Statistical Learning

Tommaso Proietti

DEF Tor Vergata

Local Polynomial Regression and Kernel Smoothing

Local polynomial regression

- Consider the regression model $Y = f(X) + \epsilon$, where $f(X)$ is an unknown deterministic function of the input X, so that $\mathrm{E}(Y|X) = f(X)$.
- A training sample of observations $(y_i, x_i), i = 1, \ldots, N$, is available. Suppose that we are interested in estimating the value of the function at a point x_0, $f(x_0)$.
- If $f(X)$ is "smooth" (i.e. continuous and differentiable), it can be locally approximated around x_0 by a polynomial of degree p, using a Taylor expansion:

$$\tilde{f}(x) = \beta_0 + \beta_1(x - x_0) + \cdots + \beta_p(x - x_0)^p.$$

It is clear from setting $x = x_0$, that $\beta_0 = f(x_0)$, $\beta_1 = f'(x_0)$, and in general $\beta_j = f^{(j)}(x_0)/j!$ (the scaled j-th derivative of the function).

- As $\tilde{f}(x)$ is valid only as a local approximation, the $p+1$ unknown coefficients $\beta_k, k = 0, \ldots, p$, are estimated by weighted least squares (WLS), giving more weight to the observations with values of x that are close to x_0.
- The WLS objective function to be minimized is:

$$\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)\left(y_i - \hat{\beta}_0 - \hat{\beta}_1(x_i - x_0) - \ldots - \hat{\beta}_p(x_i - x_0)^p\right)^2,$$

where $K\left(\frac{x-x_0}{h}\right)$ is a *kernel function*, depending on a bandwidth parameter, $h > 0$, providing a set of weights declining with the distance of x from x_0.

Denoting

$$\boldsymbol{X}_0 = \begin{bmatrix} 1 & (x_1 - x_0) & \cdots & (x_1 - x_0)^p \\ 1 & (x_2 - x_0) & \cdots & (x_2 - x_0)^p \\ \vdots & \vdots & \cdots & \vdots \\ 1 & (x_i - x_0) & \cdots & (x_i - x_0)^p \\ \vdots & \vdots & \cdots & \vdots \\ 1 & (x_N - x_0) & \cdots & (x_N - x_0)^p \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}.$$

$$\boldsymbol{K}_0 = \text{diag}\left(K\left(\frac{x_1 - x_0}{h}\right), K\left(\frac{x_2 - x_0}{h}\right), \ldots, K\left(\frac{x_N - x_0}{h}\right)\right),$$

the WLS estimator is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}_0' \boldsymbol{K}_0 \boldsymbol{X}_0)^{-1} \boldsymbol{X}_0' \boldsymbol{K}_0 \boldsymbol{y}.$$

The fitted value is then $\hat{y}_0 = \hat{\beta}_0$.

The fit is determined by 3 quantities:

- The polynomial order, p.
- The kernel function.
- The bandwidth parameter, h. The bandwidth determines the width of the local neighbourhood.

Kernels

A kernel is a symmetric and non negative function with a maximum at zero. There is a large literature on kernels and their properties.

- The uniform kernel is $K(u) = I(|u| \leq 1)$;
- The Epanechnikov kernel has certain optimality properties and is defined as follows:

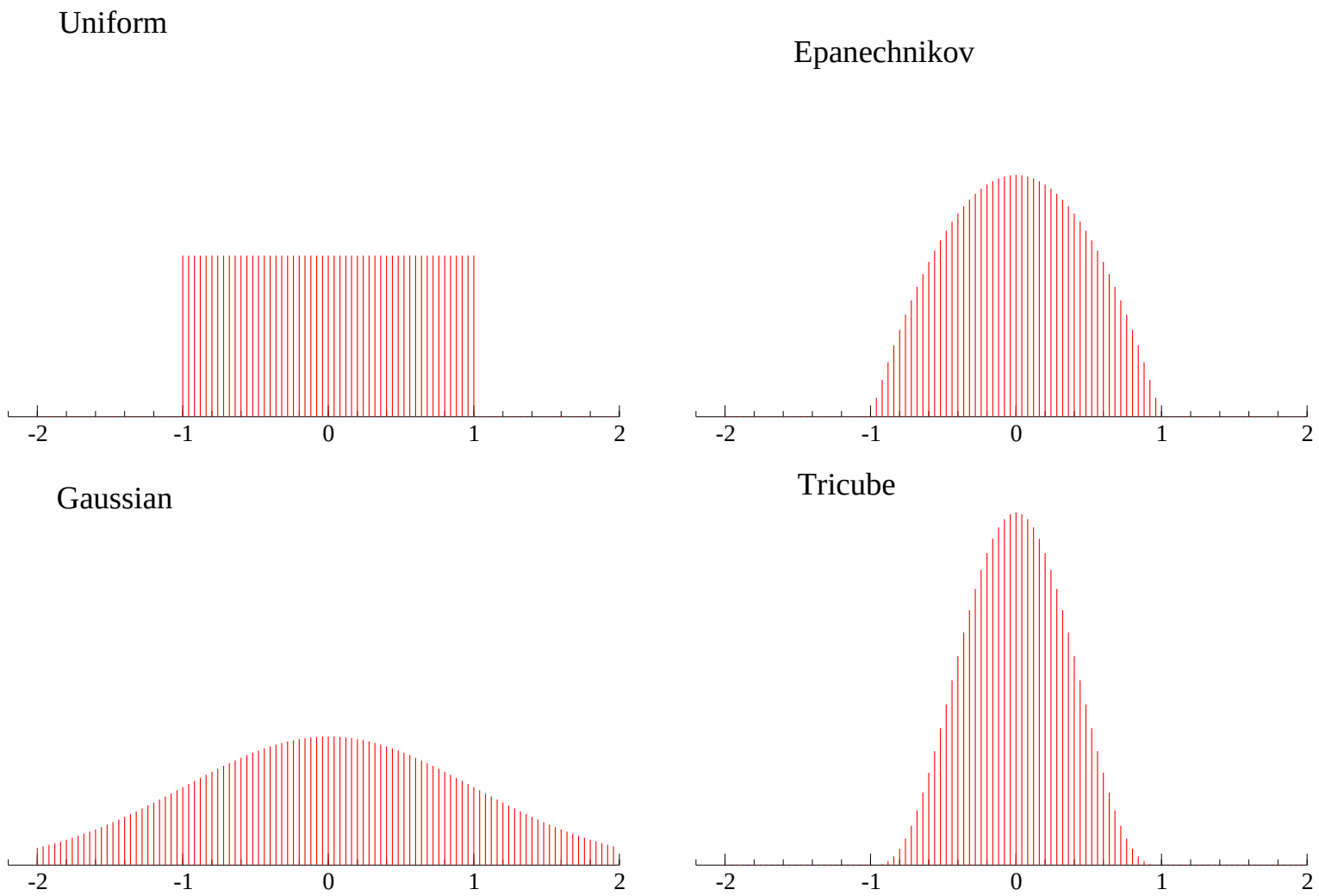
$$K(u) = \begin{cases} \frac{3}{4}(1-u^2), & \text{if } |u| \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

- The tricube kernel (used by Loess, a k-nearest-neighbour local polynomial method) is defined as follows:

$$K(u) = \begin{cases} (1-|u|^3)^3, & \text{if } |u| \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

- The Gaussian kernel is such that h is the standard deviation parameter and $K(u)$ is the standard Normal density function: $K(u) = (2\pi)^{-1} e^{-u^2/2}$.
- In the k-nearest-neighbour approach h is set equal to the distance from the k-th closest x_i to x_0. Hence, it varies with x_0.

Figure: Kernel functions.



Bias-Variance Trade-off

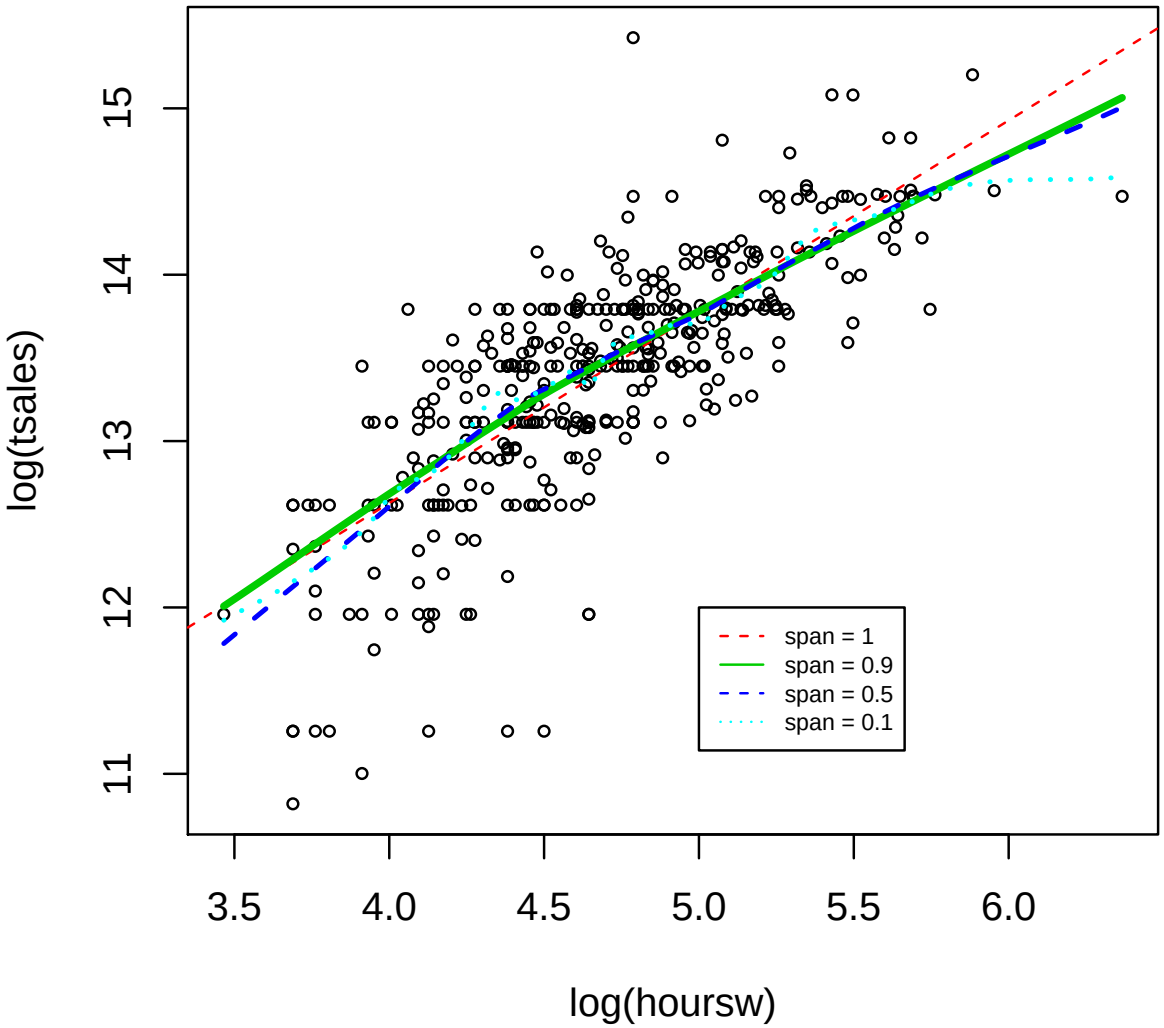
- The degree p of the polynomial and the bandwidth h are crucial quantities. They regulate the complexity of the fit, which increases with p and decreases with h.
- The mean square estimation error is

$$\text{MSE}[\hat{f}(x_0)] = \text{Bias}^2[\hat{f}(x_0)] + \text{Var}[\hat{f}(x_0)].$$

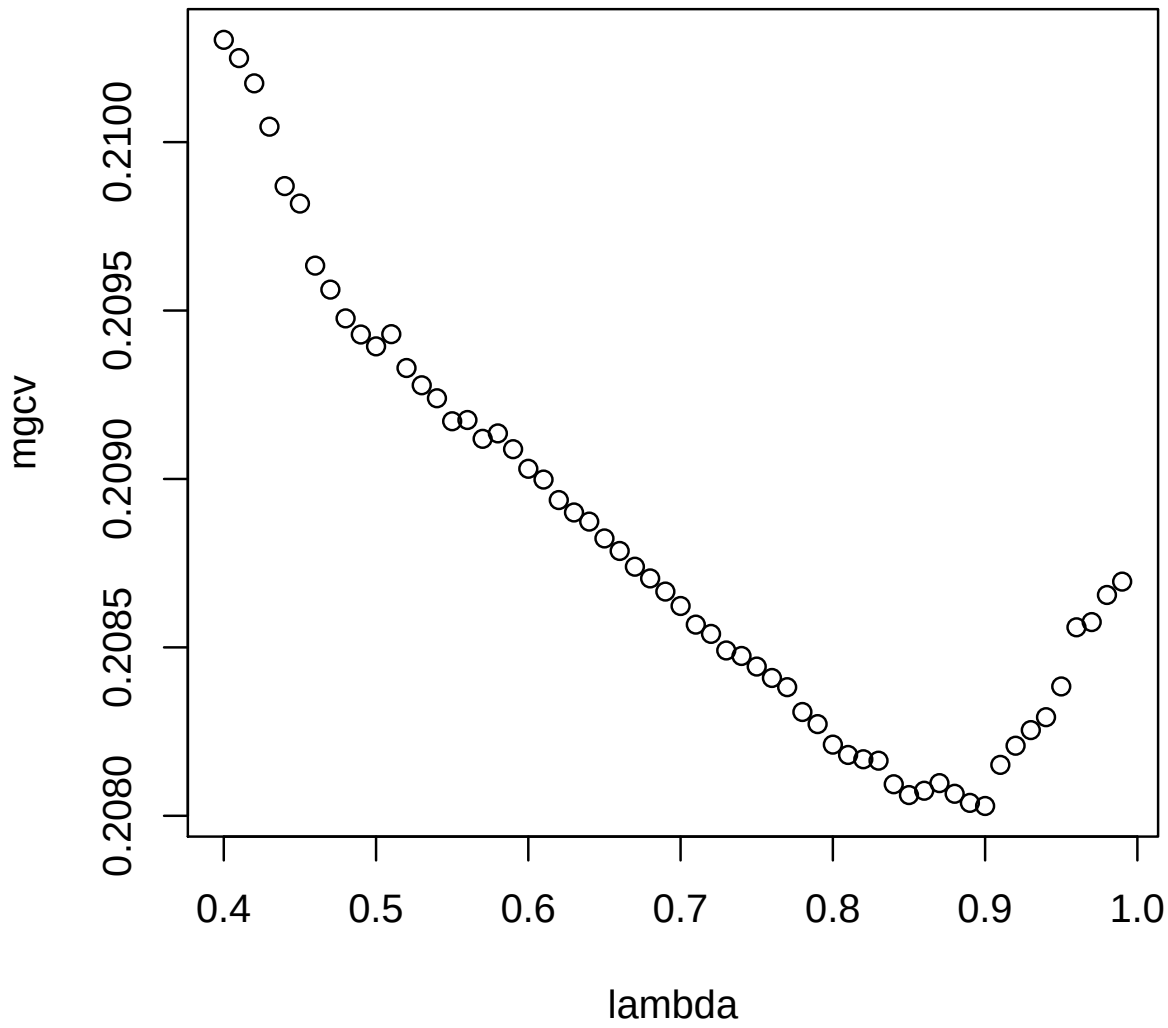
  Both p and h affect the MSE.
- The choice of the kernel has much less impact on the bias-variance trade-off.
- For a given h, by increasing p we reduce the bias (we improve the Taylor approximation by including higher order terms), but we increase the variance (there are more parameters to be estimated).
- For a given p, a large h will capture more observations and the fit tends to be the same as the global polynomial fit (with $p + 1$ df). The variance is small, but the bias is high.

- Usually, $\hat{f}(x_0)$ is biased for $f(x_0)$, unless the true regression function is a polynomial of order p.
  The bias arises from neglecting higher order terms in the Taylor expansion.
- The bias is inversely related to p and positively related to the bandwidth h.
- As far as the *variance* is concerned, for a given p, $\text{Var}[\hat{f}(x_0)]$ decreases as h increases.
- The optimal choice of (p, h) should minimise $\text{MSE}[\hat{f}(x_0)]$, which can be estimated using a variety of methods (see e.g. Fan and Gijbels, 1994).
- It is usually most effective to choose a low degree polynomial (e.g. $p = 1$, or $p = 3$) and concentrate the efforts on the selection of the bandwidth.
- The optimal h can be obtained by cross-validation.

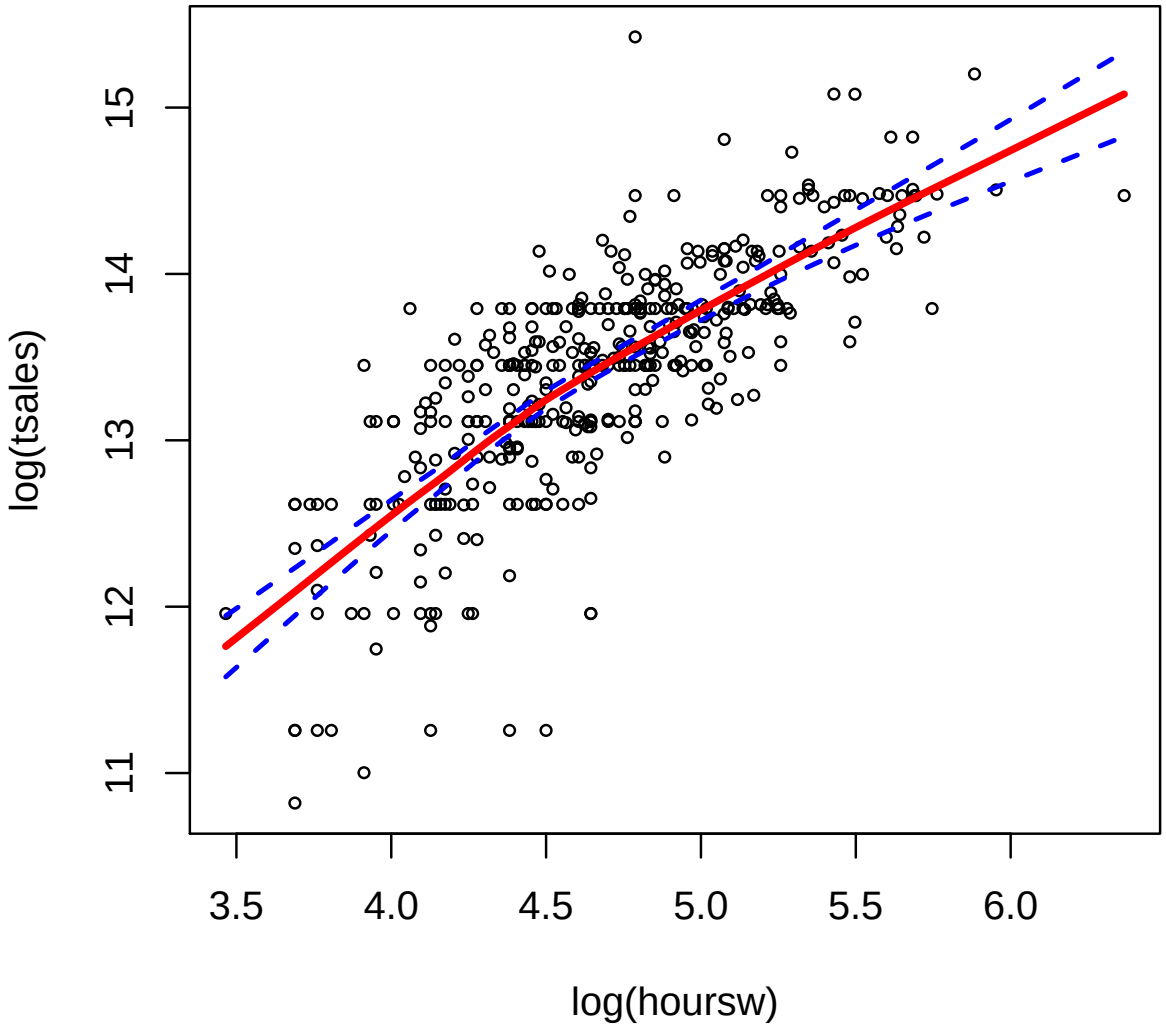
Clothing dataset: Nearest-neighbour local-polynomial regression by loess (locally estimated scatterplot smoothing, $p = 1$, tricube kernel, NN-bandwidth).



Clothing dataset: Nearest-neighbour local-polynomial regression by loess. Generalized CV criterion



Clothing dataset: Nearest-neighbour local-polynomial regression by loess



Special case: local averaging ($p = 0$) and kernel smoothing

The local constant fit ($p = 0$) is an important special case.
The local weighted average of Y,

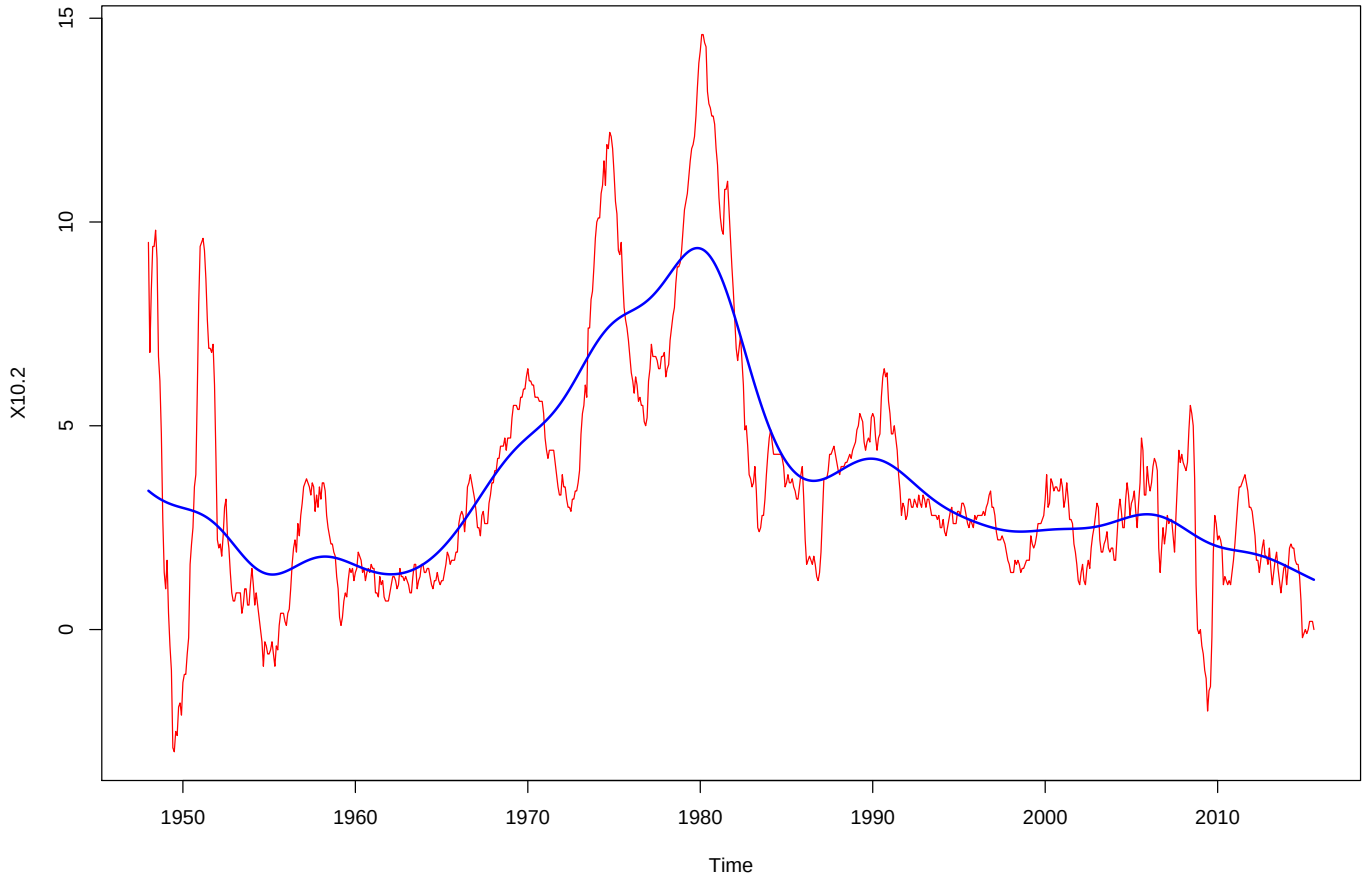
$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) y_i}{\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)}$$

is known as the Nadaraya-Watson average.
Denoting by $N_k(x)$ the set of k points nearest to x, the k-nearest-neighbour average is

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i.$$

This corresponds to the use of a uniform kernel.

NW estimate of underlying inflation (Gaussian kernel, $h = 5.4$

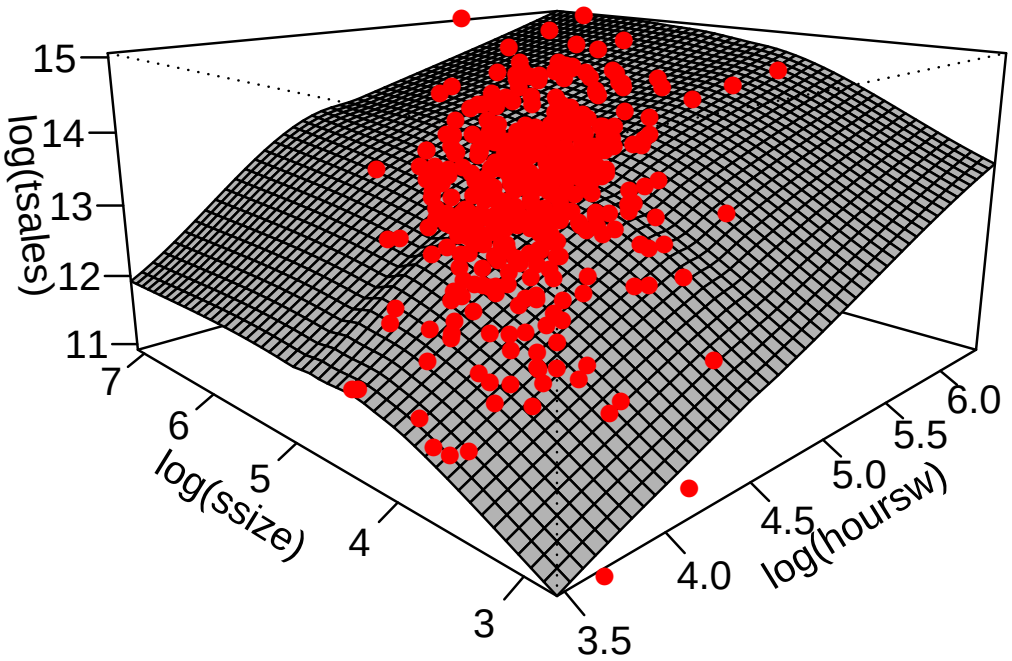


Local regression in high-dimensional spaces

- The locally weighted regression methodology extends to the vector case, $\boldsymbol{x}_0 \in \mathbb{R}^q$. The kernel is a function of the Euclidean distance of the observed $\boldsymbol{x}_i$ from $\boldsymbol{x}_0$ (or other distances, like Mahalanobis').
- However, we face the curse of dimensionality: as the dimension of the input space q increases, the number of observations N has to increase at an exponential rate with q, in order to achieve a mean square estimation error of comparable size.
- The estimation problems at the boundary get compounded.
- Visualising the fit is also difficult for $q > 2$.
- These considerations pave the way for the class of additive models and varying parameter models that will be considered later.

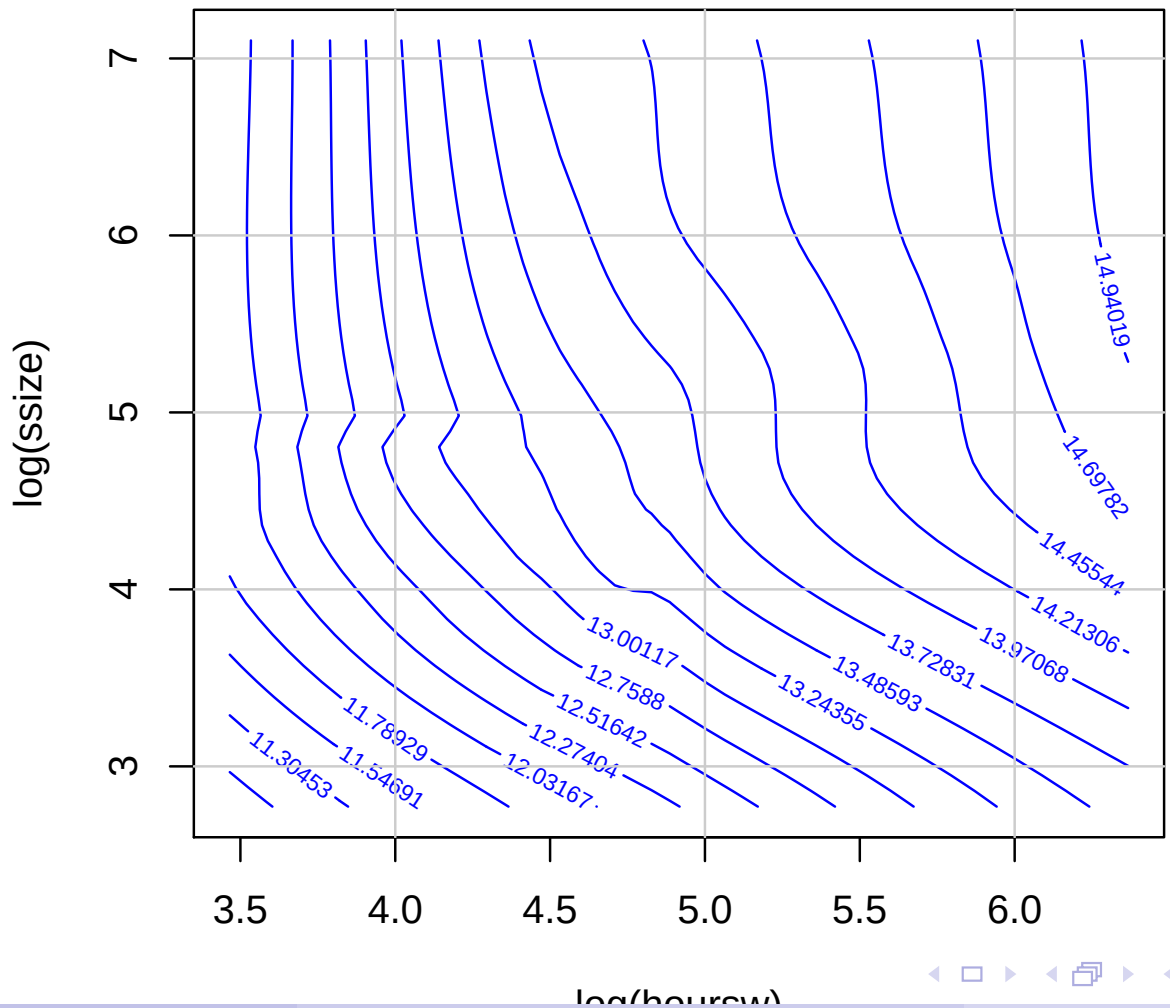
Clothing dataset: bivariate local polynomial regression of `log(tsales)` on `log(hoursw)` and `log(ssize)`

Figure



Clothing dataset: bivariate local polynomial regression of `log(tsales)` on `log(hoursw)` and `log(ssize)`, contour plot

Figure



(Nonparametric) Density Estimation

Our aim is to estimate the distribution of a continuous variable Y from a random sample $\{y_i, i = 1, \ldots, N\}$, without making a parametric assumption.
A familiar estimator is the *histogram*:

- Divide the support into non-overlapping classes: select a starting point, y_0, and divide the range of values into contiguous non-overlapping intervals of size l

  $$C_{1l} = [y_0, y_0 + l), C_{2l} = [y_0 + l, y_0 + 2l), \ldots, C_{jl} = [y_0 + (j-1)l, y_0 + jl), \ldots.$$

  The partition depends on l (which may also vary with the classes).

- Count the number N_j or frequency of cases that fall in each interval $C_{jl}, j = 1, 2, \ldots,$ and divide by the size of the interval:

$$\hat{f}_j(y) = \frac{1}{lN} \sum_{i=1}^{N} I(y_i \in C_{jl}) = \frac{N_j}{lN}.$$

- Draw rectangles with basis equal to the class size and height equal to the density of cases, which is the ratio of the frequency and the width or size of the class.
- The area of each rectangle is equal to the frequency of cases in the class, and the total area is either N or 1.

The histogram has several limitations as an estimator of the density:

1. It is a discontinuous function of y (jumps at the extremes of the intervals).
2. It is constant within each class.
3. It depends on y_0

Other properties:

1. It depends on the number and sizes of the classes.
2. The bias in estimating the true density (pdf) increases with l, but the variance decreases with l.
3. It is nonnegative and the total area is 1.

The limitations can be overridden by considering, for each value y, a symmetric interval centred at y, with size $2h$, $C_h(y) = [y - h, y + h)$ and estimating the density by

$$\hat{f}_h(y) = \frac{1}{2hN} \sum_{i=1}^{N} I(y_i \in C_h(y))$$

(the frequency of observations that belong to the interval divided by the size of the interval).

The value y_i contributes to $\hat{f}_h(y_i)$ if $y_i \in C_h(y)$, which occurs if

$$u_i = \frac{y_i - y}{h}$$

is such that $|u_i| < 1$.
Thus, $I(y_i \in C_h(y)) = I(|u_i| \leq 1)$.
Defining the uniform kernel

$$K(u) = I(|u| \leq 1),$$

we can rewrite

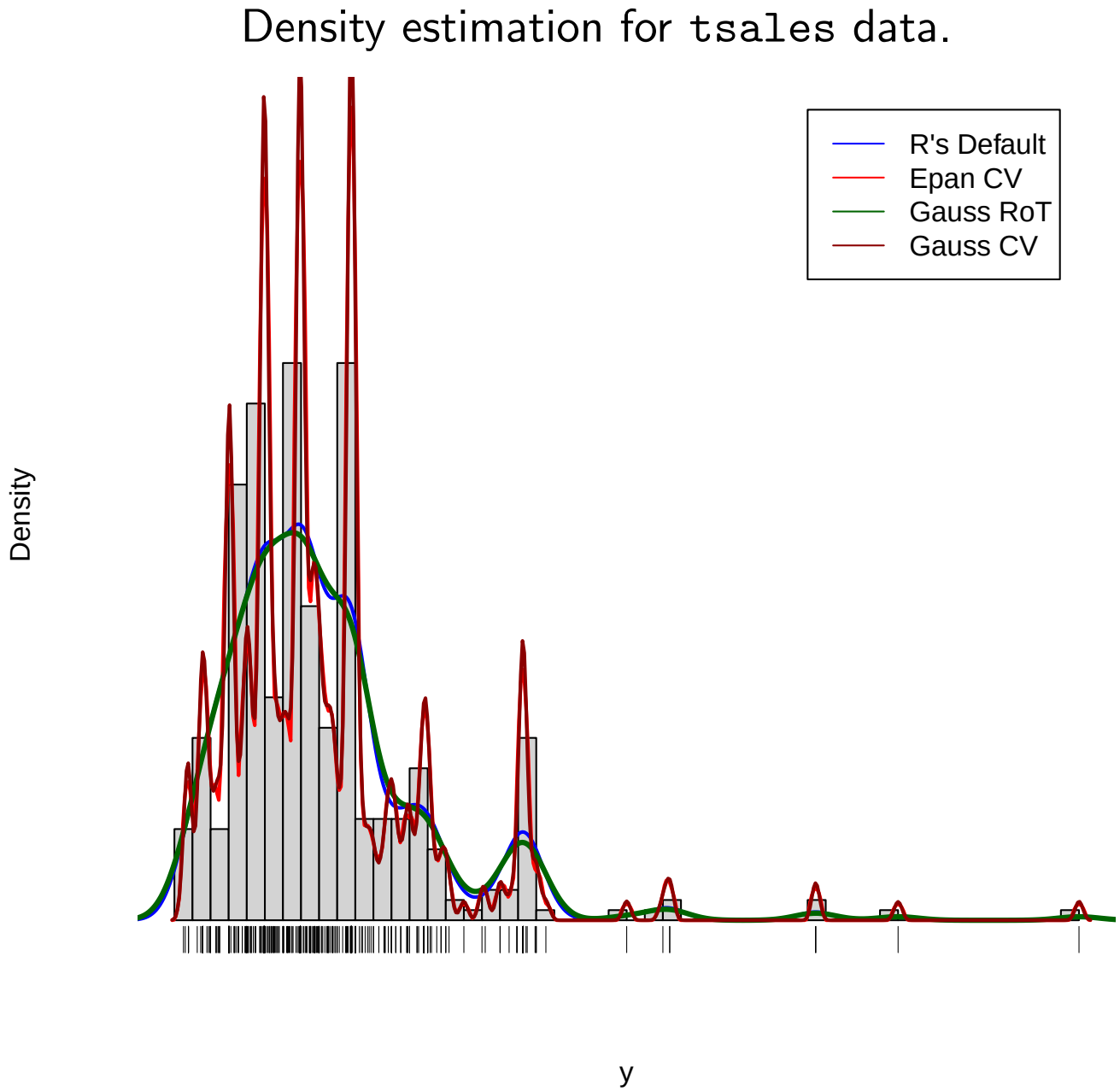
$$\hat{f}_h(y) = \frac{1}{2hN} \sum_{i=1}^{N} K\left(\frac{y_i - y}{h}\right),$$

We can replace the uniform kernel with other kernels, like the Gaussian kernel or Epanechnikov's.

- The MSE, $\mathrm{E}[\hat{f}_h(y) - f(y)]^2$ depends crucially on the bandwidth h with the usual Bias-Variance *trade-off* occurring: the bias is directly related to h; the variance is inversely related.
- The optimal value of h depends on the unknown true density and its derivatives.
- It can be estimated by crossvalidation.
- If we assume that the underlying true density is Gaussian the optimal h is approximated by the following plug-in or rule of thumb bandwidth:

$$\hat{h} = 1.06 \min \left\{ \hat{\sigma}, \frac{\hat{R}}{1.349} \right\} N^{-1/5},$$

dove $\hat{\sigma}$ is the standard deviation of the sample y_i's, and $\hat{R}$ is the interquartile range $(Q_3 - Q_1)$.


Density estimation for tsales data.
R's Default
Epan CV
Gauss RoT
Gauss CV
Density
y

Classification and Kernel Density Estimation

Suppose that the data are p-dimensional and we aim at estimating the joint pdf of $(X_1, \ldots, X_p)$ using N independently drawn samples $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})'$.
If we assume that the rv's X_j are independent, so that

$$f(x_1, \ldots, x_p) = f(x_1) f(x_2) \cdots f(x_p),$$

then we can estimate the joint pdf by multiplying the univariate pdf estimates:

$$\hat{f}(\boldsymbol{x}) = \hat{f}(x_1) \hat{f}(x_2) \cdots \hat{f}(x_p),$$

where each marginal density estimator takes the form:

$$\hat{f}(x_j) = \frac{1}{2 h_j N} \sum_{i=1}^{N} K\left(\frac{x_j - x_{ij}}{h_j} \right)$$

- The **naive Bayes classifier** uses the above estimator for the density of X in group k, $f_k(\boldsymbol{x})$.
- The posterior probability is estimated as $\hat{P}(G = k|X = x) \propto \pi_k \hat{f}_k(\boldsymbol{x})$.
- If there are only two groups, $G = \{0, 1\}$, the classifier is

$$\hat{G}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \hat{f}_1(\boldsymbol{x})/\hat{f}_0(\boldsymbol{x}) > \pi_0/\pi_1 \\ 0, & \text{otherwise} \end{cases}$$

- Each marginal univariate density is estimated separately.

Classification with Nearest Neighbours

- The k-NN (nearest neighbour) method estimates the posterior probability $P(G = g|\boldsymbol{x})$ as the fraction of points in the neighbourhood belonging to group g.
- Let $(\boldsymbol{x}_i, y_i)$ denote our training sample, with Y being a binary variable. The inputs are standardised.
- Suppose we aim at classifying a unit with input signature $\boldsymbol{x}_0$.
- We denote by $\mathcal{N}_k(\boldsymbol{x}_0)$ the neighborhood of $\boldsymbol{x}_0$, defined by k-closest points $\boldsymbol{x}_i$ in the training sample.

k-NN algorithm

1. Locate the k nearest neigbouring points (i.e. with smallest distance $||\boldsymbol{x}_i - \boldsymbol{x}_0||$)
2. Compute the weighted average of y_i for the k units ($i \in \mathcal{N}_k(\boldsymbol{x}_0)$).

$$\hat{P}(G = 1|\boldsymbol{x}_0) = \frac{\sum_i K(u_i) y_i}{\sum_i K(u_i)}.$$

- The weights are provided by a Kernel function $K(u_i)$, where $u_i = ||\boldsymbol{x}_i - \boldsymbol{x}_0||/M$ and $M = \max\{||\boldsymbol{x}_j - \boldsymbol{x}_0||\}$ for $j \in \mathcal{N}_k(\boldsymbol{x}_0)$.
- Most often the uniform kernel is used, so that $\hat{P}(G = 1|\boldsymbol{x}_0)$ is the simple average of the values y_i in the neighbourhood of $\boldsymbol{x}_0$.
- The classifier is $\hat{G}(\boldsymbol{x}) = 1$ if $\hat{P}(G = 1|\boldsymbol{x}) > 0.5$ and $\hat{G}(\boldsymbol{x}) = 0$ otherwise (classify using a majority vote).
- If there is a tie (no majority), it is broken at random.

- Main advantage is that the decision boundary is flexible.
- The crucial parameter is the 'bandwidth' k. As k increases the bias increases, but the variance reduces.
- Also, the choice of the distance measure is often critical.
- For $k = 1$ we locate the closest point and borrow its value for the prediction. The bias is low, but the variance is high.
- When the input space is highly multidimensional, the NN method is very inefficient (curse of dimensionality).