Modelling Non-linear and Non-stationary Time Series

From Smoothing to Conditional Parametric Models

Henrik Madsen

Advanced Time Series Analysis

September 2016

- Previously
  Local constant estimate
- Alternative/extension
  Local polynomial approximation
- In the following
  - Regressogram (later - in Chapter 3)
  - Locally-weighted polynomial regression
  - Conditional parametric models
  - Conditional parametric ARX-models
  - Non-linear FIR and ARX models
- Smoothing splines

Goals of smoothing

Besides providing a nice summary of a set of data points, *scatterplot smoothers* can be viewed as estimates of the regression function:

$$f(x) = \mathbb{E}[Y|X = x]$$

Sometimes in addition, the following model is useful:

$$Y = f(x) + \epsilon$$

Where ϵ is a random variable with mean zero.
Data: N pairs $(x_i, y_i), \quad i = 1, 2, \ldots, N$.
Goal: estimate f.

One-dimensional Smoothers

Four broad categories:

- Series or regression smoothers (polynomials, Fourier regression, regression splines, filtering)
- Kernel Smoothers (Nadaraya-Watson locally weighted averages, local regression, loess)
- Smoothing splines (roughness penalties)
- Near Neighbor smoothers (running means, medians, Tukey smoothers)

Locally-weighted polynomial regression

- Previously (in Section 2.3.3), the local model, $Y_t = \theta + \epsilon_t$ was estimated as:

$$\arg\min_{\theta} \frac{1}{N} \sum_{s=1}^{N} w_s(\boldsymbol{x})(Y_s - \theta)^2$$

- Extension: LWLS - Locally weighted least squares
  Model:

$$Y_t = \mu(\boldsymbol{X}_t) + \epsilon_t$$

Polynomial $P(\boldsymbol{X}_t, \boldsymbol{x})$ such that $\mu(\boldsymbol{X}_t) = P(\boldsymbol{X}_t, \boldsymbol{x})$

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}) = \arg\min_{\theta} \frac{1}{N} \sum_{s=1}^{N} w_s(\boldsymbol{x}) \left(Y_s - P(\boldsymbol{X}_s, \boldsymbol{x})\right)^2$$

The local estimate of $\mu(\boldsymbol{x})$ is $\hat{\mu}(\boldsymbol{x}) = \hat{P}(\boldsymbol{x}, \boldsymbol{x})$.

Locally-weighted polynomial regression –Example

$$P(X_t, x) = \theta_0 + \theta_1(X_t, x) + \theta_2(X_t - x)^2$$

$\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2)^T$ is estimated by

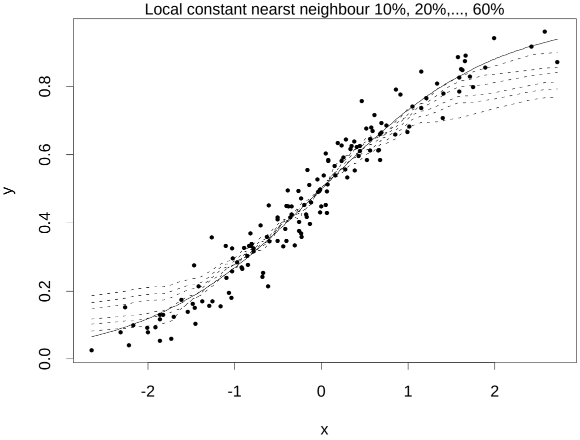
$$\hat{\theta}(x) = \arg\min_{\theta} \frac{1}{N} \sum_{s=1}^{N} w_s(x) \left(Y_s - P(X_s, x)\right)^2$$

The local estimate of μ is:

$$\hat{\mu}(x) = \hat{\theta}_0(x)$$

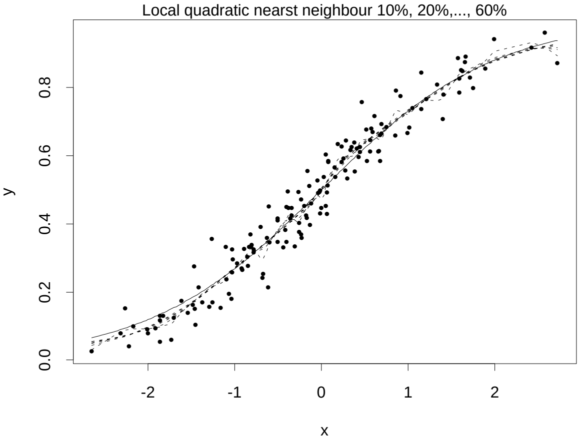
Locally-weighted polynomial regression

LWLS estimate using nearest neighbor.
First a local constant model is applied.



Locally-weighted polynomial regression

Then using local quadratic approximation.



The bias is much larger for the constant mean model.

Conditional parametric models

Also called Varying-coefficient models. The coefficients depend on a number of variables.

Conditional parametric model:

$$Y_t = \boldsymbol{Z}_t^T \boldsymbol{\theta}(\boldsymbol{X}_t) + \epsilon_t, \quad t = 1, \ldots, N$$

Estimated locally by:

$$\arg\min_{\theta} \frac{1}{N} \sum_{s=1}^{N} w_s(\boldsymbol{x}) \left(Y_s - \boldsymbol{Z}_s^T \boldsymbol{\theta}(\boldsymbol{x}) \right)^2$$

where $\boldsymbol{\theta}(x)$ are polynomials.

Conditional parametric models –Example

Locally weighted linear model: Assume $Z_t^T = (1, Z_{1t}), \quad \dim(x) = 1$
Constant setting $\theta^T = (\theta_0, \theta_1)$
Now

$$Z_t^T \theta(X_t) = (1, Z_{1t}) \begin{pmatrix} \theta_0(X) \\ \theta_1(X) \end{pmatrix}$$
$$= (1, Z_{1t}) \begin{pmatrix} \theta_{00} + \theta_{01} X_t + \theta_{02} X_t^2 \\ \theta_{10} + \theta_{11} X_t + \theta_{12} X_t^2 \end{pmatrix}$$

Conditional parametric models –example

This can be re-written as:

$$Z_t^{*T}\theta^* = \left(1, X_t, X_t^2, Z_{1t}, Z_{1t}X_t, Z_{1t}X_t^2\right) \begin{pmatrix} \theta_{00} \\ \theta_{01} \\ \theta_{02} \\ \theta_{10} \\ \theta_{11} \\ \theta_{12} \end{pmatrix}$$

After estimation:

$$\hat{\theta}_0(x) = \hat{\theta}_{00}(x) + \hat{\theta}_{01}(x)x + \hat{\theta}_{02}(x)x^2$$
$$\hat{\theta}_1(x) = \hat{\theta}_{10}(x) + \hat{\theta}_{11}(x)x + \hat{\theta}_{12}(x)x^2$$

Two important conditional parametric models

- Conditional parametric ARX-models

$$Y_t = \sum_{i=1}^{p} a_i(X_{t-m})Y_{t-i} + \sum_{i=1}^{r} b_i(X_{t-m})U_{t-i} + \epsilon_t$$

- Functional-coefficient AR-models

$$Y_t = \sum_{i=1}^{q} a_i(Y_{t-i})Y_{t-i} + \epsilon_t$$

where $\{\epsilon_t\}$ are white noise processes in both cases.

Smoothing Splines

- Consider the object function

$$RRS(f,\lambda) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int \{f''(t)\}^2 dt$$

  where λ is a fixed smoothing parameter.
- Notice the extremes obtained for $\lambda = 0$ and $\lambda = \infty$.
- It can be shown that the unique optimizer is a natural spline with knots at the unique values of $x_i, i = 1, ..., N$.
- Over-parameterized since with have N knots which implies N degrees of freedom.

Smoothing Splines (cont.)

- Let us write the natural splines as

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j$$

- Now we write

$$RRS(\theta, \lambda) = (\boldsymbol{y} - \boldsymbol{N}\theta)^T(\boldsymbol{y} - \boldsymbol{N}\theta) + \lambda\theta^T\Omega_N\theta$$

  where $\boldsymbol{N}_{ij} = N_j(x_i)$ and $\Omega_{ik} = \int N_j''(t)N_k''(t)dt$.
- The solution is easily seen to be

$$\hat{\theta} = (\boldsymbol{N}^T\boldsymbol{N} + \lambda\Omega_N)^{-1}\boldsymbol{N}^T\boldsymbol{y} \quad (1)$$

- The fitted smoothing spline is

$$\hat{f}(x) = \sum_{j=1}^{N} N_j(x)\hat{\theta}_j \quad (2)$$

Smoothing splines (cont.)

- Let $\boldsymbol{f}$ denote the N-vector of fitted values at the training predictors x_i. Then

$$\hat{\boldsymbol{f}} = \boldsymbol{N}(\boldsymbol{N}^T\boldsymbol{N} + \lambda\Omega_N)^{-1}\boldsymbol{N}^T\boldsymbol{y} \quad (3)$$
$$= \boldsymbol{S}_\lambda \boldsymbol{y} \quad (4)$$

- The matrix $\boldsymbol{S}_\lambda$ is known as the **smoother matrix**. Important: $\boldsymbol{S}_\lambda$ is not idempotent!
- Degrees of freedom is $df_\lambda = \mathrm{tr}\boldsymbol{S}_\lambda$.

Smoothing splides (cont.)

- Consider $\boldsymbol{B}_\xi$ to be a $N \times M$ matrix (ie. M cubic-spline basis functions evaluated at the N training points.
- Then the vector of fitted values is given by

$$\hat{\boldsymbol{f}} = \boldsymbol{B}_\xi(\boldsymbol{B}_\xi^T \boldsymbol{B}_\xi)^{-1}\boldsymbol{B}_\xi^T \boldsymbol{y} \quad (5)$$

$$= \boldsymbol{H}_\xi \boldsymbol{y} \quad (6)$$

  where $\boldsymbol{H}_\xi$ is a **projection operator** or **hat matrix** (idempotent).
- Degrees of freedom is $df = \mathrm{tr}\boldsymbol{H}_\xi = M$.