SD 204: Beyond Simple Linear Models

Joseph Salmon
http://josephsalmon.eu
Télécom Paristech, Institut Mines-Télécom

Outline

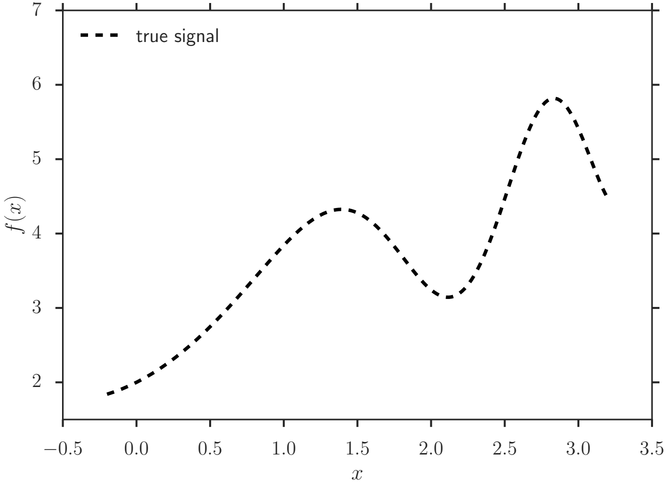
Generalizing linear models

Robustness

Table of Contents

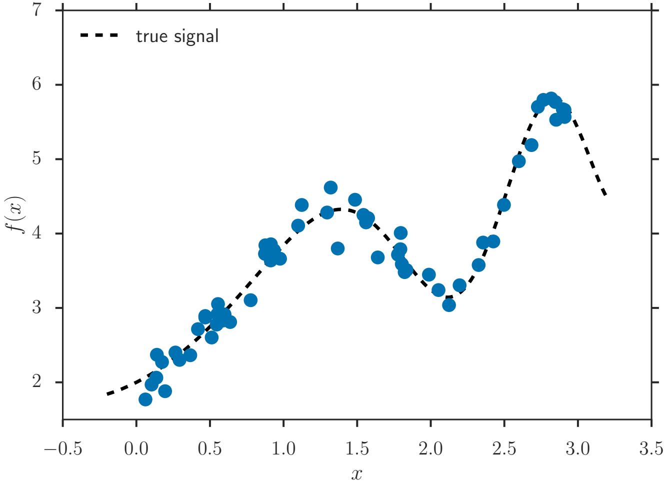
Limits of linear models

True signal: $f(x_i)$ for $i = 1, \ldots, n$



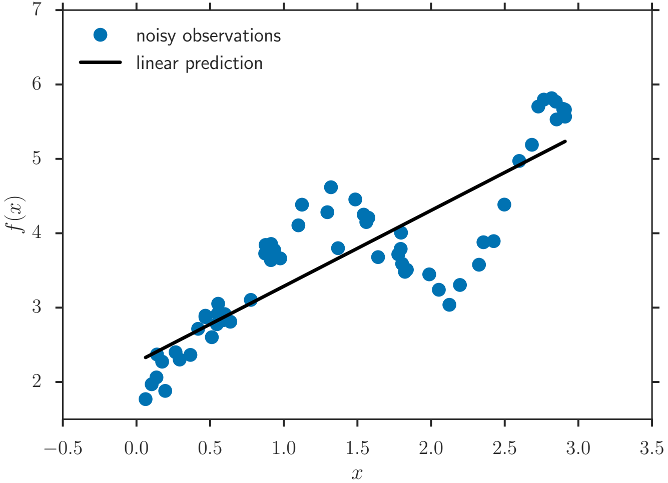
Limits of linear models

Noisy observations: $y_i = f(x_i) + \varepsilon_i$ for $i = 1, \ldots, n$



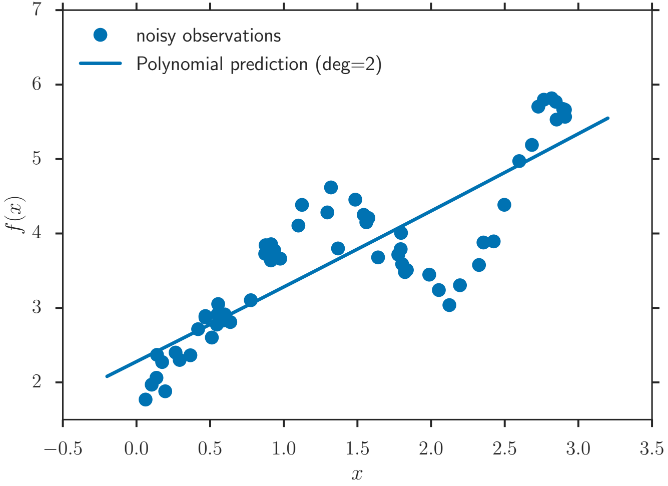
Limits of linear models

Linear model: not well suited here



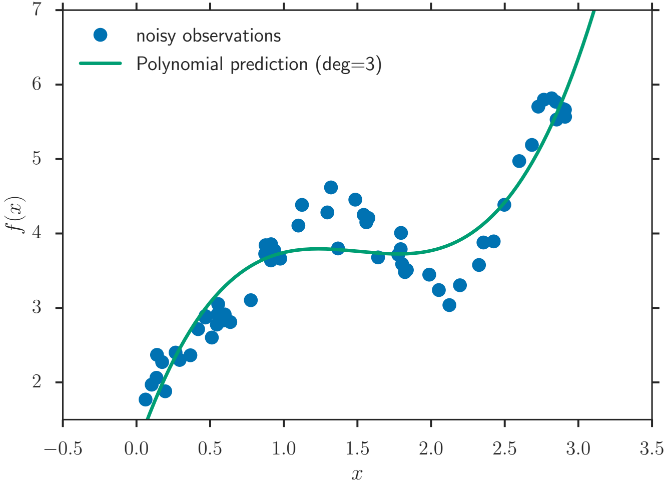
Limits of linear models

Polynomial model: better suited here



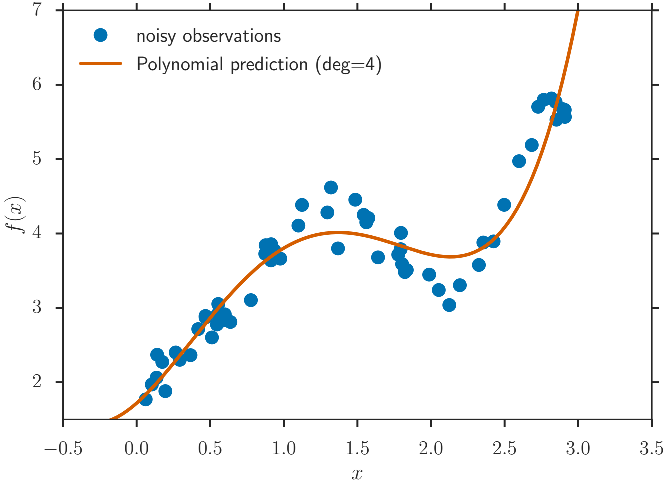
Limits of linear models

Polynomial model: better suited here



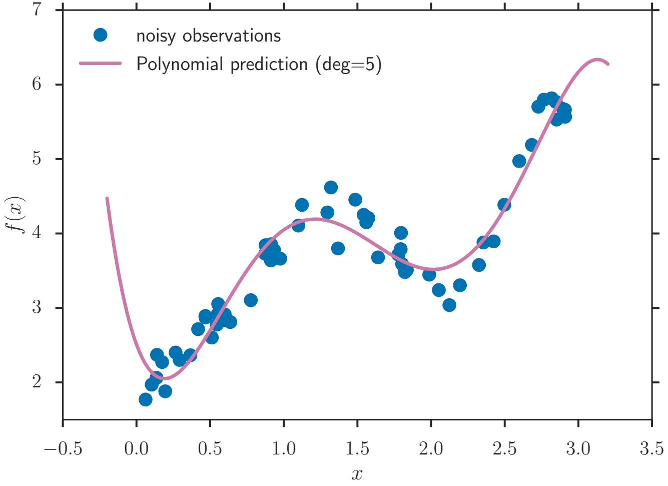
Limits of linear models

Polynomial model: better suited here



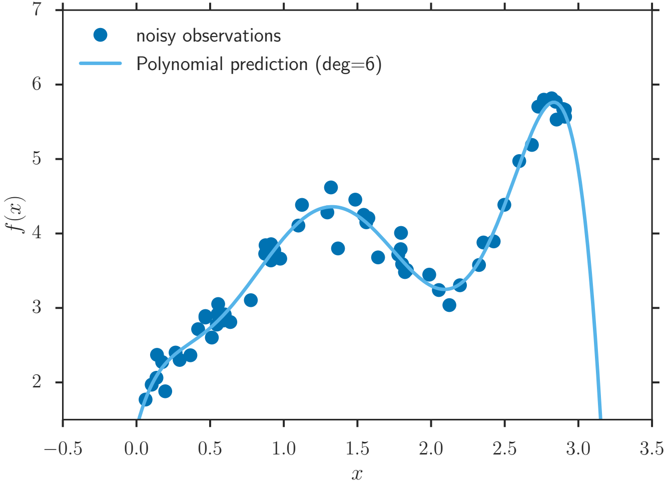
Limits of linear models

Polynomial model: better suited here



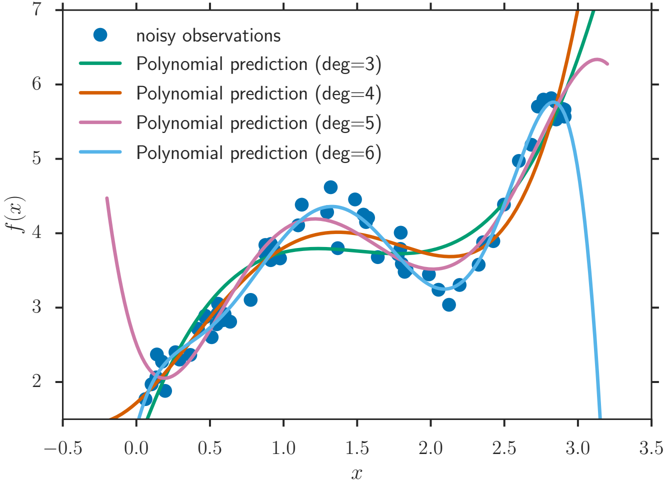
Limits of linear models

Polynomial model: better suited here



Limits of linear models

Polynomial model: better suited here



Polynomial modeling

Let D denote the degree of the polynomial:

$$y_i = \theta_0 + \sum_{d=1}^{D} \theta_d x_i^d$$

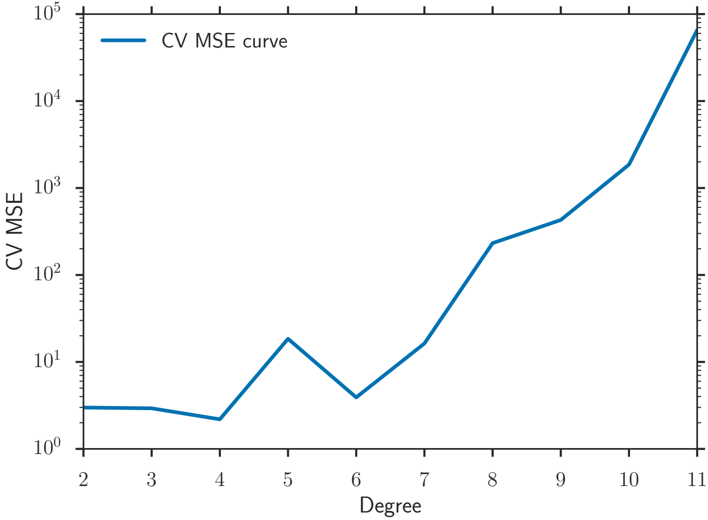
$$X = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^D \\ 1 & x_2 & x_2^2 & \dots & x_2^D \\ 1 & x_3 & x_3^2 & \dots & x_3^D \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^D \end{pmatrix}$$

Equivalently $X_{i,j} = x_i^{j-1}$ and $\boldsymbol{\theta} = (\theta_0, \dots, \theta_D)^\top \in \mathbb{R}^{D+1}$ and

$$\mathbf{y} \approx X\boldsymbol{\theta}$$

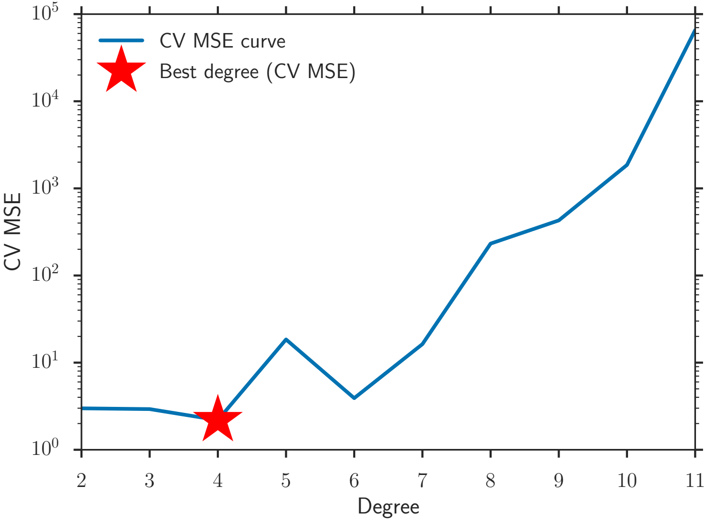
Choosing the degree

As often, one can use Cross-Validation (CV) to choose the degree



Choosing the degree

As often, one can use Cross-Validation (CV) to choose the degree



Pros/cons of polynomial regression

Pros

- Flexibility for low degree
- Useful for non-parametric estimation (*cf.*SD205, Fan and Gijbels (1996), Green and Silverman (1994))

Cons

- Polynomials are not local
- Size of the expanded data can be huge

Beyond one feature : $p = 2$ and $D = 2$

Let us consider a case where $x_i \in \mathbb{R}^2$
Hence $x_i = [a_i, b_i]$. The polynomial expansion of order 2 reads:

$$[1, a_i, b_i, a_i^2, a_i b_i, b_i^2]$$

The terms $a_i b_i$ represent interactions between feature 1 and 2

It can be modeled in a compact manner:

$$\begin{aligned} y_i =& \theta_0 + \theta^\top x_i + \frac{1}{2} x_i^\top \Theta x_i + \varepsilon_i \\ =& \theta_0 + \sum_{j=1}^{p} \theta_j x_{i,p} + \frac{1}{2} \sum_{1 \leqslant j \leqslant k \leqslant p} \Theta_{j,k} x_{i,j} x_{i,k} + \varepsilon_i \end{aligned}$$

where Θ is a $p \times p$ (symmetric) matrix

Beyond one feature : $p = 2$ and $D = 3$

Let us consider a case where $x_i \in \mathbb{R}^2$
Hence $x_i = [a_i, b_i]$. The polynomial expansion of order 3 reads:

$$[1, a_i, b_i, a_i^2, a_i b_i, b_i^2, a_i^3, a_i^2 b_i, a_i b_i^2, b_i^3]$$

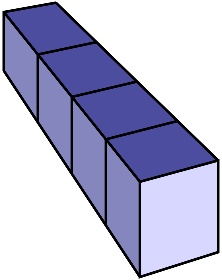
The terms $a_i b_i c_i$ represent interactions between feature 1, 2 and 3

It can be modeled in a compact manner:

$$\begin{aligned} y_i =& \theta_0 + \sum_{j=1}^{p} \theta_j x_{i,p} + \frac{1}{2} \sum_{1 \leqslant j \leqslant k \leqslant p} \Theta_{j,k} \cdot x_{i,j} x_{i,k} \\ &+ \frac{1}{6} \sum_{1 \leqslant j \leqslant k \leqslant \ell \leqslant p} \boldsymbol{\Theta}_{j,k,\ell} \cdot x_{i,j} x_{i,k} x_{i,\ell} + \varepsilon_i \end{aligned}$$

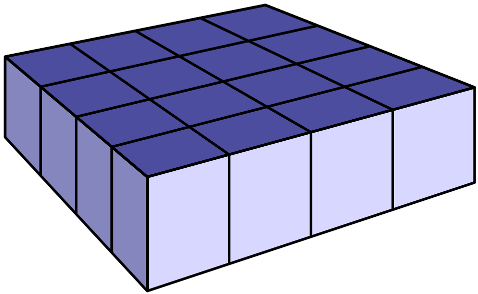
where Θ is a $p \times p$ matrix, $\boldsymbol{\Theta}$ is a $p \times p \times p$ tensor (symmetric)

Tensor representation 1D



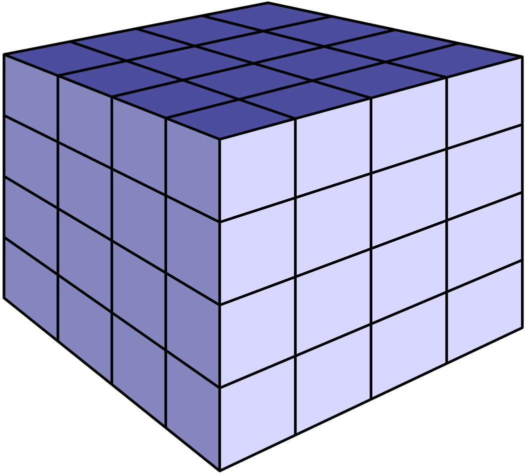
Vector case

Tensor representation 2D



Matrix case

Tensor representation 3D



General case

Splines (🇫🇷 : *cerces*)

Definition: Splines

A **spline** f is piecewise-polynomial function on an interval $[a,b], f : [a,b] \to \mathbb{R}$, composed of n subintervals $[x_{i-1}, x_i]$ with $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$. The restriction of f to an interval $[x_{i-1}, x_i]$ is a polynomial $P_i : [x_{i-1}, x_i] \to \mathbb{R}$, so that

$$
\begin{aligned}
f(x) =& P_1(x), \quad x_0 \leqslant t < x_1 \\
f(x) =& P_2(x), \quad x_1 \leqslant t < x_2 \\
& \vdots \\
f(x) =& P_i(x), \quad x_{n-1} \leqslant t \leqslant x_n.
\end{aligned}
$$

The highest order of the polynomials P_i is the **order** of the spline f, and the x_i's are called the **knots**

Rem:cubic most popular (*i.e.*,third degree) splines
Rem:generally smooth splines targeted (C^0, C^1, C^2, etc.)

Usage

- statistics
- computer science, *cf.*Bézier curves in Inkscape and other vector graphics softwares

Usage

- statistics
- computer science, *cf.*Bézier curves in Inkscape and other vector graphics softwares
- numerical analysis

Usage

- statistics
- computer science, *cf.*Bézier curves in Inkscape and other vector graphics softwares
- numerical analysis
- etc.

Usage

- statistics
- computer science, *cf.*Bézier curves in Inkscape and other vector graphics softwares
- numerical analysis
- etc.

Algorithms

Standard spline fitting when observing points (x_i, y_i), $i = 1, \ldots, n$: look for the spline with least curvature, *i.e.*,solve

$$\hat{f} \triangleq SP_\lambda(\mathbf{y}) \in \underset{f \text{is a spline}}{\arg\min} \left(\sum_{i=1}^{n} (f(x_i) - y_i)^2 + \lambda \int_a^b |f''(t)|^2 dt \right)$$

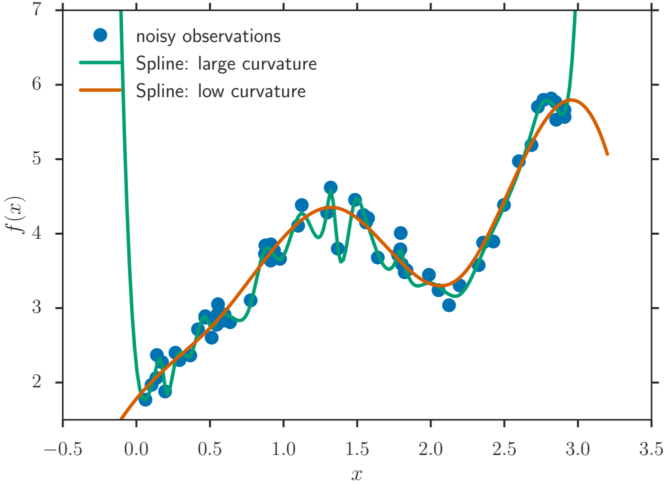
Fact: the solution is reached for a cubic spline, and can be obtained by a regularized least square, with $\Omega \in \mathbb{R}^{n \times n}$

$$\arg\min_g \|\mathbf{y} - g\|^2 + \lambda g^\top \Omega g$$

See details in Ch. 2, Green and Silverman (1994)

Notes: with the regularization used the spline obtained has the x_i as knots

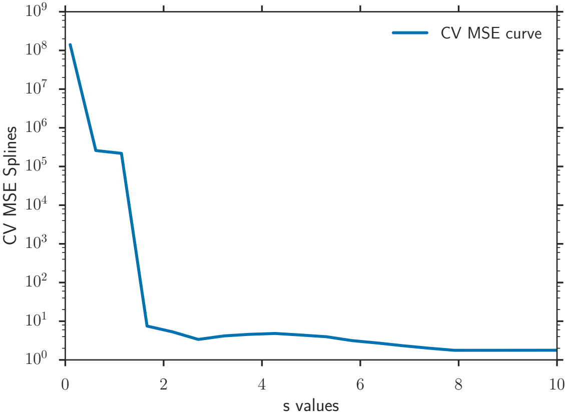
Visual



noisy observations
Spline: large curvature
Spline: low curvature
7
6
5
4
3
2
$f(x)$
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
x

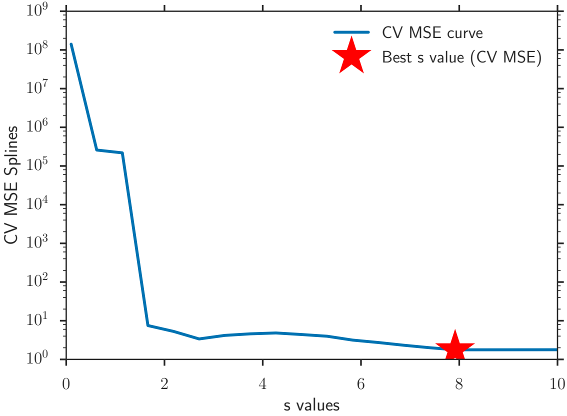
Choosing the smoothing parameter

One can use Cross-Validation to choose the smoothing parameter



Choosing the smoothing parameter

One can use Cross-Validation to choose the smoothing parameter



MSE Spline $= 0.2498$ vs. MSE Polynomials $= 2.1899$

Additive Models for regression

With ε_i modeling noise, the model reads

$$y_i = \sum_{j=1}^{p} f_j(x_{i,j}) + \varepsilon_i$$

or equivalently:

$$\mathbf{y} = \sum_{j=1}^{p} f_j(\mathbf{x}_j) + \boldsymbol{\varepsilon}$$

Rem:possibly one of the f_j encodes the intercept

Rem:GAM extend to general linear model, *e.g.*,logistic regression, $g(y_i) = \sum_{j=1}^{p} f_j(x_{i,j})$, with g a link function

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Initialize: $f_1 \equiv 0, \ldots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Initialize: $f_1 \equiv 0, \ldots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)

while *not converged* **do**

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Initialize: $f_1 \equiv 0, \ldots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)

while *not converged* **do**

  for $j = 1, \ldots, p$ **do**

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \dots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Initialize: $f_1 \equiv 0, \dots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)

while *not converged* **do**

  for $j = 1, \dots, p$ **do**

    $\mathbf{r} \leftarrow \mathbf{r} + f_j(x_j))$ `// Partial residual update`

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Initialize: $f_1 \equiv 0, \ldots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)

while *not converged* **do**

  for $j = 1, \ldots, p$ **do**

    $\mathbf{r} \leftarrow \mathbf{r} + f_j(x_j))$ `// Partial residual update`

    $f_j \leftarrow SP_{\lambda_j}(\mathbf{r})$ `// update with spline (param. ` λ_j `)`

Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \dots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$

Initialize: $f_1 \equiv 0, \dots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)

while *not converged* **do**

  for $j = 1, \dots, p$ **do**

    $\mathbf{r} \leftarrow \mathbf{r} + f_j(x_j))$ `// Partial residual update`

    $f_j \leftarrow SP_{\lambda_j}(\mathbf{r})$ `// update with spline (param.` λ_j`)`

    $\mathbf{r} \leftarrow \mathbf{r} - f_j(x_j))$ `// Partial residual un-update`

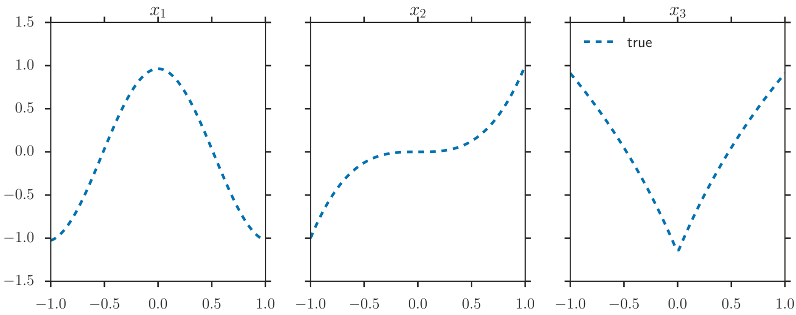
Back-fitting

Algorithm: Back-fitting Additive models

Input : $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$, $\mathbf{y} \in \mathbb{R}^n$
Initialize: $f_1 \equiv 0, \ldots, f_p \equiv 0$ and $\mathbf{r} = \mathbf{y}$ (residual)
while *not converged* **do**
    for $j = 1, \ldots, p$ **do**
        $\mathbf{r} \leftarrow \mathbf{r} + f_j(x_j))$ `// Partial residual update`
        $f_j \leftarrow SP_{\lambda_j}(\mathbf{r})$ `// update with spline (param.` λ_j`)`
        $\mathbf{r} \leftarrow \mathbf{r} - f_j(x_j))$ `// Partial residual un-update`
Output: $f_1, \ldots, f_p$

Rem:Back-fitting is a kind of coordinate descent method where a loop is run over each feature.

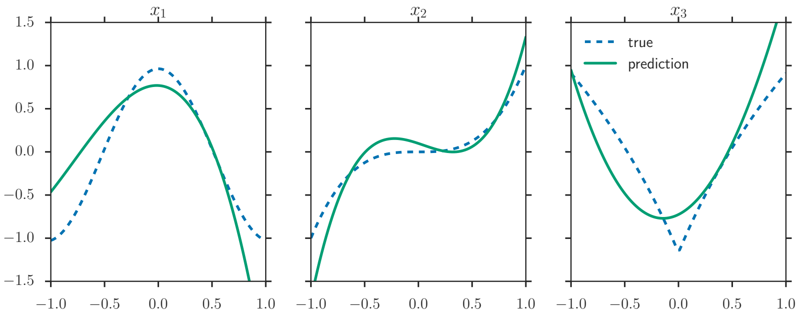
GAM in action



where $\mathbf{y} = f(\mathbf{x}) + \varepsilon$ with $f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + f_3(x_3)$ and

$$f_1(x) = \cos(3x)$$
$$f_2(x) = x^3$$
$$f_3(x) = 3\log(1 + |x|)$$

GAM in action



where $\mathbf{y} = f(\mathbf{x}) + \varepsilon$ with $f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + f_3(x_3)$ and

$$f_1(x) = \cos(3x)$$
$$f_2(x) = x^3$$
$$f_3(x) = 3\log(1 + |x|)$$

Pros and cons of GAM

Pros

- automatically model non-linear effect
- easy interpretation / visualization thanks to 1D functions

Cons

- non-convex optimization / algorithm (local minima, initialization, stopping criterion, etc.)
- hard to tune : at least one parameter by feature

More information

- More details on GAMs: https://vimeo.com/125940125
- play with the code (*cf.*course website)

More information

- More details on GAMs: https://vimeo.com/125940125
- play with the code (*cf.*course website)
- Standard textbook: Hastie and Tibshirani (1990)

More information

- More details on GAMs: https://vimeo.com/125940125
- play with the code (*cf.*course website)
- Standard textbook: Hastie and Tibshirani (1990)

Table of Contents

Least-squares paternity

Adrien-Marie Legendre:
"Nouvelles méthodes pour la détermination des orbites des comètes", 1805

Carl Friedrich Gauss:
"Theoria Motus Corporum Coelestium in sectionibus conicis solem ambientium" 1809

And before...

Definition

The Least Absolute Deviation (LAD) estimator:

$$(\hat{\boldsymbol{\theta}}) \in \underset{\boldsymbol{\theta} \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} |y_i - x_i^\top \boldsymbol{\theta}|$$

where we write $X = [x_1, \ldots, x_n]^\top$ (row description)

Rem: harder to optimize than least-squares, non-smooth optimization (*i.e.*,non-differentiable function)

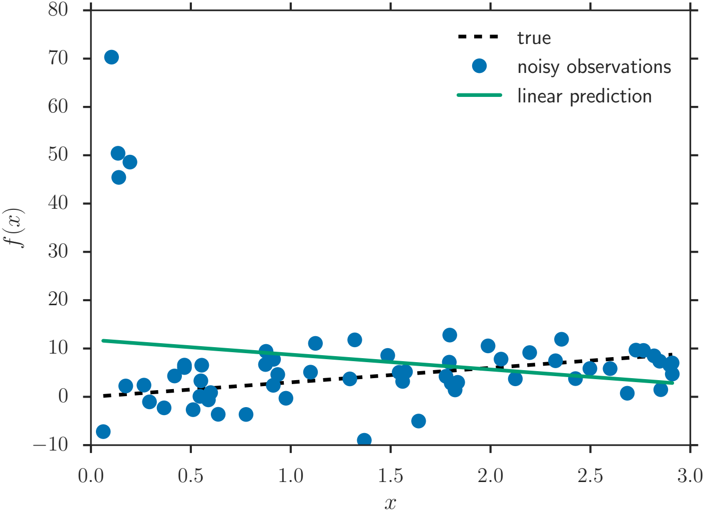
Rem: estimator less sensitive to **outliers** (than OLS/Ridge/Lasso, etc.), *e.g.*,observations where ε_i is large

Least absolute deviation paternity

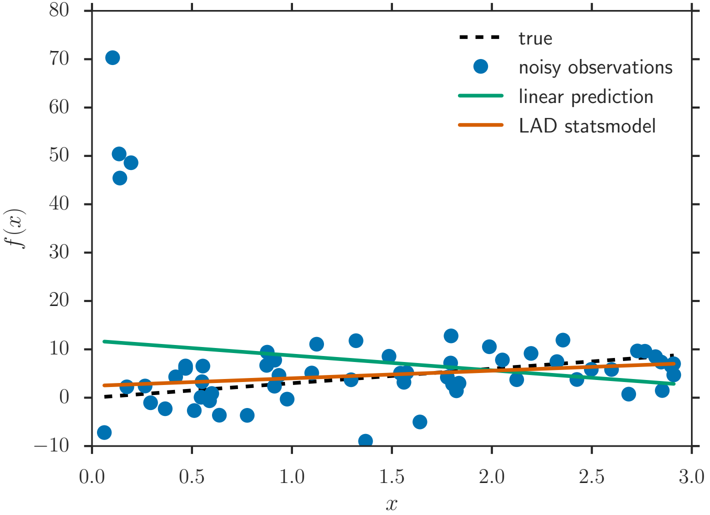
Ruđer Josip Bošković:"???", 1757

Pierre-Simon de Laplace
"Traité de mécanique céleste", 1799

LAD in action



LAD in action



References I

- J. Fan and I. Gijbels.

  Local polynomial modelling and its applications, volume 66 of *Monographs on Statistics and Applied Probability*.

  Chapman & Hall, London, 1996.

- P. J. Green and B. W. Silverman.

  Nonparametric regression and generalized linear models, volume 58 of *Monographs on Statistics and Applied Probability*.

  Chapman & Hall, London, 1994.

  A roughness penalty approach.

- T. J. Hastie and R. J. Tibshirani.

  Generalized additive models, volume 43.

  CRC press, 1990.