Functional Data Analysis (Lecture 1)

Zoltán Szabó

October 4, 2016

Email, course link, references

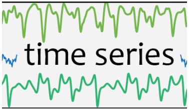
- Email: zoltan (dot) szabo (at) polytechnique (dot) edu
- Course: `http://www.cmap.polytechnique.fr/~zoltan.szabo/teaching_FDA.html`

References:

- [1] J.O. Ramsay, B.W. Silverman. Functional Data Analysis. Springer, 2005.
- [2] J.O. Ramsay, Giles Hooker, Spencer Graves. Functional Data Analysis with R and Matlab. Springer, 2009.

Contents

- Motivation: examples, challenges.



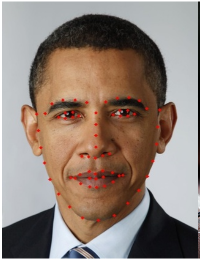
- Closely related methods:
    - Smoothing by least squares → linear smoother.
    - Localized least squares = kernel smoothing.
    - Their combination = 'locally linear smoother'.

Motivation: examples, challenges.

Time-series: examples

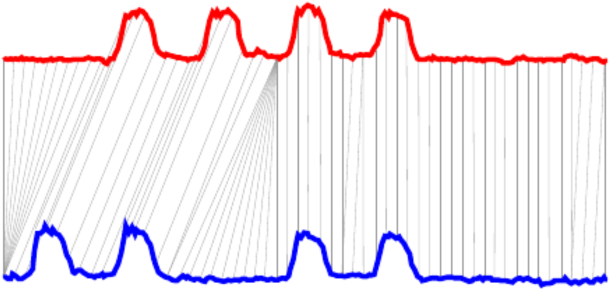
Bank transactions, oil refinement, MOCAP, face tracking, weather, brain imaging, . . .





DFA in nutshell

- Input data: functions. Challenges+:
  - curves: might not be aligned (registration).



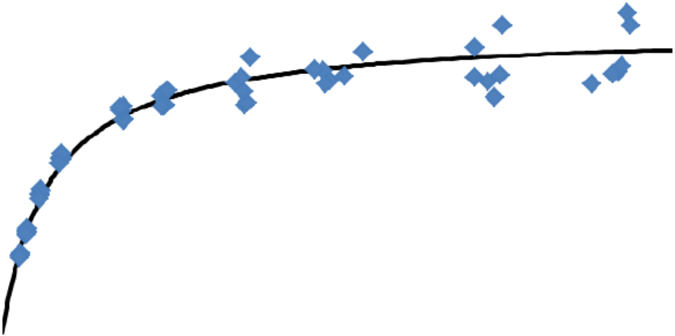
  - noisy observations (denoising).
  - interplay between noise & smoothness. Dominant patterns=?
  - possibly constraints: non-negativity, monotonicity, . . .
- Assumption: smoothness.

Smoothing

Problem formulation

Given:

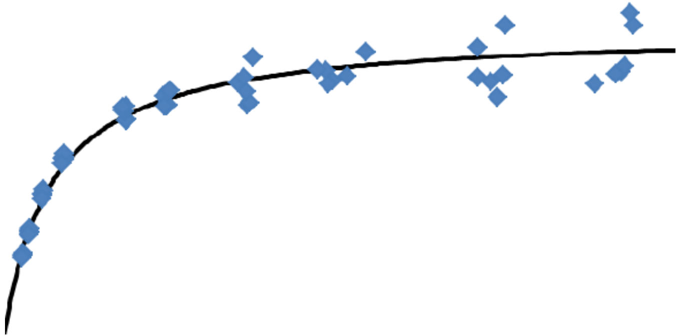
$$y_i = x(t_i) + e_i, (i = 1, \ldots, n).$$



Problem formulation

Given:

$$y_i = x(t_i) + e_i, (i = 1, \ldots, n).$$



t: often time; but can be space, ...

Assumptions

Given:

$$y_i = x(t_i) + e_i, (i = 1, \dots, n).$$

Assumptions:

- x: *smooth*,
- standard model: $var(\mathbf{y}) = var(\mathbf{e}) = \sigma^2 \mathbf{I}$.

Assumptions

Given:

$$y_i = x(t_i) + e_i, (i = 1, \ldots, n).$$

Assumptions:

- x: *smooth*,
- standard model: $var(\mathbf{y}) = var(\mathbf{e}) = \sigma^2 \mathbf{I}$.

Some critisism:

- $var(e_j)$ might change (height vs ages),
- $cov(e_i, e_j) = 0$ $(i \neq j)$: also simplifying.

Basis function techniques

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$.

Basis function techniques

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle ,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$. Examples:

- Monomials: $1, t, t^2, t^3, \ldots$

Basis function techniques

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle ,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$. Examples:

- Monomials: $1, t, t^2, t^3, \ldots$
- Fourier system: $1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t), \ldots$

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$. Examples:

- Monomials: $1, t, t^2, t^3, \ldots$
- Fourier system: $1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t), \ldots$
- Splines: piecewise polynomials, connecting smoothly.

Basis function techniques

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle ,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$. Examples:

- Monomials: $1, t, t^2, t^3, \dots$
- Fourier system: $1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t), \dots$
- Splines: piecewise polynomials, connecting smoothly.
- Wavelets: $2^{\frac{j}{2}} \psi(2^j \cdot - k)$, $(j, k) \in \mathbb{Z}^2$; ψ: mother wavelet.

Basis function techniques

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$. Examples:

- Monomials: $1, t, t^2, t^3, \ldots$
- Fourier system: $1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t), \ldots$
- Splines: piecewise polynomials, connecting smoothly.
- Wavelets: $2^{\frac{j}{2}}\psi(2^j \cdot -k)$, $(j,k) \in \mathbb{Z}^2$; ψ: mother wavelet.
- Exponential base: $e^{\lambda_1 t}$, $e^{\lambda_2 t}$, $\ldots (\lambda_i > 0)$.

Basis function techniques

Representation:

$$x(t) = \sum_{k=1}^{B} c_k \phi_k(t) = \langle \mathbf{c}, \phi_t \rangle \, ,$$

where ϕ_k-s are linearly independent, $c_k \in \mathbb{R}$. Examples:

- Monomials: $1, t, t^2, t^3, \ldots$
- Fourier system: $1, \sin(\omega t), \cos(\omega t), \sin(2\omega t), \cos(2\omega t), \ldots$
- Splines: piecewise polynomials, connecting smoothly.
- Wavelets: $2^{\frac{j}{2}} \psi(2^j \cdot -k)$, $(j,k) \in \mathbb{Z}^2$; ψ: mother wavelet.
- Exponential base: $e^{\lambda_1 t}$, $e^{\lambda_2 t}$, $\ldots (\lambda_i > 0)$.
- Power base: t^{λ_1}, t^{λ_2}, $\ldots$

Smoothing: by least squares.

Least squares fit: unweighted

$$J(\mathbf{c}) = \sum_{j=1}^{n} \left[y_j - \sum_{k=1}^{B} c_k \underbrace{\phi_k(t_j)}_{\Phi_{jk}} \right]^2 = \|\mathbf{y} - \mathbf{\Phi c}\|_2^2 \to \min_{\mathbf{c} \in \mathbb{R}^B}.$$

Solution ($\frac{\partial J}{\partial \mathbf{c}} = \mathbf{0}$):

$$\hat{\mathbf{c}} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{y},$$
$$\hat{\mathbf{y}} = \mathbf{\Phi} \hat{\mathbf{c}} = \underbrace{\mathbf{\Phi}(\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T}_{=:\mathbf{S}:\text{ linear smoother}} \mathbf{y}.$$

Note: here **S** is the projection to $Im(\mathbf{\Phi})$.

Least squares fit: Weighted

Given: $\mathbf{W}$ symmetric, positive definite.

$$J(\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})^T \mathbf{W} (\mathbf{y} - \mathbf{\Phi c}) \rightarrow \min_{\mathbf{c} \in \mathbb{R}^B}.$$

Solution:

$$\hat{\mathbf{c}} = (\mathbf{\Phi}^T \mathbf{W} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{W} \mathbf{y},$$

$$\hat{\mathbf{y}} = \mathbf{\Phi} \hat{\mathbf{c}} = \underbrace{\mathbf{\Phi} (\mathbf{\Phi}^T \mathbf{W} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{W}}_{= \mathbf{S}} \mathbf{y}$$

Unweighted case: $\mathbf{W} = \mathbf{I}$. Ideally: $\mathbf{W} = \mathbf{\Sigma}_{\boldsymbol{e}}^{-1}$.

Bias-variance tradeoff: danger of overfitting

B: level of smoothing. Large B (overfitting):

$$Bias[\hat{x}(t)] = x(t) - \mathbb{E}[\hat{x}(t)] \to \text{small}; \; Bias = 0 \text{ for } B = n,$$

$$Var[\hat{x}(t)] = \mathbb{E}\left[\hat{x}(t) - \mathbb{E}\hat{x}(t)\right]^2 \to \text{high}.$$

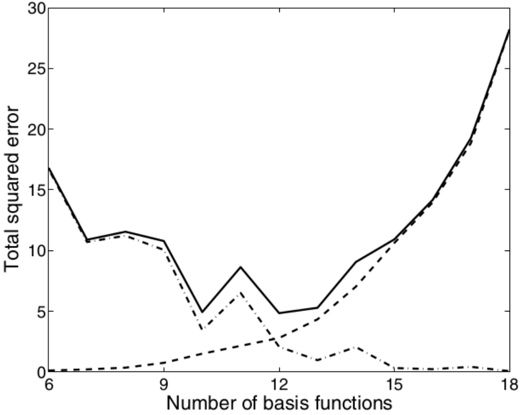
Bias-variance tradeoff: danger of overfitting

B: level of smoothing. Large B (overfitting):

$$\textit{Bias}[\hat{x}(t)] = x(t) - \mathbb{E}[\hat{x}(t)] \to \text{small}; \; \textit{Bias} = 0 \text{ for } B = n,$$

$$\textit{Var}[\hat{x}(t)] = \mathbb{E}\left[\hat{x}(t) - \mathbb{E}\hat{x}(t)\right]^2 \to \text{high}.$$

$\textit{MSE}[\hat{x}(t)] = \mathbb{E}[x(t) - \hat{x}(t)]^2 = \textit{Bias}^2[\hat{x}(t)] + \textit{Var}[\hat{x}(t)]$. Tradeoff:



Heuristics for B choice

- Forward procedure: increase B,
- Backward procedure: decrease B

until 'no significant change'.

Local summary

Smoothing by least squares:

- basis function method,
- linear smoother ($\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$),
- smoothing is achieved by $B = |base|$,
- there exist heuristics to choose B.

Localized least squares or kernel smoothing

Kernel smoothing

- Recall (linear smoother):

$$\hat{x}(t) = \sum_{j=1}^{n} S_j(t) y_j.$$

- Idea:
  - smoothness $\Rightarrow$ more weight to nearby t_j-s.

Kernel smoothing

- Recall (linear smoother):

$$\hat{x}(t) = \sum_{j=1}^{n} S_j(t) y_j.$$

- Idea:
  - smoothness $\Rightarrow$ more weight to nearby t_j-s.
  - (smoothing) kernels:

$$K_U(u) = 0.5 \times \chi_{[-1,1]}(u)\text{: uniform,}$$
$$K_Q(u) = 0.75(1-u^2) \times \chi_{[-1,1]}(u)\text{: quadratic,}$$
$$K_G(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\text{: Gaussian.}$$

Kernel smoothing: weight choice

Choose the weights in

$$\hat{x}(t) = \sum_{j=1}^{n} S_j(t) y_j$$

as

$$S_j(t) = K\left(\frac{t_j - t}{h}\right) : \text{ unnormalized},$$

$$S_j(t) = \frac{K\left(\frac{t_j - t}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{t_i - t}{h}\right)} : \text{Nadaraya-Watson estimator},$$

$$S_j(t) = \frac{1}{h}\int_{\bar{t}_{j-1}}^{\bar{t}_j} K\left(\frac{u - t}{h}\right) du : \text{Gasser-Müller estimator},$$

$\bar{t}_j = \frac{t_{j+1} + t_j}{2}$, $\bar{t}_0 = t_1$, $\bar{t}_n = t_n$. Last: nice formulas for K_Q.

Local summary

Linear smoother $\xrightarrow{\text{specifically}}$

- least squares method (basis function technique),
- kernel smoother.

Localized basis function estimator:

combine the 2 directions.

Localized basis function estimator

Localized basis function estimator

- Idea:
  - weighted least squares,
  - weight is
    - local: $\mathbf{c} = \mathbf{c}_t$,
    - determined by a smoothing kernel.
- Objective function ($\mathbf{c}_t := [c_1; \ldots; c_B]$):

$$J(\mathbf{c}_t) = \sum_{j=1}^{n} w_j(t) \left[y_j - \sum_{k=1}^{B} c_k \phi_k(t_j) \right]^2, \; w_j(t) = K\left(\frac{t_j - t}{h} \right).$$

- Prediction: $\hat{x}(t) = \langle \hat{\mathbf{c}}_t, \phi_t \rangle$.

Localized basis function estimator

- Objective [$\mathbf{W}_t = diag(w_j(t))$]:

$$J(\mathbf{c}_t) = \sum_{j=1}^{n} w_j(t)\left[y_j - \sum_{k=1}^{B} c_k \phi_k(t_j)\right]^2 = (\mathbf{y} - \mathbf{\Phi}\mathbf{c}_t)^T \mathbf{W}_t (\mathbf{y} - \mathbf{\Phi}\mathbf{c}_t).$$

- Solution (see weighted least squares):

$$\hat{\mathbf{c}}_t = (\mathbf{\Phi}^T \mathbf{W}_t \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{W}_t \mathbf{y},$$
$$\hat{x}(t) = \langle \phi_t, \mathbf{c}_t \rangle = \underbrace{\phi_t^T (\mathbf{\Phi}^T \mathbf{W}_t \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{W}_t}_{=S_t\text{: locally linear}} \mathbf{y}.$$

Localized basis function estimator

- Specifically:
  - $B = 1, \phi_1(t) \equiv 1$ gives the Nadaraya-Watson estimator.
  - In other words, it is a 'locally constant' technique.

Localized basis function estimator

- Specifically:
  - $B = 1, \phi_1(t) \equiv 1$ gives the Nadaraya-Watson estimator.
  - In other words, it is a 'locally constant' technique.
- More generally: locally polynomial estimators

$$J(\mathbf{c}_t) = \sum_{j=1}^{n} K\left(\frac{t_j - t}{h}\right)\left[y_j - \sum_{k=1}^{B} c_k (t - t_j)^k\right]^2.$$

  - superior behaviour @ boundaries,
  - adapts well to unequally spaced t_j-s.

Localized basis function methods

- Bandwith h: control of smoothness.
- Choice: some heuristics / visually.

Localized basis function methods

- Bandwith h: control of smoothness.
- Choice: some heuristics / visually.
- Properties (kernel smoothing, localised basis):
  - h – intuitive meaning.
  - kernel smoothing: instability around boundaries.
  - localised polynomials: better boundary behaviour.

Summary

- Linear smoother ($\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$):
  - basis + least squares,
  - kernel smoothing: e.g., Nadaraya-Watson.
- Locally-linear smoother ($\mathbf{S}_t\mathbf{y}$):
  - local polynomial smoothing $\xrightarrow{\text{specifically}}$ Nadaraya-Watson.

Summary

- Linear smoother ($\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$):
  - basis + least squares,
  - kernel smoothing: e.g., Nadaraya-Watson.
- Locally-linear smoother ($\mathbf{S}_t\mathbf{y}$):
  - local polynomial smoothing $\xrightarrow{\text{specifically}}$ Nadaraya-Watson.
- We covered Chapter 1-4 from [1].