Statistical methods of data science

An introduction to Functional Data Analysis

Alessandra Menafoglio[1*]

[1]MOX, Department of Mathematics, Politecnico di Milano

*alessandra.menafoglio@polimi.it

2. Smoothing and interpolation of functional data

Agenda

2. Smoothing and interpolation of functional data

2.1. Basis functions

2.2. Least square smoothing

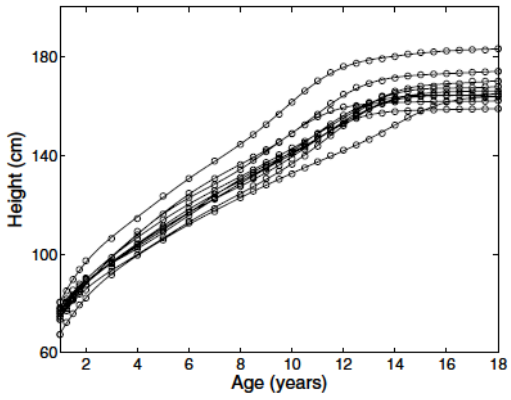
2.3. Smoothing with a differential penalization

2. Smoothing and interpolation of functional data

From raw observations to functional data

- Typical observations of functional data are **discrete** and **noisy**. Indeed, the record of each function x_i usually consists of n_i pairs (t_{ij}, y_{ij}), with $j=1,\ldots,n_i$.
- We model these pairs as $y_j = x(t_j) + \epsilon_j$.
  Note. The argument values t_{ij} may or may not be the same for each datum.
- For each i, we aim to reconstruct the underlying functional observation function x_i from the records (t_{ij}, y_{ij}), with $j=1,\ldots,n_i$
  Note: The assumptions on the properties of x_i (e.g., the smoothness) will reflect on the way we proceed to reconstruct the data

Example: Height of 10 girls in Berkeley Growth data



- Raw data are depicted as symbols
- Reconstructed functional data are depicted as lines

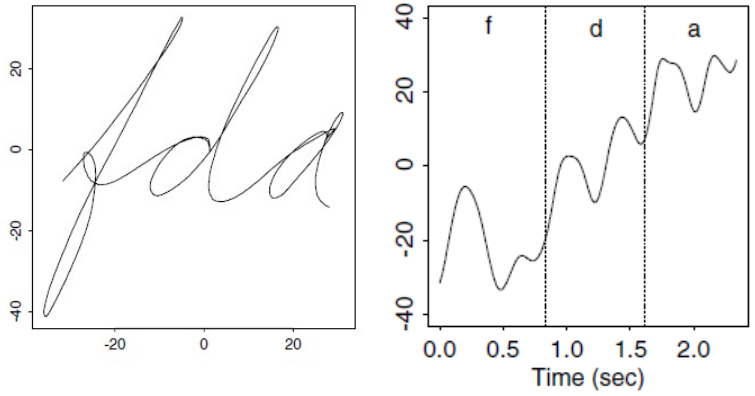
2. Smoothing and interpolation of functional data

From raw observations to functional data

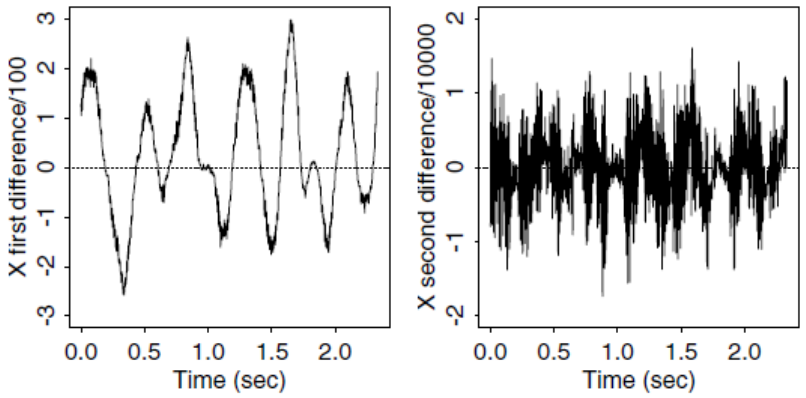
- Depending on our prior knowledge on the measurement error (i.e., on the properties of the noise ϵ_j), we can decide to perform
  - Interpolation: the functional form reconstructed interpolates its discrete observations (noiseless measurements)
  - Smoothing: the functional form is smoother than the actual observations (noisy measurement)

- In most cases, smoothing is preferred to interpolation.
  Note. Differential operations (i.e., derivatives) amplify the effect of noise. Smoothing raw data enhances the estimation of derivatives

Example: Hand writing data



Functional datum: X-coordinate in hand-writing



First and second derivatives estimated via finite-differences

If we aim to interpolate or smooth discrete data, we typically perform the following steps:

- Choose a **target functional form** for x_i , that possibly depends on parameters
- **Estimate** the functional form, based on the pairs (t_{ij}, y_{ij})

The **choice of the functional form** depends on various factors:

- **Features that we want to extract**: e.g., regularity of the functional form if the target is the differential information of the function (first, second derivatives)
- **Functional space embedding**: when we choose a Hilbert space embedding, we automatically identify possible orthonormal bases

In most cases:

- Hilbert space embedding is employed (especially, $H=L^2$)
- Functions are represented by basis functions

Agenda

2. Smoothing and interpolation of functional data

2.1. Basis functions

2.2. Least square smoothing

2.3. Smoothing with a differential penalization

Roughly speaking: a **system of basis functions** is a set of known functions that are linearly independent and allows us to approximate arbitrarily well any function as a linear combination of (a sufficiently large number of) *K* of these functions

Basis expansion of a functional observation

Given a system of basis functions ϕ_k , we will express a function x by the linear expansion

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t)$$

or in matrix notation

$$x = \mathbf{c}'\boldsymbol{\phi} = \boldsymbol{\phi}'\mathbf{c} \ .$$

Recall. In **Hilbert spaces**, we can always find an **orthonormal basis** that allows approximating, with any desired precision, any element of the space through the expansion

$$x = \sum_{n=1}^{K} \langle x, u_n \rangle u_n.$$

Note. In the following, we will mainly refer to L^2, although basis system and smoothing methods exist for other embedding space too.

2.1. Basis functions

Fourier basis functions

One of the most known basis expansion in L^2 is provided by the **Fourier series**

$$\hat{x}(t) = c_0 + c_1 \sin \omega t + c_2 \cos \omega t + c_3 \sin 2\omega t + c_4 \cos 2\omega t + \dots$$

i.e., with the previous notation

$$\phi_0(t) = 1, \phi_{2r-1}(t) = \sin r\omega t, \text{ and } \phi_{2r}(t) = \cos r\omega t.$$

Properties:

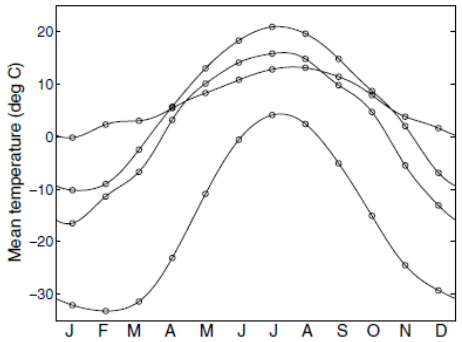
- The basis is periodic, of period $2\pi/\omega$
- If the values of t_j are equally spaced in T and the period is equal to the length of T than the basis is orthogonal (it can be made orthonormal via a proper rescaling)

Useful for:

- Extremely stable functions (i.e., no strong local features), for which uniformly smooth behavior is expected
- Periodic data

Inappropriate for:

- Discontinuous functions (or with discontinuous derivatives)

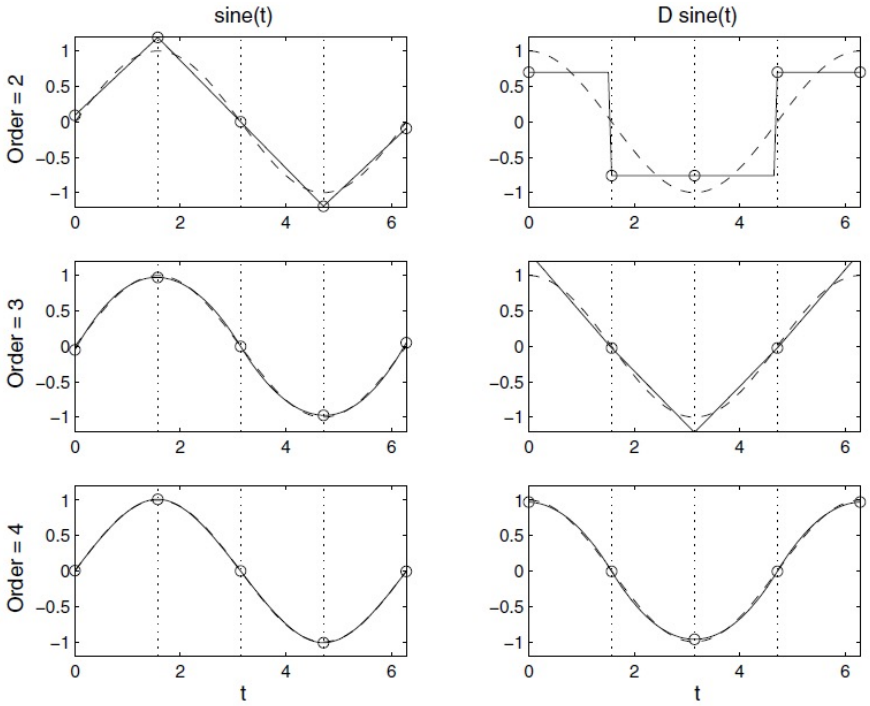


Temperatures in Canada

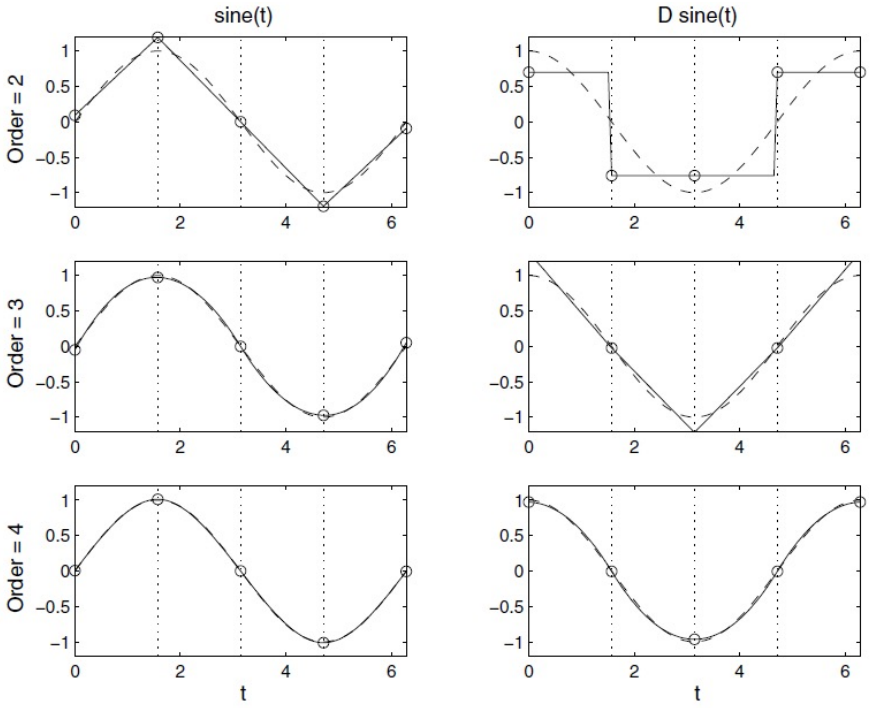
2.1. Basis functions

Spline functions

- Spline functions are widely-used as approximation system for **non-periodic functional data**
- Construction of a ***m-order spline***:
  - Divide the interval of definition *T* into *L* subintervals, i.e. fix a set of **knots**
  - Over each interval, the spline is defined as a polynomial or order *m (# of constant to define the polynomial)*
  - The polynomials are constrained as to guarantee that **adjacent polynomials join with continuity** in their values and in those of the derivatives up to order *m-2*



- To gain **flexibility** in a spline one can increase the number of its knots, e.g., by locating more knots where the function exhibits more variability
- The **number of parameters** required to define a spline function with non-overlapping knots is the order plus the number of interior knots *m+L-1*

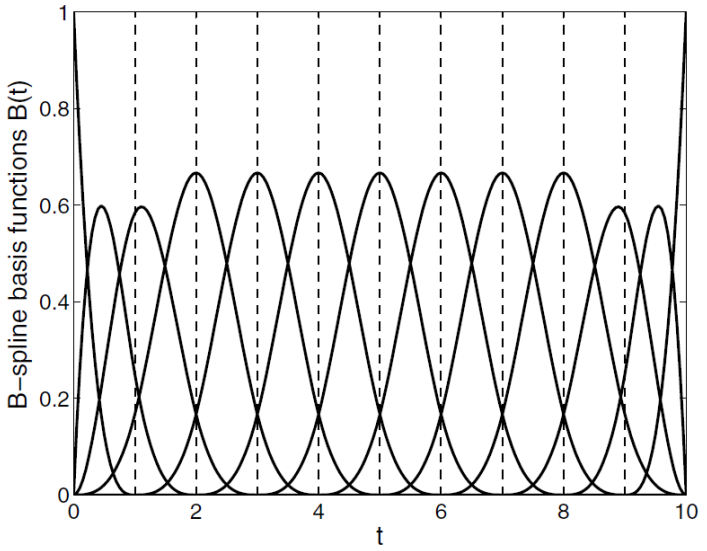


- B-spline basis functions are systems of spline basis functions ϕ_k with the key properties:
  - Each basis function is a spline
  - A linear combination of the basis elements is a spline function
  - Any spline function can be expressed as a linear combination of these basis functions

- We call $B_k(t, \tau)$ a B-spline basis function in t with sequence of knots τ .
- A spline function is then defined as

$$S(t) = \sum_{k=1}^{m+L-1} c_k B_k(t, \tau)$$

- **Smoothing splines**: knots are placed at each argument value



Agenda

2. Smoothing and interpolation of functional data

- 2.1. Basis functions
- 2.2. Least square smoothing
- 2.3. Smoothing with a differential penalization

2.2. Least square smoothing

- We defined basis systems, that allows us to express a functional datum as a linear combinations of these basis elements

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t)$$

- Our next goal is to estimate the parameters c_k from the observed pairs (t_j, y_j) under the model

$$y_j = x(t_j) + \epsilon_j ,$$

or, in matrix form

$$x = \mathbf{c}'\boldsymbol{\phi} = \boldsymbol{\phi}'\mathbf{c} \ .$$

Note 1. We can interpret this problem in the framework of classical linear models, and apply least square estimators.

Note 2. We smooth/interpolate one data point at a time; hence we here omit the index *i* of the statistical unit.

2.2. Least square smoothing

Ordinary least square fit

Goal:

Estimate the coefficient c_k of the linear model

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) + \epsilon_j$$

from the pairs (t_j, y_j)

- **Solution 1**: We minimize the sum of squared errors between fitted values and observations:

$$\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = \sum_{j=1}^{n} [y_j - \sum_{k}^{K} c_k \phi_k(t_j)]^2.$$

- From the theory of linear regression we know the solution of this problem

$$\hat{\mathbf{c}} = (\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'\mathbf{y}$$

$$\hat{\mathbf{y}} = \mathbf{\Phi}\hat{\mathbf{c}} = \mathbf{\Phi}(\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'\mathbf{y} \ .$$

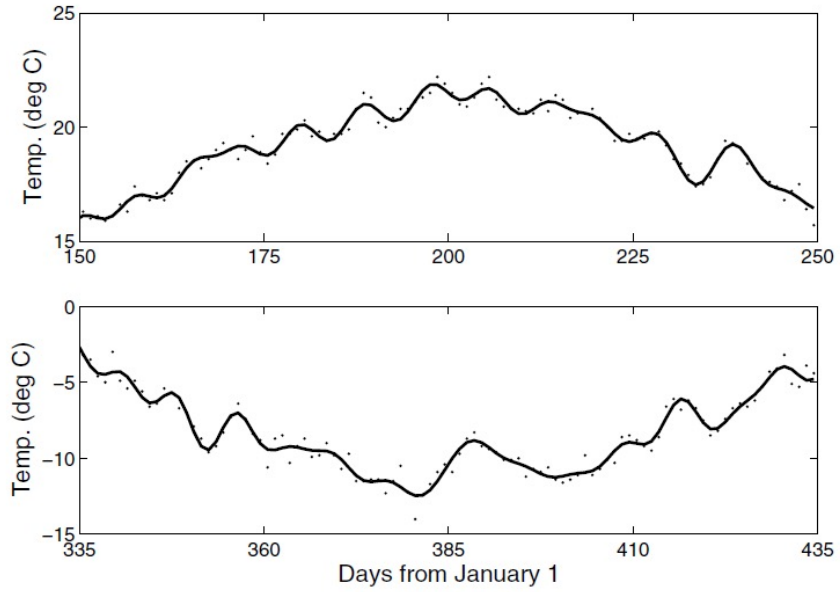
Note. $\hat{\mathbf{y}}$ is the projection of $\mathbf{y}$ over the space generated by the columns of $\mathbf{\Phi}$, that are the evaluations of the basis functions in the measurement points.

2.2. Least square smoothing

Ordinary least square fit

- Ordinary least squares are appropriate if the measurement error may be assumed to be iid
- The degree of smoothness of the estimated curve depends on the number of basis functions employed

Example. Smoothing of temperature data in Montreal using 109 Fourier basis functions



- We choose a Fourier basis because data are periodic
- We truncate the basis to 109 basis functions, that allows to catch (109-1)/2=54 different harmonic frequencies (about 1 per week)
- Performing OLS estimate means that the noise in the observations is iid across the days

If data are not iid (e.g., there is autocorrelation in the measurement process), we can use a weighted least squares.

- **Solution 2**: We minimize the weighted sum of squared errors between fitted values and observations:

$$\mathrm{SMSSE}(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})' \mathbf{W} (\mathbf{y} - \mathbf{\Phi c})$$

- Matrix $\mathbf{W}$ is assumed to be positive definite, and can be set e.g. to the covariance matrix of the errors

$$\mathbf{W} = \mathbf{\Sigma}_e^{-1} .$$

- The solution of this minimization problem is found as

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

- Approximate point-wise confidence intervals can be built based upon the estimated model
- As in classical linear models, the variance of the estimator for the coefficients is

$$\mathtt{Var}[\mathbf{c}] = (\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}'\mathbf{W}\mathbf{\Sigma}_e\mathbf{W}\mathbf{\Phi}(\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi})^{-1}$$

which in case of unweighted least squares and iid errors reduces to

$$\mathtt{Var}[\mathbf{c}] = \sigma^2(\mathbf{\Phi}'\mathbf{\Phi})^{-1}$$

- The variance of the point-wise estimate of the curve is then obtained as the diagonal of the matrix

$$\mathtt{Var}[\hat{\mathbf{y}}] = \mathbf{\Phi}\mathtt{Var}[\mathbf{c}]\mathbf{\Phi}'$$

which in case of unweighted least squares and iid errors reduces to

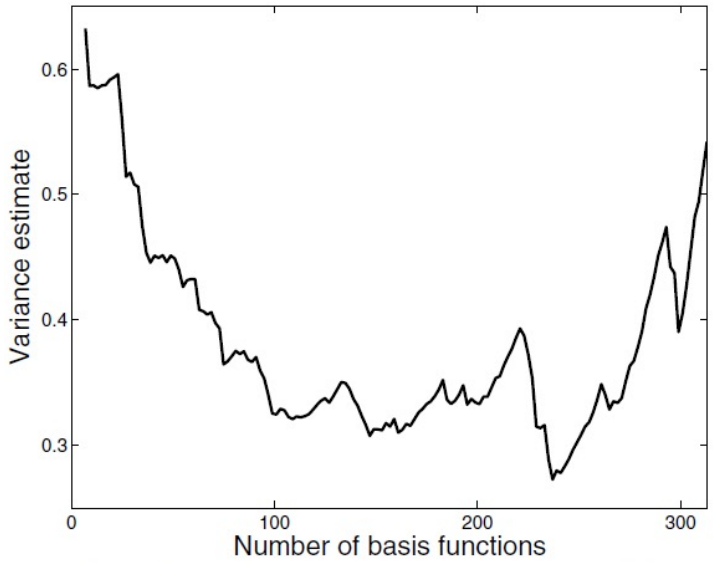
$$\mathtt{Var}[\hat{\mathbf{y}}] = \sigma^2\mathbf{\Phi}(\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}' = \sigma^2\mathbf{S}$$

- The variance of the errors can be estimated from the residual sum of squares

$$s^2 = \frac{1}{n-K}\sum_j^n (y_j - \hat{y}_j)^2$$

- To choose *K*, one may evaluate when a drop in sampling variance occurs for a range of candidate *K*

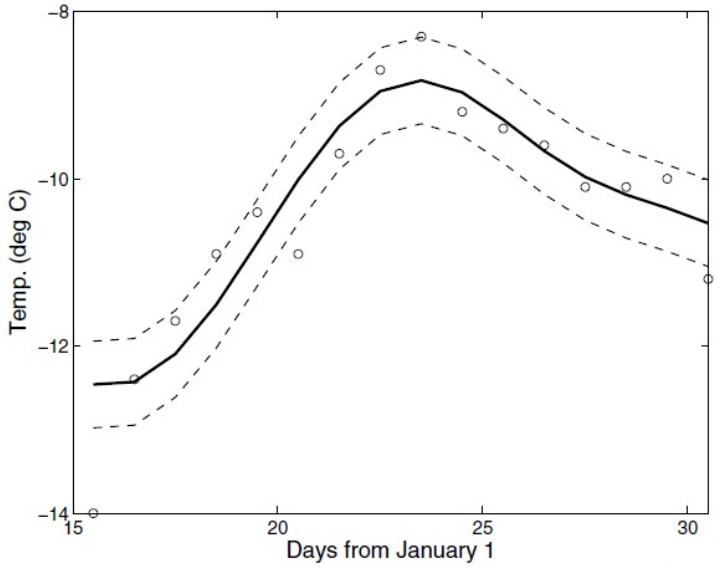
Example. Smoothing of temperature data in Montreal using 109 Fourier basis functions



- A drop in variance is obtained around 100 basis functions
- We truncated the basis to 109 basis functions, that allowed to catch (109-1)/2=54 different harmonic frequencies (about 1 per week)
- Lower variances may be obtained but overfitting might occurs then

- Based on previous expressions, approximate confidence limits may be built

Example. Smoothing of temperature data in Montreal using 109 Fourier basis functions



- Confidence limits are build summing/subtracting 2 standard deviations
- Quantiles of the normal can be used instead
- Confidence limits must be interpreted point-wise

- A key point of least square smoothing is how to set the order of the basis expansion. Algorithms to set *K* can be borrowed from the context of linear regression (e.g., step-wise algorithms). Nevertheless, one should pay close attention to the fact that:
  - The larger *K*, the better the fit to the data, but higher risk to fit the noise (or non-interesting variations)
  - If *K* is too small we may miss important features of the underlying function that we wish to estimate

- In fact, as in linear regression, we have a bias/variance trade-off

$$\texttt{Bias}[\hat{x}(t)] = x(t) - \mathrm{E}[\hat{x}(t)],$$

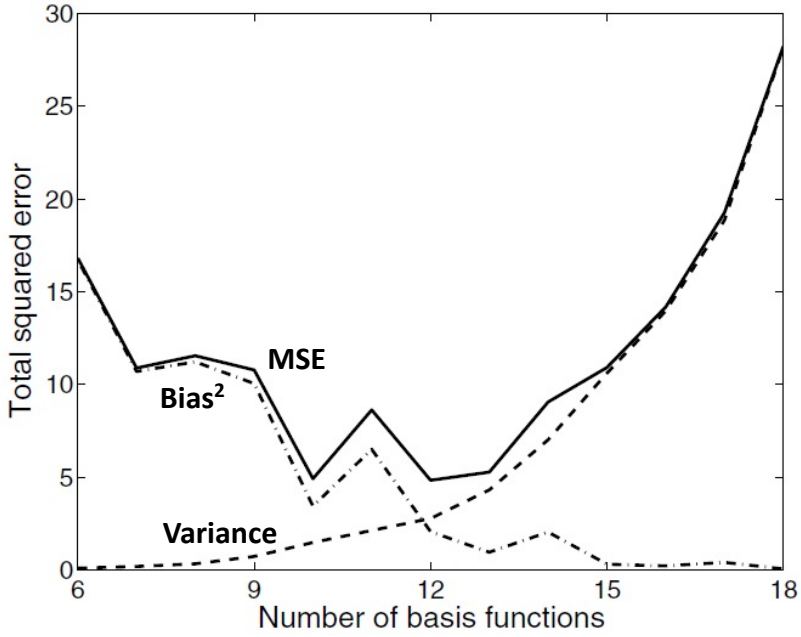
$$\texttt{Var}[\hat{x}(t)] = \mathrm{E}[\{\hat{x}(t) - \mathrm{E}[\hat{x}(t)]\}^2]$$

  - For large values of *K* the bias is small, the variance is high
  - For small values of *K*, the bias is high, the variance is low

- The mean-squared error summaries what we actually would like to minimize

$$\texttt{MSE}[\hat{x}(t)] = \mathrm{E}[\{\hat{x}(t) - x(t)\}^2] = \texttt{Bias}^2[\hat{x}(t)] + \texttt{Var}[\hat{x}(t)]$$

Example. Bias/variance trade of a simulated example inspired by the Berkeley Growth Study



Agenda

2. Smoothing and interpolation of functional data

2.1. Basis functions

2.2. Least square smoothing

2.3. Smoothing with a differential penalization

2.3. Smoothing with a differential penalization

Penalized regression

- We now focus on estimating a non-periodic function x on the basis of a vector $\mathbf{y}$ of discrete and noisy observations.
  Note: we are not (yet) assuming any functional form for x
- A way to circumvent the bias/variance trade-off is to impose a certain degree of smoothing on the curve (this reduces the variance at the expense of increasing bias)
- A popular way to do this is to quantify the notion of **roughness** through a **differential property of the curve**, and perform a **regression with** the corresponding **penalization**

Definition of penalized regression problem

Let us quantify *roughness* through the second derivative

$$\texttt{PEN}_2(x) = \int [D^2 x(s)]^2 \, ds$$

Measure of ***curvature*** of the function ($PEN_2(x)=0$ if x is a straight line)

Given λ, find x that minimizes

$$\texttt{PENSSE}_\lambda(x|\mathbf{y}) = [\mathbf{y} - x(\mathbf{t})]'\mathbf{W}[\mathbf{y} - x(\mathbf{t})]^2 + \lambda \times \texttt{PEN}_2(x)$$

Penalized SSE

- Let's give a closer look to the penalized SSE functional

$$\texttt{PENSSE}_\lambda(x|\mathbf{y}) = [\mathbf{y} - x(\mathbf{t})]'\mathbf{W}[\mathbf{y} - x(\mathbf{t})]^2 + \lambda \times \texttt{PEN}_2(x)$$

- Parameter λ is called *smoothing parameter* and controls the importance of the penalization with respect to the residual sum of squares:
  - If $\lambda \to \infty$ the functional gives emphasis to the penalization and the fitted curve will be a straight line ($\texttt{PEN}_2(x) = 0$)
  - If $\lambda \to 0$ the curve approaches the smoothest twice-differentiable curve that interpolates the data

- **Key result** (de Boor, 2002): the curve x that minimizes $\texttt{PENSSE}_\lambda(x|\mathbf{y})$ is a cubic spline with knots at the data points

  → *the functional form of* t_j *is a consequence of the objective functional!*

- **Common computational technique:** use a four order B-spline basis (called *cubic spline*) and minimize the $\texttt{PENSSE}_\lambda(x|\mathbf{y})$ with respect to the coefficients of the expansion of x

  λ set via GCV.

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t)$$

2.3. Smoothing with a differential penalization

Penalized regression – computational details

- We can re-express the penalization as

$$\begin{aligned} \texttt{PEN}_m(x) &= \int [D^m x(s)]^2 \, ds \\ &= \int [D^m \mathbf{c}'\phi(s)]^2 \, ds \\ &= \mathbf{c}'\mathbf{R}c \ , \end{aligned}$$

with $\mathbf{R} = \int D^m\phi(s) D^m\phi'(s) \, ds$.

- Plugging this expression in the objecting functional yields

$$\texttt{PENSSE}_m(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})'\mathbf{W}(\mathbf{y} - \mathbf{\Phi c}) + \lambda \mathbf{c}'\mathbf{R}\mathbf{c} \ .$$

that is minimized for

$$\hat{\mathbf{c}} = (\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi} + \lambda\mathbf{R})^{-1}\mathbf{\Phi}'\mathbf{W}\mathbf{y} \ .$$

Closing remarks

- We have seen basis expansions as a way to smooth raw functional observations
- Many other bases and smoothing techniques are available in the literature, e.g., Wavelet bases and Local polynomial smoothing

Local polynomial smoothing

LS smoothing on neighborhoods of the point t_j through polynomial basis $x(t_j) = \sum_{\ell}^{n} w_\ell y_\ell$.

$$\texttt{SMSSE}_t(\mathbf{y}|\mathbf{c}) = \sum_{j=1}^{n} w_j(t)[y_j - \sum_{k=1}^{K} c_k \phi_k(t_j)]^2,$$

$$w_\ell(t) = \texttt{Kern}\left(\frac{t_\ell - t_j}{h}\right)$$

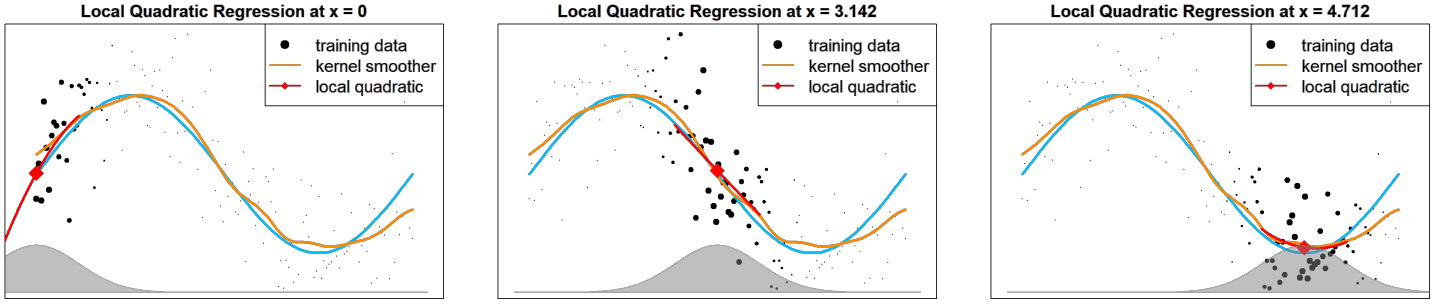
Uniform:	$\texttt{Kern}(u) = 0.5$ for $\lvert u\rvert \leq 1$,	0 otherwise
Quadratic:	$\texttt{Kern}(u) = 0.75(1-u^2)$ for $\lvert u\rvert \leq 1$,	0 otherwise
Gaussian:	$\texttt{Kern}(u) = (2\pi)^{-1/2}\exp(-u^2/2)$.	

Closing remarks

- We have seen basis expansions as a way to smooth raw functional observations
- Many other bases and smoothing techniques are available in the literature, e.g., Wavelet bases and Local polynomial smoothing

Local polynomial smoothing

LS smoothing on neighborhoods of the point t_j through polynomial basis $x(t_j) = \sum_{\ell}^{n} w_\ell y_\ell$.



Taken from R. Zhu (2024): " Statistical Learning and Machine Learning with R", teazrq.github.io/SMLR/

Closing remarks

- We have seen basis expansions as a way to smooth raw functional observations
- Many other bases and smoothing techniques are available in the literature, e.g., Wavelet bases and Local polynomial smoothing

- Ad hoc smoothing techniques need to be employed in case of constrained data, e.g.,
  - Monotonically increasing functions
  - Probability density functions

Smoothing or interpolation is the very first step of a functional data analysis and all the subsequent results depend on this step.
One should to pay close attention in applying the most appropriate technique for smoothing the data