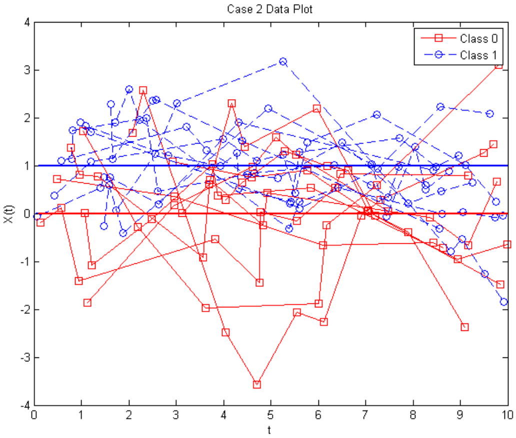
- For small n, our methods appear to be best, with the “centered” slightly outperforming the “uncentered
- Roughly the same performance regardless of problem difficulty (regulated by c)
- As sample size increases and sparsity decreases, the “centered”, “uncentered”, and logistic methods converge while FLDA lags slightly behind

Case 2

For Case 2, $K = 2$, $\beta_0 = 0$, and $\beta_1 = \frac{1}{c}$, which means the difference in the mean functions, $\mu_0(t) - \mu_1(t)$, is in the space spanned by the basis functions, $\{\phi_l(t)\}$.

- Basis functions should not be able to model the two mean functions
- "Uncentered" method should perform well

Case 2 data



Results for Case 2

“Uncentered” method is best, “centered performs poorly.

Table: Classification errors for Case 2

N	n	c	Centered	Uncentered	Logistic	FLDA
10	50	1	0.117	0.060	0.077	0.052
10	150	1	0.125	0.043	0.050	0.048
20	50	1	0.112	0.022	0.031	0.018
20	150	1	0.106	0.014	0.025	0.017
10	50	2	0.296	0.224	0.230	0.187
10	150	2	0.263	0.177	0.180	0.173
20	50	2	0.272	0.138	0.153	0.132
20	150	2	0.243	0.105	0.107	0.114
	Std Dev		0.056	0.011	0.015	0.006

Results for Case 2

The “uncentered” method shows better performance than the “centered” method

- When less data is available, FLDA shows best performance.
- As n and N increase, the “uncentered” method produces superior results.
- “Centered” method struggles because the mean functions are orthogonal to basis functions

Linear mean functions

Also looked at data with the following mean functions:

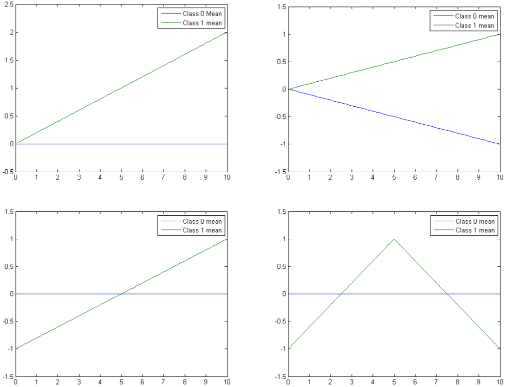


Figure: Linear 1, Linear 2, Cross, and Triangle mean functions

Results for special cases

Table: Classification error for Linear 2 mean functions ($n = 100$)

N	c	Centered	Uncentered	Logistic	FLDA
10	1	0.091	0.035	0.048	0.033
10	1.5	0.199	0.111	0.116	0.111
10	2	0.249	0.172	0.176	0.169
20	1	0.109	0.018	0.027	0.018
20	1.5	0.164	0.055	0.066	0.065
20	2	0.230	0.095	0.102	0.109
Std Dev		0.051	0.011	0.016	0.007

Results from special cases

Table: Classification error for Cross mean functions ($n = 100$)

N	c	Centered	Uncentered	Logistic	FLDA
10	1	0.275	0.256	0.254	0.226
10	1.5	0.389	0.382	0.380	0.327
10	2	0.397	0.403	0.402	0.360
20	1	0.268	0.210	0.212	0.198
20	1.5	0.354	0.305	0.301	0.271
20	2	0.395	0.368	0.364	0.333
Std Dev		0.041	0.030	0.030	0.016

Results for special cases

Results were similar to Case 2 for Linear 1, Linear 2, and Triangle mean functions

- Our "Uncentered" method was similar to FLDA for $N = 10$ and better for $N = 20$
- As difficulty grew, our method also improved relative to FLDA
- "Centered" method struggled with this data
- For the "Cross" mean function, FLDA was the clear winner, with our methods and logistic regression lagging badly behind.

Mutliclass results

N	c	Centered	Uncentered	FLDA
10	1	0.048	0.007	0.023
10	1.5	0.108	0.064	0.077
10	2	0.138	0.130	0.122
20	1	0.065	0.007	0.021
20	1.5	0.096	0.062	0.072
20	2	0.144	0.130	0.109
Std Dev.		0.029	0.036	0.009

Results from $K = 3$ case

Data was similar to Case 1 from before, with $n_k = 40$, and the coefficient functions within the span of the basis functions.

- "Centered" method showed poor results, although as problem became harder, the difference shrunk
- For easier problems, the "uncentered" method outperformed FLDA, but as c increased, FLDA's error rate deteriorated more slowly
- Sparsity was not a major factor for any of the classifiers.

Outline

1. Introduction
2. Sparse functional linear regression
   - Model assumptions and existing methods
   - New Methodology
   - Simulations
   - Real Data
3. Classification for functional data
   - Motivation and Goals
   - Existing methods
   - New Method
   - Simulations
   - Real Data
4. Future Work
5. References

Real Data Examples

In addition to simulated data, "real world" data from three data sets was examined.

- Berkeley Growth Data - Densely sampled data from 91 individuals (39 males, 54 females) from the ages 1 to 18. Each individual's height was observed at 31 fixed time points. Ramsay and Silverman [2005]
- Gene expression data from Leng and Müller [2006] discussed above.

  92 genes, 44 involved in G1 phase regulation
- Spinal bone mineral density data. 261 observations, sparsely and irregularly sampled.

Results from real world data

Table: Leave-one-out CV classification errors for “real world” data sets

Data Set	Centered	Uncentered	Logistic	FLDA
Growth Curves	0.1166	0.0938	0.1361	0.1695
Genes	0.1748	0.1858	0.1538	0.1978
Bone Density	0.4049	0.3269	0.3500	0.3923

Real Data results

For the real data examples, the uncentered method compares favorably to other candidate methods

- Both “uncentered” and “centered” beat competitors for growth curves
- Our methods are comparable to competitors for genes data, with the “centered” method performing slightly worse than logistic regression
- Logistic regression strictly better than FLDA for these examples
- The “uncentered” proved to be clearly superior for the bone density data

Future Work

- Show consistency of classification estimator
- 3rd project: Functional nonlinear regression for sparse and irregular data
- Uses principal components decomposition through PACE and reproducing kernel Hilbert spaces to produce nonlinear estimates of $\beta(t)$
- Simulation study comparing new method to Functional Linear Regression and Functional Quadratic Regression [Yao and Müller, 2010]

Laura K. Bachrach, Trevor Hastie, May-Choo Wang, Balasubramanian Narasimhan, and Robert Marcus. Bone mineral acquisition in healthy asian, hispanic, black, and caucasian youth: A longitudinal study. *Journal of Clinical Endocrinology & Metabolism*, 84(12):4702–4712, 1999.

Jianqing Fan and Irène Gijbels. *Local polynomial modelling and its applications*. Chapman & Hall, 1996.

Gareth M. James and Trevor J. Hastie. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 63(3):pp. 533–550, 2001.

Gareth M. James, Jing Wang, and Ji Zhu. Functional linear regression that's interpretable. *ANNALS OF STATISTICS*, 37:2083, 2009.

GM James, TJ Hastie, and CA Sugar. Principal component models for sparse functional data. *Biometrika*, 87(3):587–602, 2000.

Xiaoyan Leng and Hans-Georg Müller. Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1):68–76, 2006.

Changyi Park, Ja-Yong Koo, Sujong Kim, Insuk Sohn, and Jae Won Lee. Classification of gene functions using support vector machine for time-course gene expression data. *Computational Statistics and Data Analysis*, 52(5):2578 – 2587, 2008.

James O Ramsay and B W Silverman. *Functional Data Analysis (2nd Edition)*. Springer, 2005.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1): 91–108, 2005.

Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernndez, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1457):969–981, 2005.

Fang Yao and Hans-Georg Müller. Functional quadratic regression. *Biometrika*, 97(1):49–64, 2010.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005.