Some selected topics in Functional Data Analysis

Juhyun Park

ENSIIE, Evry, France

Motivation

What is functional data

Exploring funcional data

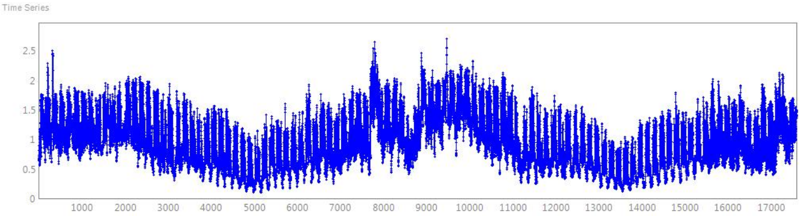
- Principal Component Analysis (PCA)
- Functional Principal Component Analysis

Smoothing methods

Section 1

Motivation

Forecasting challenges

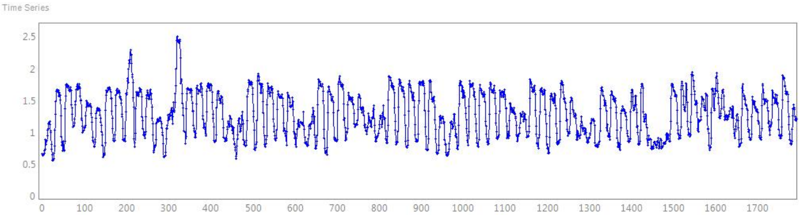


- Novel time series data in high frequency
  - Advances in data processing, recording and storage of vast amounts of data gives observations on previously unobserved periods, recorded at higher, near continuous sampling frequencies
- Time series exhibit novel patterns and pose new challenges in the analysis

Examples of high frequency data

- Blood pressure of patients is constantly monitored and recorded over 24 hours
- EEG brain signals recorded continuously over a period of time
- Meteorological measurements (temperature, humidity..) monitored continuously
- Retail sales volumes store daily and intra-day sales observations or even transaction records (ticks)
- Bidding history at online auction sites such as eBay can be tracked continuously
- Stock market transactions are recorded (and executed) at millisecond intervals of tick-by-tick data

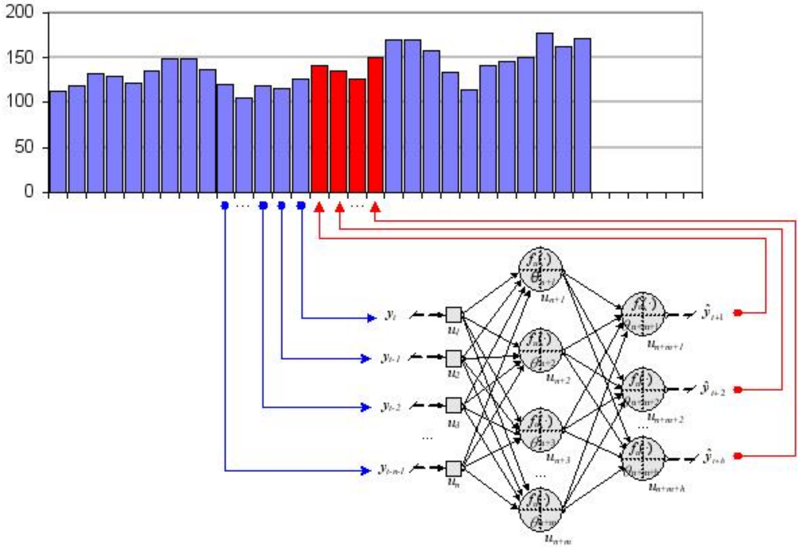
Forecasting gas consumption - zoomed in



Model multiple seasonal patterns... for ARIMA /Neural Networks /Smoothing?

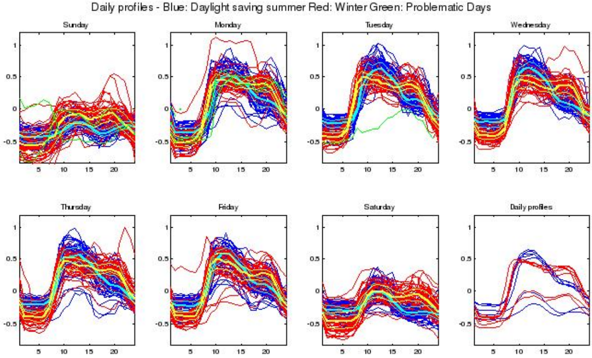
- Daylight savings in summer $\neq$ winter?
- Leap years in AR-processes?
- Variable selection for high-frequency data?

A standard approach - discrete time series



Functional data view

Model multiple seasonal patterns... as functions!



- Model (smooth) transition between functions?
- Consider covariates (temperature, windspeed etc.) in transition

Advantages of functional data view

- Enrich information from discrete data to replicates of functions
  - Periodic, non-stationary behaviour
  - Smooth change between measurements
- Guided analysis by focusing on recovering the common functional structure while taking into account the variations among the replicates.
- Time series forecasting is built on limited number of past observations whereas functional data view encourages us to look at all the past observations.
- Functional data facilitates an analysis of the smooth transitions between these different groups, possibly formulated as having hierarchical (multi-level) structure

Section 2

What is functional data

Big picture: Statistical Learning

Statistical framework

- Population: the set of all possible *units* of interest
- Sample: a (random) subset of population

From finite sample, we would like to draw conclusions about the population.

Statistical analysis

- Data: multiple observation $\{x_i, i = 1, \ldots, n\}$
- Aim: understanding and characterizing variability in the sample of observations

What is the unit of the analysis

Multiple observation $\{X_1, \ldots, X_n\}$, X_i characteristics of subject i

- Number: univariate analysis

  $$X_i : \text{commuting distance in km}, \quad x_i = 5.2$$

- Vector: multivariate analysis

  $$X_i = (\text{distance(km), age (y)}), \quad x_i = (5.2, 32)$$

- Function: functional data analysis

  $$X_i = \text{commuting trajectory}, \quad x_i = ?$$

- Shape analysis, image analysis, topological data analysis, manifold analysis ...
- *Object Oriented Data Analysis*

The unit of analysis is the whole trajectory, not the finite number of available observations!

Functional data

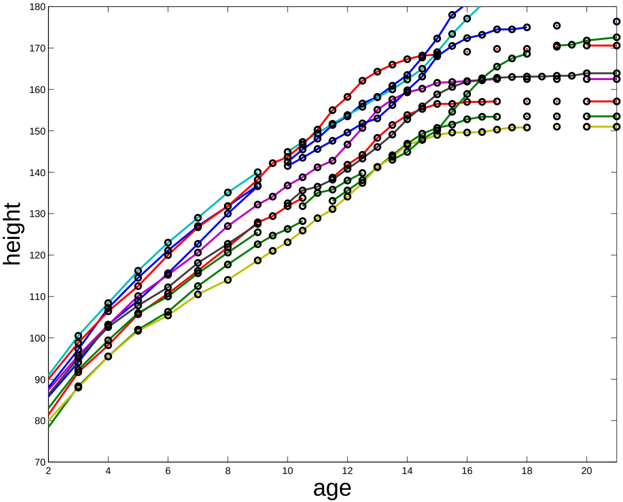
Most often, functional data refers to measurements on a curve but in a broader sense, it is also used to encompass images, tree-like objects and many other non-Euclidian objects arising in modern applications $\rightarrow$ *Object Oriented Data Analysis*

Observations from functional data:

- repeated measurements available from multiple subjects
- often densely observed, though sparse observations are also dealt with.
- often represent the underlying *continuous*, possibly smooth, (physical or biological) process

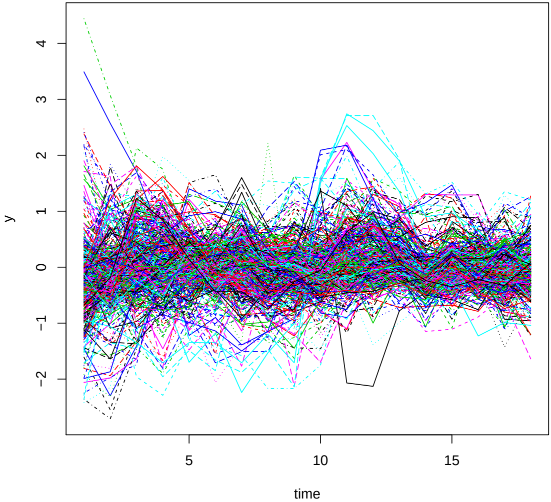
Example: Growth curves

A sample of child's growth measurements (height) until 21 years



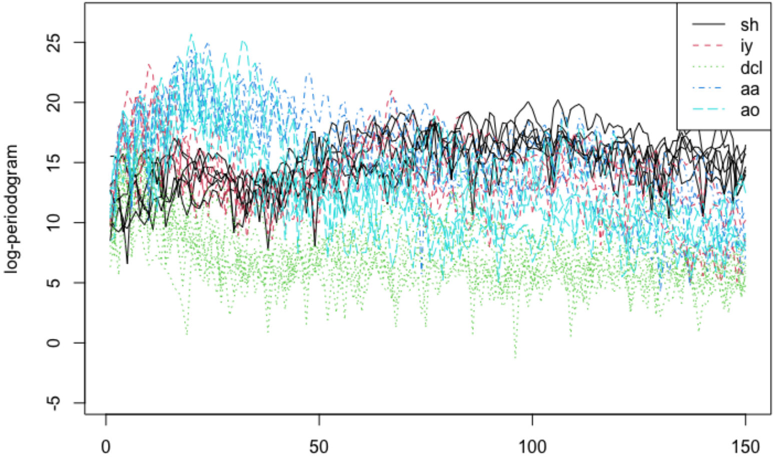
Example: Genome's cell cycle regulation

A sample of gene expression (mRNA levels) data measured every 7 minutes during 119 minutes



Example: Speech recognition

A sample of log-periodograms of speech recording of 32 ms for 5 classes of phonemes in the first 150 frequencies

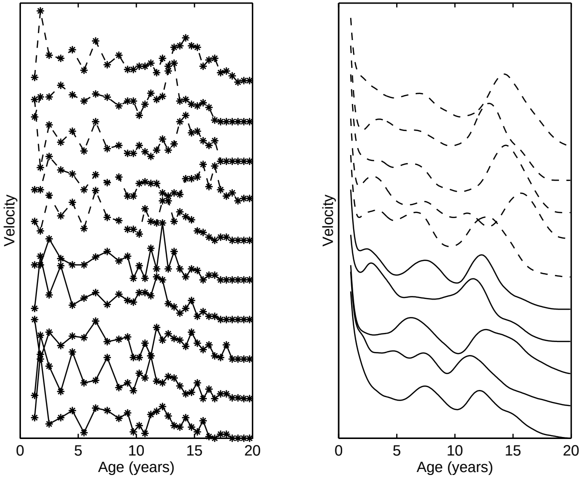


Common characteristics of measurements

- only discrete measurements are available: $X_j, j = 1, \ldots, n$
- often replicates of functions are available: $X_{ij}, i = 1, \ldots, n; j = 1, \ldots, m$
- sampling points may vary from one record to another; $X_{ij}, i = 1, \ldots, n; j = 1, \ldots, n_i$
- measurement error may be present: $Y_{ij} = X_{ij} + \varepsilon_{ij}$
- values reflect a smooth variation that could be assessed at *any* time or as often as desired: $X_{ij} = f_i(t_{ij})$
- values are continuous in nature thus should be viewed as a *function*: $f_i(\cdot)$ is continuous
- functional features such as derivatives could be of main interest: $f_i(\cdot)$ differentiable

Estimating velocity curves for trunk length

Velocities of trunk length for 5 boys (above) and 5 girls (below).
left = raw velocities, right = kernel estimated velocities.



Velocity estimation

Raw velocity:

$$\tilde{y}_j = \frac{y(t_j) - y(t_{j-1})}{t_j - t_{j-1}} \qquad s_j = \frac{t_j + t_{j-1}}{2}$$

Estimated velocity:

$\tilde{y}(t) = \text{smooth}\{(s_1, \tilde{y}_1), (s_2, \tilde{y}_2), \ldots, (s_p, \tilde{y}_p)\}$ and evaluate at t

Function representation: uni-dimensional case

- Functions are infinite-dimensional objects: $f : \mathcal{I} \to \mathrm{R}$
- Finite observations are available: $f(t_1), \ldots, f(t_n), t_j \in \mathcal{I}$.
- Observations without error made on fine grid points: *numerical interpolation* on a finite grid

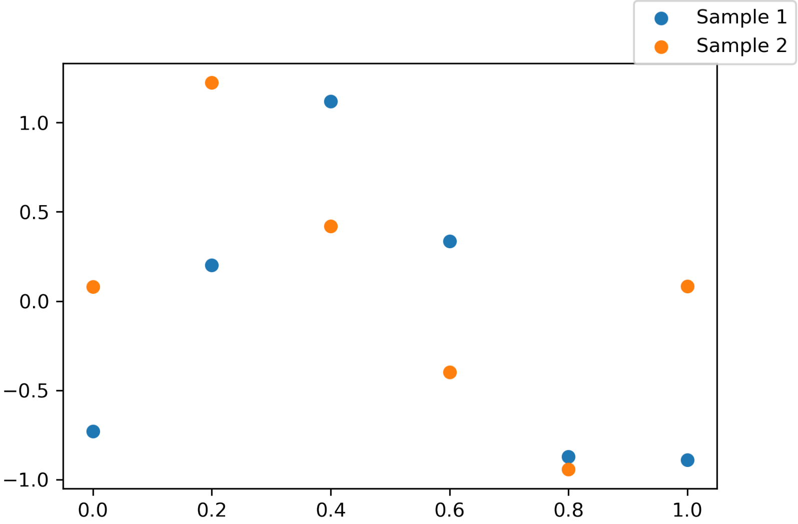
$$\boldsymbol{X} = (f(t_1), \ldots, f(t_m))$$

- Measurement error in data: $Y_1, \ldots, Y_n$, noisy observations of X_i:

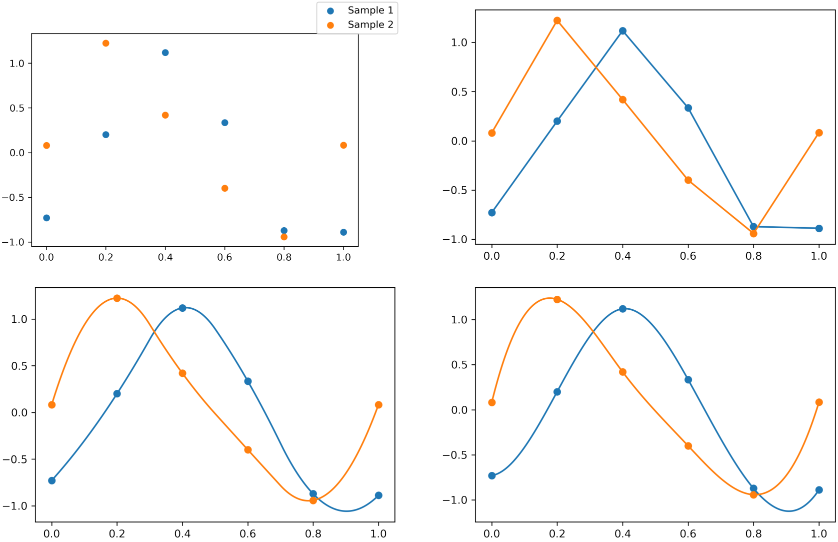
$$Y_i(t) = X_i(t) + \varepsilon_i(t)$$

  where ε_i's are independent of X_i's and are independent and identically distributed zero mean stationary processes. $\Rightarrow$ *statistical interpolation = smoothing*

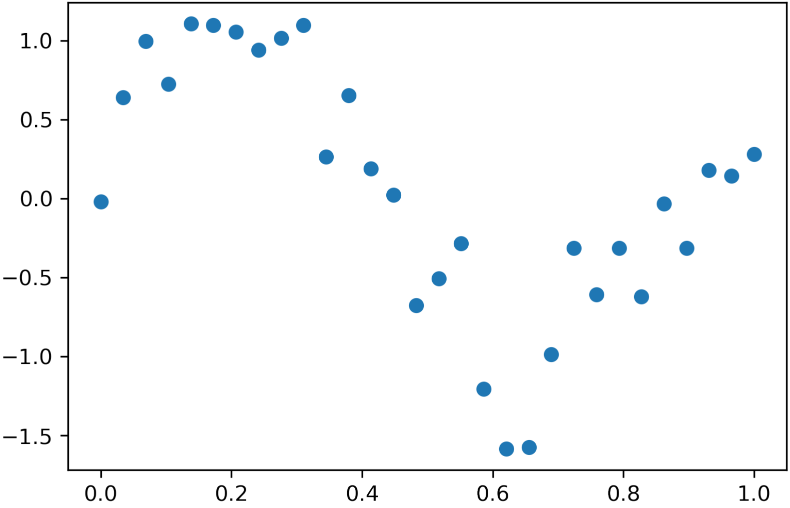
Function representation: without noise



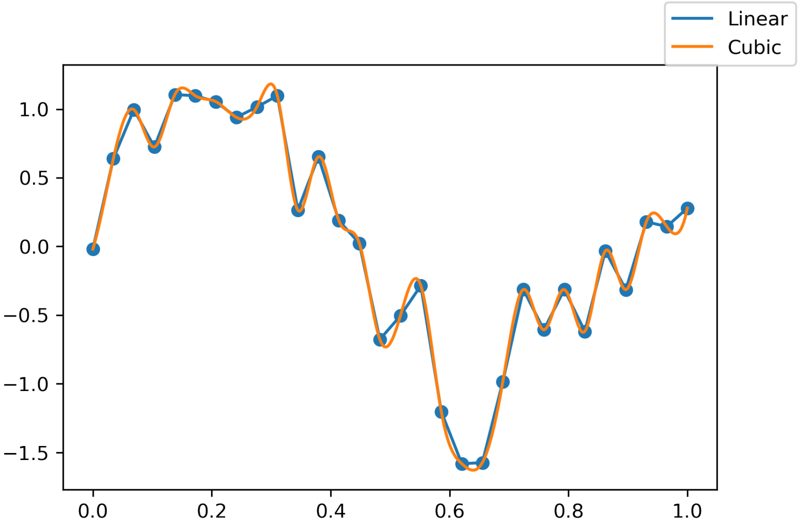
Interpolation (linear, quadratic, cubic)



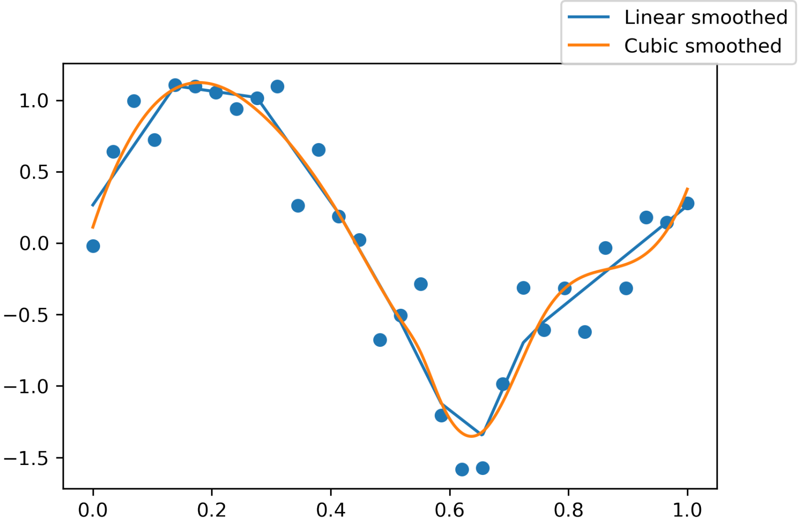
Function representation: with noise



Interpolation (linear, cubic)



Statistical interpolation (linear, cubic)



Vector vs Function

- Although a vector representation gives a convenient mean to represent/compute/manipulate function valued variables, they are not equivalent.
- Multivariate analysis deals with *fixed dimension* (number of available observations per curve), whereas functional data does not have a fixed dimension to begin with.
- Vector representation cannot account for the fact that future observations (test data) may not have the same representation as the available data (training data) for functional data
- For functional data, often the interest lies in derivatives of the functions!

Overview of functional data analysis

Features of data:

- similar pattern of variation among curves: share common structure or *common shape*
- complex relationship and complex variability
- interest in functional characteristics: nonlinear relationship, derivatives ...

Aims of the analysis:

- understand and characterize the *common features* of the homogeneous population
- discriminate and classify *distinct* populations
- extract *maximal* information with an *efficient* representation

Some references

- Ramsay, J. O. and Silverman, B. W. (1997, 2005) Functional Data Analysis
- Ramsay, J. O. and Silverman, B. W. (2002) Applied Functional Data Analysis: Methods and Case Studies
- Ramsay, J. O., Hooker, G., Gaves, S. (2005) Functional Data Analysis with R and Matlab
- Ferraty, F. and Vieu, P. (2006) Nonparametric Functional Data Analysis: Theory and Practice
- Hastie, T., Tibshirani, R. and Friedman, J. (2001, 2008) The Elements of Statistical Learning: Data Mining, Inference and Prediction

Section 3

Exploring funcional data

Multivariate data

Let $X = (X_1, \dots, X_p)^\top \in \mathbb{R}^p$ be a p-dimensional random vector with mean μ and variance Σ. Then Σ is a $p \times p$ symmetric and positive-definite (excluding constant variable case) matrix.

Data:

$$X_i = (X_{i1}, \dots, X_{ip})^\top \qquad i = 1, \dots, n$$

Mean $\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)$ and $\Sigma = (\sigma_{ij})$ where

$$\mathrm{E}[X_{ij}] = \mu_j \qquad \mathrm{cov}(X_{ij}, X_{ik}) = \sigma_{ij}$$

$$\Sigma = \mathrm{cov}(X) = \begin{pmatrix} \mathrm{var}(X_1) & \mathrm{cov}(X_1, X_2) & \dots & \mathrm{cov}(X_1, X_p) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cov}(X_p, X_1) & \mathrm{cov}(X_p, X_2) & \dots & \mathrm{var}(X_p) \end{pmatrix}$$

Sample mean and variance

- Sample mean for jth component: $\bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}$.
- Sample mean vector:

$$\bar{X} = (\bar{X}_1, \ldots, \bar{X}_p)^\top$$

- Residual: $\tilde{X}_i = X_i - \bar{X}$
- Sample covariance:

$$\begin{aligned}
\Sigma_n &= \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_i \tilde{X}_i^\top \\
&= \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_{i1}^2 & \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_{i1}\tilde{X}_{i2} & \ldots & \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_{i1}\tilde{X}_{ip} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_{ip}\tilde{X}_{i1} & \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_{ip}\tilde{X}_{i2} & \ldots & \frac{1}{n}\sum_{i=1}^{n} \tilde{X}_{ip}^2 \end{pmatrix}
\end{aligned}$$

Functional random variable

- Population: let $X \in L^2(\mathcal{I})$ be the functional random variable on $\mathcal{I}$ with

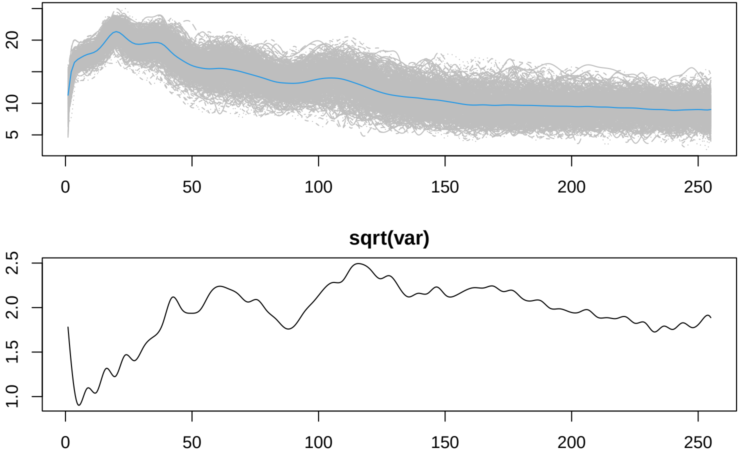
$$\mathrm{E}[X(t)] = \mu(t) \qquad \mathsf{cov}(X(s), X(t)) = \Gamma(s,t)$$

- Sample: $X_1, \ldots, X_n$, assume that these are independent and identically distributed as X.
- Sample mean: $\bar{X}(t) = \frac{1}{n}\sum_{i=1}^{n} X_i(t)$ for $t \in \mathcal{I}$
- Sample cov: for $(s,t) \in \mathcal{I} \times \mathcal{I}$

$$\hat{\Gamma}(s,t) = \frac{1}{n}\sum_{i=1}^{n}(X_i(s) - \bar{X}(s))(X_i(t) - \bar{X}(t))$$

- Sample variance: $\hat{v}(t) = \hat{\Gamma}(t,t)$
- Sample standard deviation: $\hat{s}(t) = \sqrt{\hat{v}(t)}$

Phoneme: mean and std function



Multivariate: Eigen-decomposition of covariance matrix

Let λ_k be the eigenvalue of Σ with the corresponding eigenvector $\boldsymbol{u}_k$, that is, $\Sigma \boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k$. Then

$$\begin{aligned} \Sigma &= UDU^\top = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p]\text{diag}(\lambda_1, \ldots, \lambda_p\} \begin{pmatrix} \boldsymbol{u}_1^\top \\ \vdots \\ \boldsymbol{u}_p^\top \end{pmatrix} \\ &= \sum_{k=1}^{p} \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^\top \end{aligned}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$ and $U^\top U = UU^\top = I_p$.

For sample covariance matrix:

$$\Sigma_n = U_n D_n U_n^\top = \sum_{k=1}^{p} \hat{\lambda}_k \widehat{\boldsymbol{u}}_k \widehat{\boldsymbol{u}}_k^\top$$

Variance decomposition

Total variation = Mean variation + Mean residual variation:

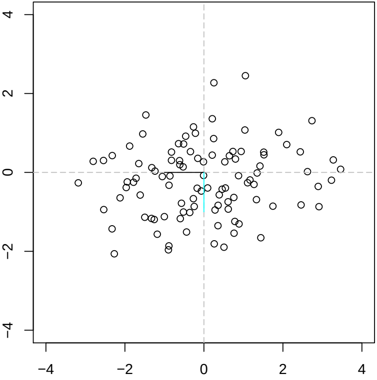
$$\sum_{i=1}^{n} \|X_i\|^2 = \sum_{i=1}^{n} \|\bar{X}\|^2 + \sum_{i=1}^{n} \|X_i - \bar{X}\|^2$$

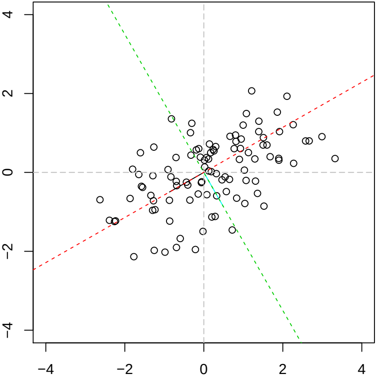
Decomposition of mean residual variation:

$$\begin{aligned}
\sum_{i=1}^{n} \|X_i - \bar{X}\|^2 &= \sum_{i=1}^{n} (X_i - \bar{X})^\top (X_i - \bar{X}) \\
&= \operatorname{trace}\Big(\sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})^\top \Big) \\
\frac{1}{n} \sum_{i=1}^{n} \|X_i - \bar{X}\|^2 &= \operatorname{trace}\big(\Sigma_n\big) \\
&= \operatorname{trace}(U_n D_n U_n^\top) = \operatorname{trace}(D_n) \\
&= \sum_{j=1}^{p} \hat{\lambda}_j
\end{aligned}$$

PCA - toy example

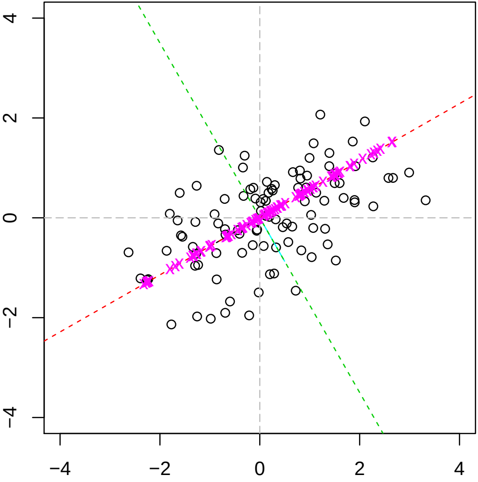
$p = 2$





PCA

Find *direction of greatest variability*



PCA data representation

- Eigenvectors are orthonormal basis:

$$X_i = \boldsymbol{\mu} + \sum_{k=1}^{p} \langle X_i - \boldsymbol{\mu}, \boldsymbol{u}_k \rangle \boldsymbol{u}_k = \boldsymbol{\mu} + \sum_{k=1}^{p} a_{ik} \boldsymbol{u}_k$$

where $a_{ik} \sim N(0, \lambda_k)$.

- Sample representation:

$$X_i = \bar{X} + \sum_{i=1}^{p} \langle X_i - \bar{X}, \widehat{\boldsymbol{u}}_k \rangle \widehat{\boldsymbol{u}}_k = \bar{X} + \sum_{k=1}^{p} \hat{a}_{ik} \widehat{\boldsymbol{u}}_k$$

where $\hat{\alpha}_{ik} \sim (0, \hat{\lambda}_k)$

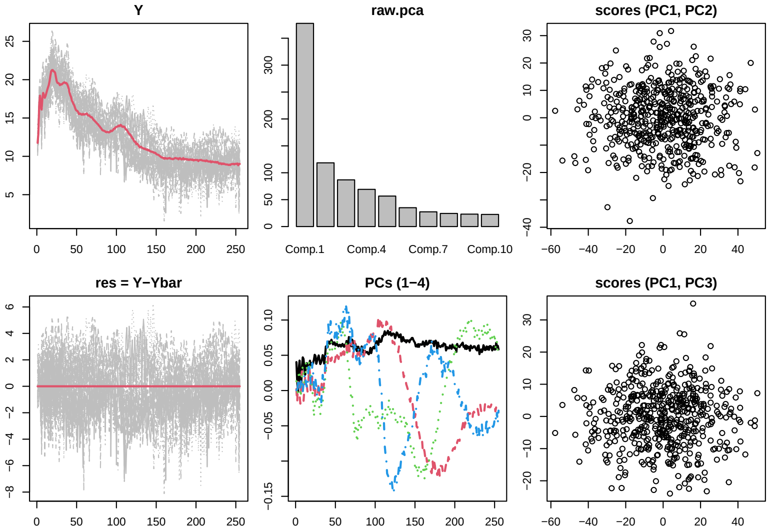
- Dimension reduction: use $K < p$ components

$$\hat{X}_i = \bar{X} + \sum_{i=1}^{K} \langle X_i - \bar{X}, \widehat{\boldsymbol{u}}_k \rangle \widehat{\boldsymbol{u}}_k = \bar{X} + \sum_{k=1}^{K} \hat{a}_{ik} \widehat{\boldsymbol{u}}_k$$

- Estimation of the inverse of covariance matrix:

$$\Sigma_n \approx \sum_{k=1}^{K} \hat{\lambda}_k \widehat{\boldsymbol{u}}_k \widehat{\boldsymbol{u}}_k^\top, \quad \widehat{\Sigma^{-1}} = \sum_{k=1}^{K} \frac{1}{\hat{\lambda}_k} \widehat{\boldsymbol{u}}_k \widehat{\boldsymbol{u}}_k^\top$$

Phoneme: (multivariate) PCA



Functional PCA

Karhunen-Lóeve decomposition:
Assume that $\mathrm{E}[X_i(t)] = \mu(t)$, $\mathsf{cov}(X_i(s), X_i(t)) = \Gamma(s,t)$

$$\Gamma(s,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(s) \phi_k(t)$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ are eigenvalues and $\phi_1, \phi_2, \ldots$ are the corresponding eigenfunctions.
Eigen-decomposition of the variance-covariance function:

$$\int_{\mathcal{I}} \Gamma(s,t) \phi_k(s)\, ds = \lambda_k \phi_k(t)$$

subject to $\int_{\mathcal{I}} \phi_k^2(t) = 1$ and $\int_{\mathcal{I}} \phi_k(t) \phi_\ell(t) = 0$ for $k \neq \ell$.

Functional PCA

Functional variable X_i can be expressed as

$$X_i(t) = \mu(t) + \sum_{k=1}^{\infty} \langle X_i, \phi_k \rangle \phi_k(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik} \phi_k(t)$$

where $\xi_{ik} \sim (0, \lambda_k)$.

Approximation based on FPCA:

$$X_i^K(t) = \mu(t) + \sum_{k=1}^{K} \xi_{ik} \phi_k(t)$$

Dimension reduction based on FPCA:

$$\hat{X}_i(t) = \hat{\mu}(t) + \sum_{k=1}^{K} \hat{\xi}_{ik} \hat{\phi}_k(t)$$

where $\hat{\xi}_{ik} \sim (0, \hat{\lambda}_k)$.

Functional PCA with noisy data

$$Y_{ij} = X_i(t_{ij}) + \varepsilon_i(t_{ij})\,, \quad X_i(t) = \mu(t) + \sum_{k=1}^{K} \xi_{ik}\phi_k(t)$$

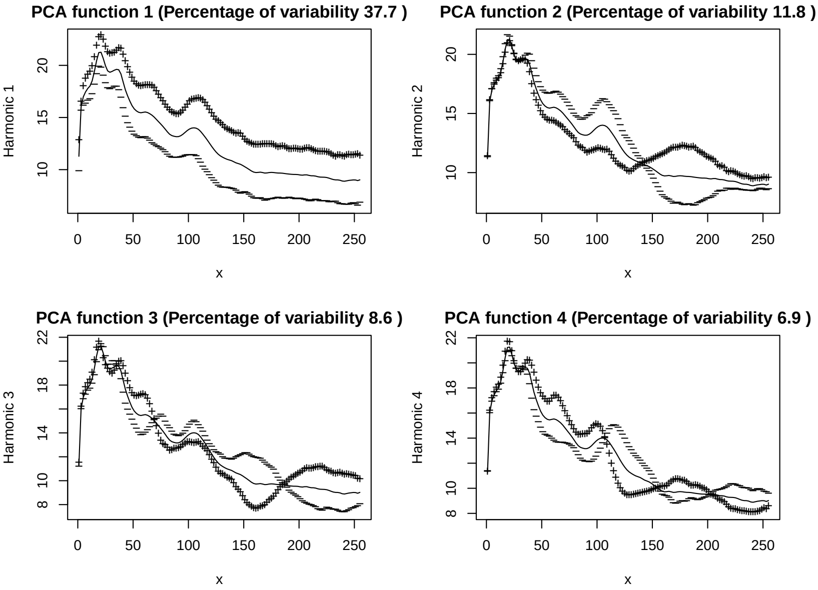
Method 1: discretization on common grid

- Individual smoothing: $\hat{X}_i(t) = \mathsf{Smooth}(t, Y_{ij}, j = 1, \ldots, n_i)$
- Interpolate at common grid points at $t_1, \ldots, t_m$:
  $X_i = (\hat{X}_i(t_1), \ldots, \hat{X}_i(t_m))^\top$
- Apply multivariate PCA (up to scaling factors correction)

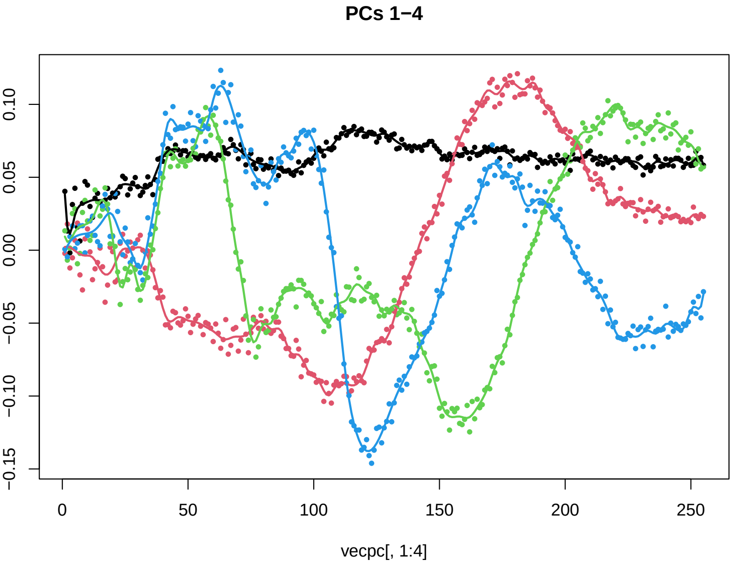
Method 2: FPCA by regularization

- Single smoothing for mean: $\hat{\mu}(t) = Smooth(\frac{1}{n}\sum_{i=1}^{n} Y_{ij})$
- Residuals: $r_{ij} = Y_{ij} - \hat{\mu}(t_{ij})$
  - Smooth covariance ($r_{ij}r_{ik}$) before eigen-decomposition
  - Smooth eigenfunction by regularization

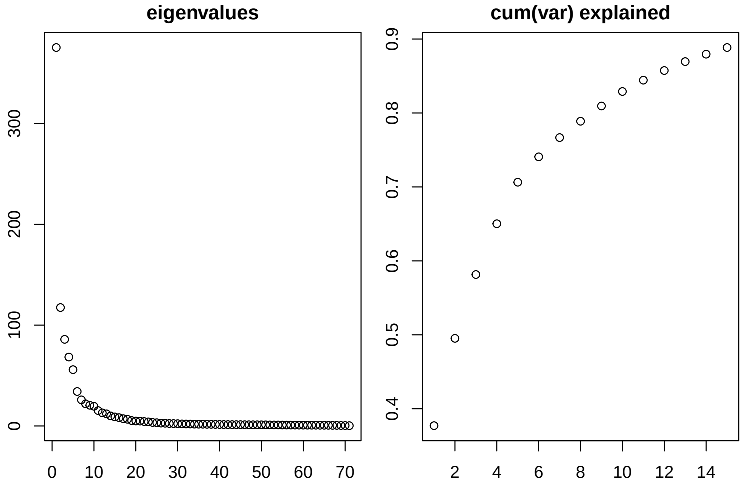
Phoneme: FPCA eigenfun



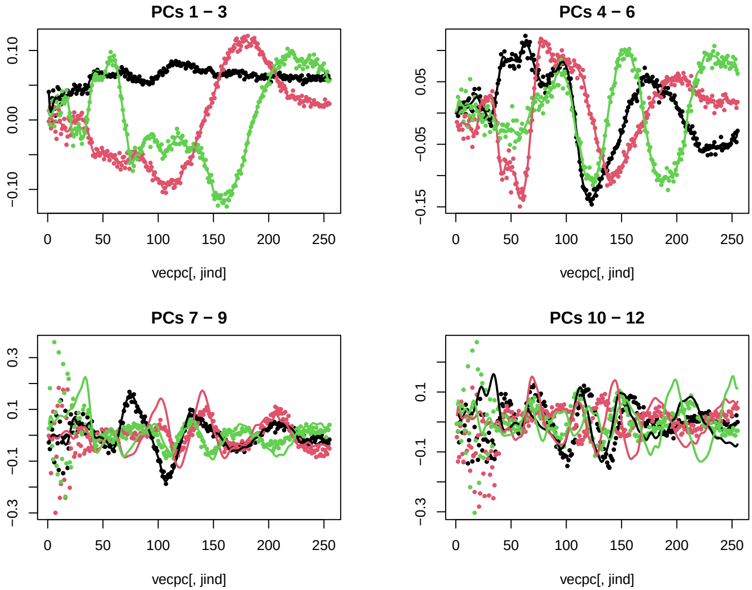
Phoneme: FPCA vs PCA eigenfunc



Phoneme: FPCA eigenvalues



Phoneme: FPCA eigenfunctions (cont)



Section 4

Smoothing methods

Representing functions with noisy data

- Data: n noisy observations ($Y_i, i = 1, \ldots, n$) available at fixed or random design points ($T_i \in [a, b]$)
- Assumption: data represent unknown smooth function (f)
- Statistical Model:

$$Y_i = f(T_i) + \varepsilon_i \qquad i = 1, \ldots, n$$

  where the error has mean zero and finite variance.
- The underlying function represents

$$f(t) = \mathrm{E}[Y|T = t] \qquad f(T) = \mathrm{E}[Y|T]$$

  the conditional mean of Y given $T = t$.
- Aim: recover $f(t), t \in [a, b]$ from finite number of noisy data
- Point estimation at t: $\hat{X}(t) = \hat{f}(t), t \in [a, b]$

$\Rightarrow$ *Nonparametric regression problem*

Nonparametric regression

Observed data: $(t_i, y_i) : i = 1, \ldots, n$

Find $f(\cdot) \in \mathcal{F} = \{f : \text{ continuous}\}$ that minimizes the squared error

$$\sum_{i=1}^{n} \{y_i - f(t_i)\}^2$$

- feasible set too large: can find an exact solution (*overfit*)
- need to impose contraints: use *smoothness* constraints
  - Local (polynomial) approximation: for t in the neighborhood t_0

$$\begin{aligned} f(t) &\approx f(t_0) + f'(t_0)(t - t_0) + \ldots + \frac{f^{(p)}(t_0)}{k!}(t - t_0)^p \\ &= \beta_0 + \beta_1(t - t_0) + \ldots + \beta_p(t - t_0)^p \end{aligned}$$

  - Global approximation: choose basis functions $\{\phi_1, \ldots, \phi_k\}$ for all $t \in [a, b]$

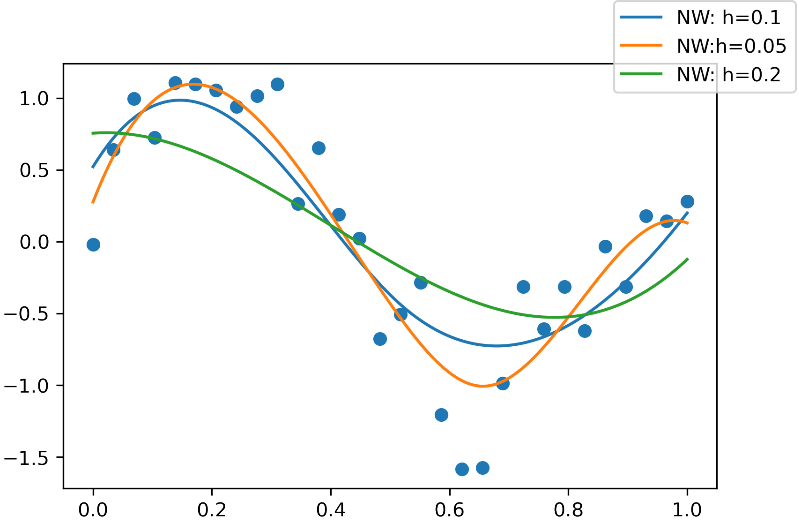
$$f(t) \approx \alpha_1\phi_1(t) + \alpha_2\phi_2(t) + \ldots \alpha_k\phi_k(t)$$

Standard smoothing methods

Different ways to control smoothness in the function:

- *Kernel smoothing* or *Local polynomial regression*: minimize
$$\sum_{i=1}^{n} w_i\{y_i - \beta_0 - \beta_1(t_i - t) - \dots \beta_p(t_i - t)^p\}^2$$
  - smoothing parameter: size of neighborhood (h)
- *Regression splines*: minimize
$$\sum_{i=1}^{n}\{y_i - a_1\phi_1(t_i) - \dots - a_k\phi_k(t_i)\}^2$$
  - smoothing parameter: number of basis functions k
- *Smoothing splines*: minimize
$$\sum_{i=1}^{n}\{y_i - a_1\phi_1(t_i) - \dots - a_k\phi_k(t_i)\}^2 + \lambda P(f)$$
  where $P(f)$ is smoothness penalty, often $P(f) = \int_a^b \{f''(t)\}^2\, dt$
  - smoothing parameter: λ

Kernel smoothing: smoothing parameter



Smoothing splines: smoothing parameter