Statistical Methods - Nonparametric Regression

Lecture 6

Mattias Villani

Sveriges Riksbank and Stockholm University

April 30, 2010

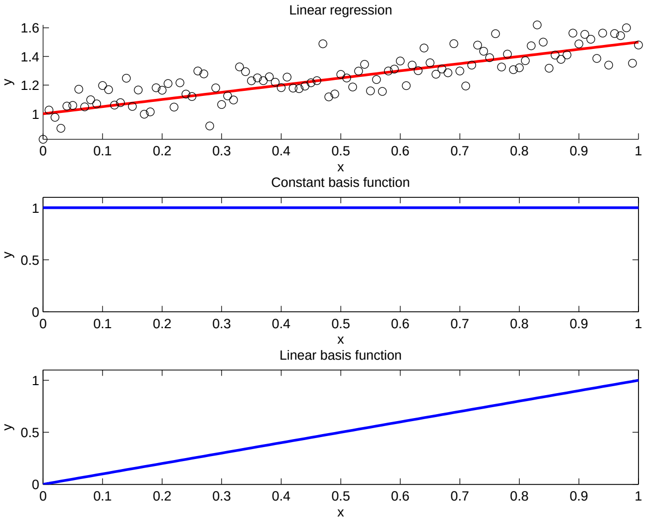
Overview Lecture 6

- Splines, all day long ...

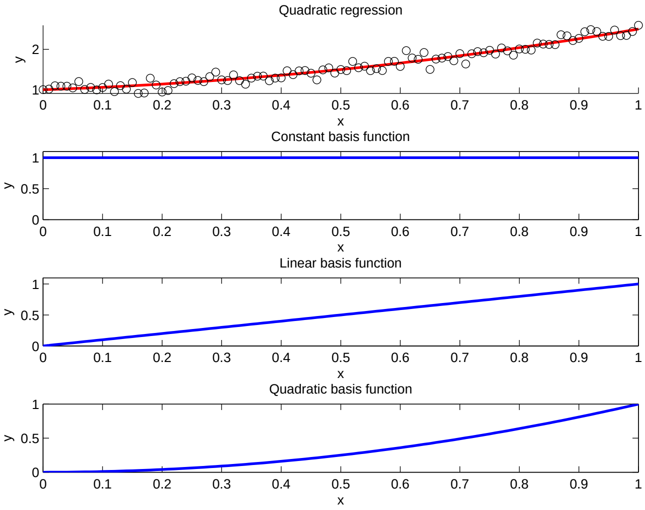
- The problem with polynomials are their global nature; changing an observation y_i may affect the fit $(\hat{y}_j)$ of another observation even if x_j is far from x_i.
- Kernel regression tries to solve this by fitting local polynomials at every x. But kernel regression is hard to generalize to situations with more than a few covariates, at least when we want to allow for interactions between covariates.
- Polynomials are linear combinations of basis functions $1, x, x^2, ..., x^k$

$$E(y|x) = \beta_0 \cdot 1 + \beta_1 \cdot x + \beta_2 \cdot x^2 + ...\beta_k \cdot x^k$$

Basis functions of the linear model



Basis functions of the quadratic model



Splines - a single knot

- Splines can be viewed as a generalization of polynomials that includes also truncated polynomial terms, e.g.

$$E(y|x) = \beta_0 + \beta_1 \cdot x + b_1 \cdot (x - \kappa)_+$$

where

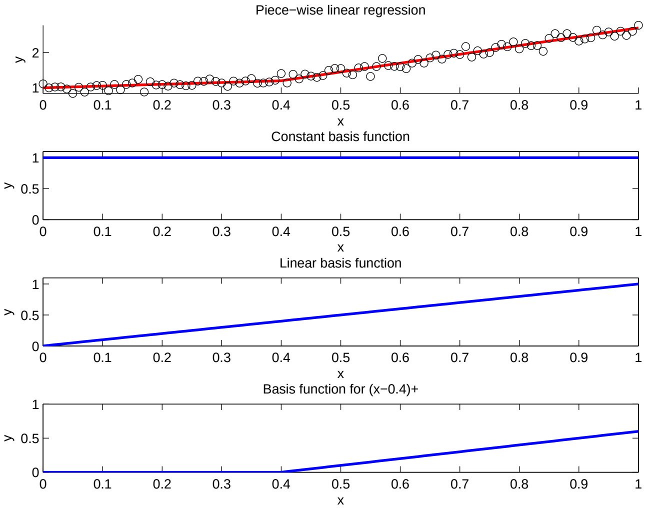
$$(x-\kappa)_+ = \begin{cases} 0 & \text{if } x \leq \kappa \\ x - \kappa & \text{if } x > \kappa \end{cases}$$

- κ is a **knot**.
- Data matrix

$$\underset{n \times 3}{X} = \begin{bmatrix} 1 & x_1 & (x_1 - \kappa)_+ \\ \vdots & \vdots & \vdots \\ 1 & x_n & (x_n - \kappa)_+ \end{bmatrix}$$

1	1.0000	0	0
2	1.0000	0.0500	0
3	1.0000	0.1000	0
4	1.0000	0.1500	0
5	1.0000	0.2000	0
6	1.0000	0.2500	0
7	1.0000	0.3000	0
8	1.0000	0.3500	0
9	1.0000	0.4000	0
0	1.0000	0.4500	0.0500
1	1.0000	0.5000	0.1000
2	1.0000	0.5500	0.1500
3	1.0000	0.6000	0.2000
4	1.0000	0.6500	0.2500
5	1.0000	0.7000	0.3000
6	1.0000	0.7500	0.3500
7	1.0000	0.8000	0.4000
8	1.0000	0.8500	0.4500
9	1.0000	0.9000	0.5000
0	1.0000	0.9500	0.5500
1	1.0000	1.0000	0.6000

Basis functions of the broken-stick model



Splines - multiple knots

- Data matrix
- Spline with many knots

$$E(y|x) = \beta_0 + \beta_1 \cdot x + b_1 \cdot (x - \kappa_1)_+ + ... + b_K \cdot (x - \kappa_K)_+$$

- Data matrix

$$\underset{n\times(2+K)}{X} = \begin{bmatrix} 1 & x_1 & (x_1 - \kappa_1)_+ & (x_1 - \kappa_2)_+ & \cdots & (x_1 - \kappa_K)_+ \\ \vdots & \vdots & \vdots & & & \vdots \\ 1 & x_n & (x_n - \kappa_1)_+ & (x_n - \kappa_2)_+ & \vdots & (x_n - \kappa_K)_+ \end{bmatrix}$$

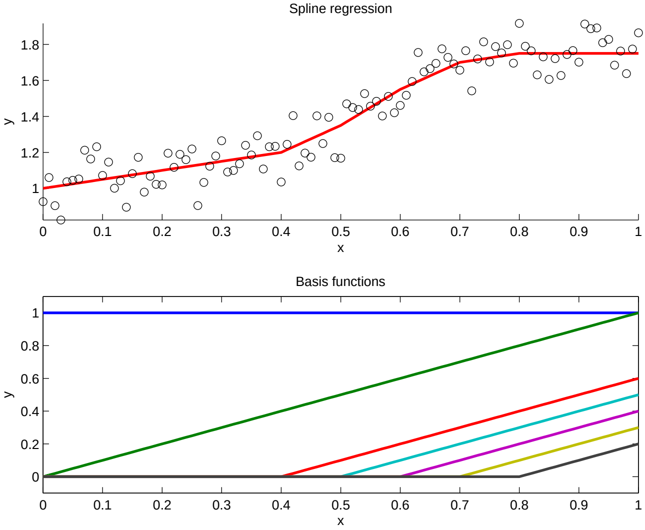
- From now on, let use write the model as

$$y = X\beta + \varepsilon,$$

with X defined as above.

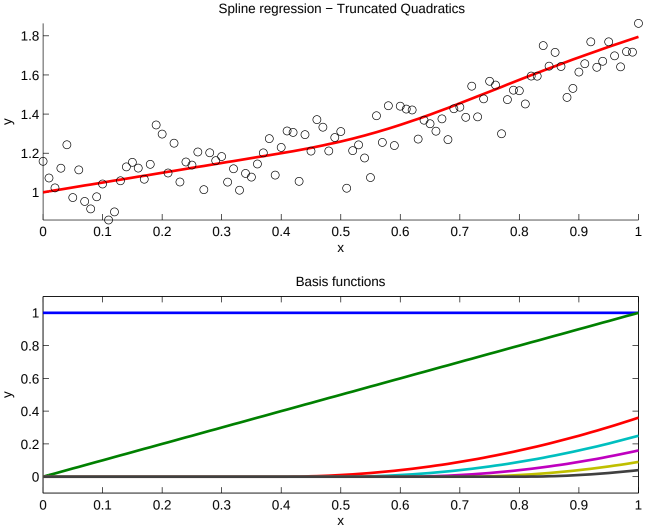
1.0000	0	0	0	0	0	0
1.0000	0.0500	0	0	0	0	0
1.0000	0.1000	0	0	0	0	0
1.0000	0.1500	0	0	0	0	0
1.0000	0.2000	0	0	0	0	0
1.0000	0.2500	0	0	0	0	0
1.0000	0.3000	0	0	0	0	0
1.0000	0.3500	0	0	0	0	0
1.0000	0.4000	0	0	0	0	0
1.0000	0.4500	0.0500	0	0	0	0
1.0000	0.5000	0.1000	0	0	0	0
1.0000	0.5500	0.1500	0.0500	0	0	0
1.0000	0.6000	0.2000	0.1000	0	0	0
1.0000	0.6500	0.2500	0.1500	0.0500	0	0
1.0000	0.7000	0.3000	0.2000	0.1000	0	0
1.0000	0.7500	0.3500	0.2500	0.1500	0.0500	0
1.0000	0.8000	0.4000	0.3000	0.2000	0.1000	0
1.0000	0.8500	0.4500	0.3500	0.2500	0.1500	0.0500
1.0000	0.9000	0.5000	0.4000	0.3000	0.2000	0.1000
1.0000	0.9500	0.5500	0.4500	0.3500	0.2500	0.1500
1.0000	1.0000	0.6000	0.5000	0.4000	0.3000	0.2000

Basis functions of spline with piece-wise linear basis

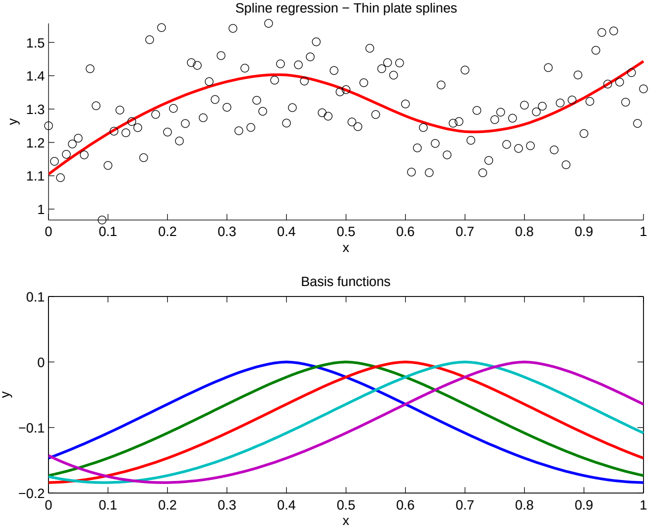


- We need to decide:
  - how many knots to include (K)
  - the location of the K knots
  - the form of the spline basis functions
- Alternative forms of the spline basis functions
  - Truncated linear $(x-\kappa)_+$. Continuos, but discontinuous first-derivative.
  - Truncated polynomial $(x-\kappa)_+^p$. Continuous derivatives up to order $p-1$.
  - Thin-plate $(x-\kappa)^2 \ln|x-\kappa|$
- Typical practice: use many knots (10-20) and spread them evenly over the x-axis, or at quantiles (10%,20%,...,90%) of x. Problem: **over-fitting**.

Truncated quadratic basis



Thin plate basis



- We can keep all the knots, but restrain their influence.
- Constrained least-squares fit:

$$\min_{\beta}(y - X\beta)'(y - X\beta) \text{ subject to } \beta' D\beta < C,$$

where

$$D = \begin{bmatrix} \mathbf{0}_{(p+1)\times(p+1)} & \mathbf{0}_{(p+1)\times K} \\ \mathbf{0}_{K\times(p+1)} & \mathbf{I}_{K\times K} \end{bmatrix}$$

so there are no constaints on the (global) polynomial terms.

- This is equivalent to the Lagrange problem

$$\min_{\beta}(y - X\beta)'(y - X\beta) + \lambda^{2p}\beta' D\beta$$

which has the solution

$$\hat{\beta}_{\lambda} = (X'X + \lambda^{2p}D)^{-1}X'y.$$

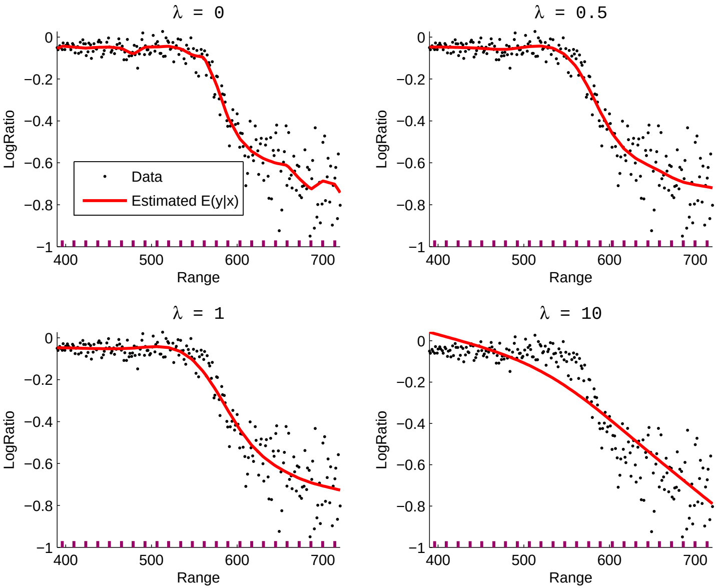
- $\hat{\beta}_\lambda$ can be interpreted as the posterior mode of β under the prior (let's assume $p = 1$)

$$\beta \sim N(0, \lambda^{-2} D^{-1}),$$

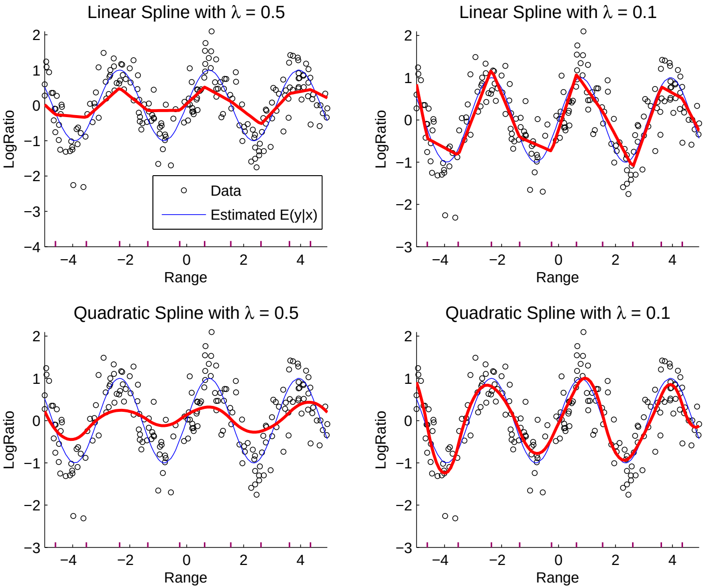
so λ^2 is the prior precision of each the coefficients on the spline terms. Since the prior mean of β is zero, all spline coefficients are shrunk toward zero. This avoids over-fitting.

- We can also attempt to estimate λ. Just put prior on λ (Inv-χ^2 makes life easy) and do Gibbs sampling.
- We will talk more about how to choose λ in the last part of the course. Cross-validation is one option.

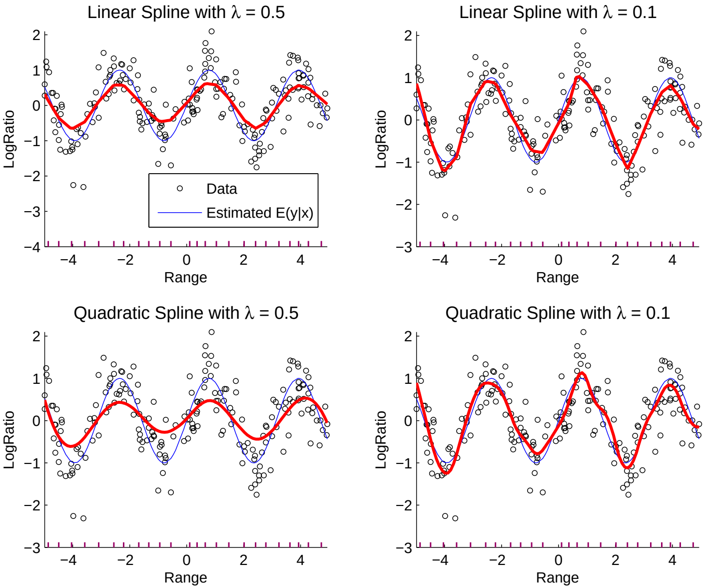
Lidar data - Truncated linear splines with 24 knots



Linear vs quadratic splines - Sine data. 10 knots



Linear vs quadratic splines - Sine data. 24 knots



Quantifying the flexibility of a spline?

- Splines are also linear smoothers:

$$\hat{y} = X\hat{\beta}_\lambda = X(X'X + \lambda^{2p}D)^{-1}X'y = Ly$$

  with

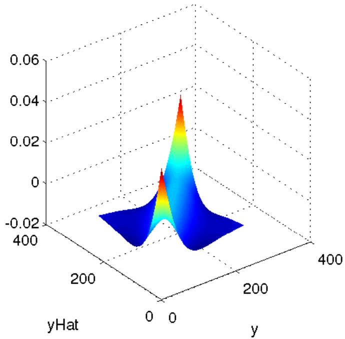
$$L = X(X'X + \lambda^{2p}D)^{-1}X'$$

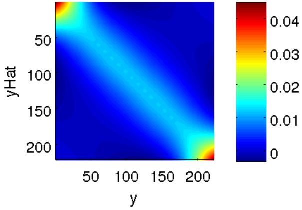
- Is S_λ symmetric? Yes. Is S_λ idempotent? No.
- The larger λ gives more smoothing, but how much more?
- Degrees of freedom:

$$\nu = \mathrm{tr}S_\lambda$$

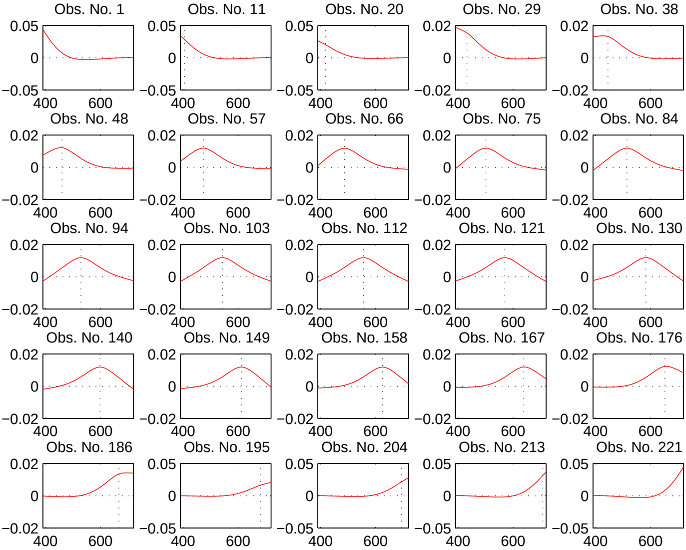
- A spline with $\nu = \mathrm{tr}S_\lambda$ degrees of freedom has approximately the same flexibility as a polynomial of degree $\nu - 1$.
- It is to see that $p + 1 < \nu < p + 1 + K$.

The smoother matrix - Lidar data $\lambda = 5$

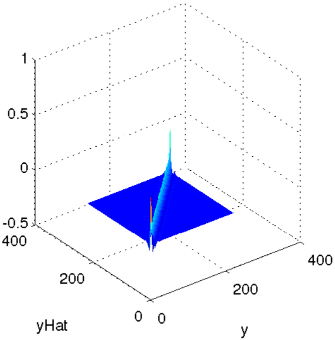


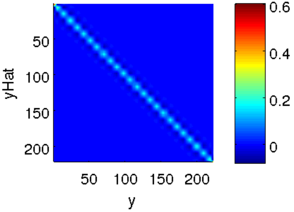


The equivalent kernel - Lidar data $\lambda = 5$

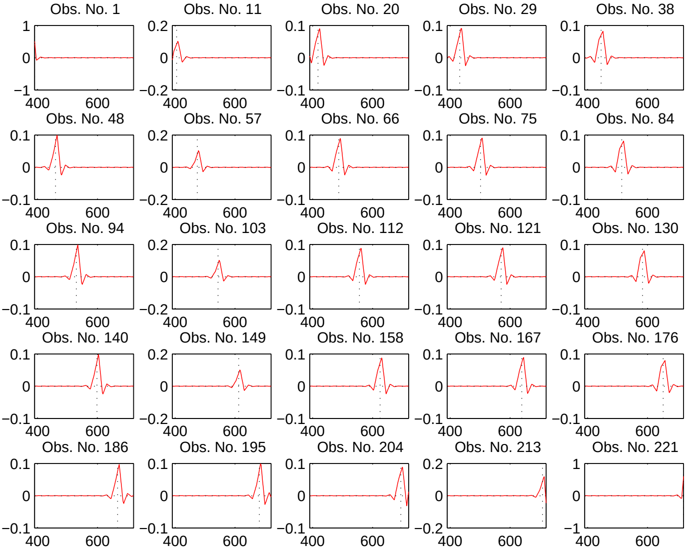


The smoother matrix - Lidar data $\lambda = 0$





The equivalent kernel - Lidar data $\lambda = 0$



- What is the expected error of the estimator $\hat{f}(x)$ of $f(x)$?
- Mean Squared Error at x

$$\mathrm{MSE}[\hat{f}(x)] = \mathrm{E}\left[\hat{f}(x) - f(x)\right]^2 = \left[\mathrm{E}\hat{f}(x) - f(x)\right]^2 + \mathrm{Var}[\hat{f}(x)]$$

- If we care about the accuracy of the whole curve we can use Mean Integrated Squared Error (MISE)

$$\mathrm{MISE}(\hat{f}) = \int_{\mathcal{X}} \mathrm{MSE}[\hat{f}(x)]dx$$

or Mean Summed Squared Errors (MSSE)

$$\mathrm{MSSE}[\hat{f}(x)] = \mathrm{E}\sum_{i=1}^{n}[\hat{f}(x_i) - f(x_i)]^2$$

- For linear smoothers with homoscedastic errors

$$\mathrm{MSSE}[\hat{f}(x)] = \|(L - I)f\|^2 + \sigma_\varepsilon^2 \mathrm{tr}(LL')$$

which shows the **Bias-Variance trade-off**. Extreme case: $L = I$, so $\hat{y} = y$, then Bias=0 and Variance=$\sigma_\varepsilon^2 n$.