

Nonparametric Estimation Methods

Ian Crawford

Department of Economics

II. Nonparametric Regression Estimation (Part 1)

i. Motivation/derivation

ii. Properties

iii. Estimation and Inference

iv. Bandwidth selection

v. Local adaptation of the smoothing parameter

vi. Local polynomial regression

We are interested in the regression

$$y = m(x) + e$$

where the functional form of $m(x)$ is to remain unspecified. We can motivate nonparametric regression methods from two approaches

1. Smoothing
2. Definitional

Smoothing

Suppose we have paired data $(Y_1, X_1), \dots, (Y_n, X_n)$ arranged so that $X_1 \leq X_2 \leq \dots \leq X_n$.

In what follows we have $n = 100$ observations on the model

$$y = 10 + \cos(\pi 4x) - 5x + e$$

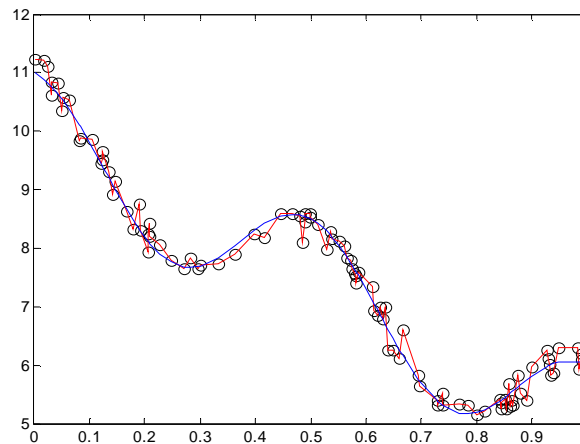
where

$$e \sim N(0, 0.09)$$

$$x \in [0, 1]$$

The simplest NP regression method for describing the relationship between (y, x) is *linear interpolation*.

But this leads to an unacceptably jumpy/discontinuous non-differentiable regression function.



A wide variety of smoothing techniques have been suggested.

Moving Average Estimators

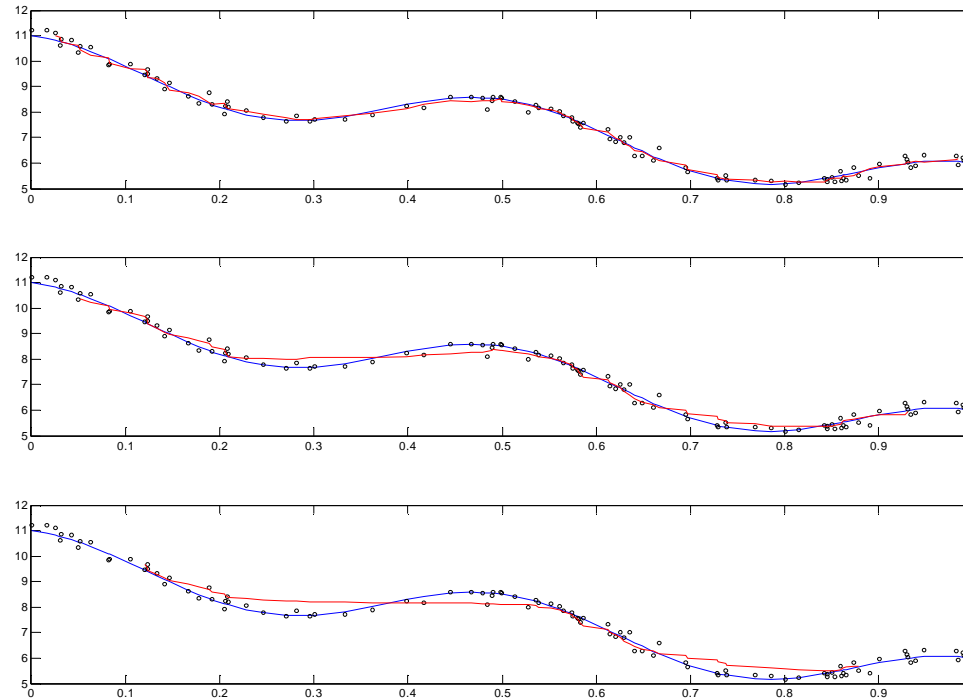
The moving average estimator of $m(x)$ is the average of h consecutive observations centered at X_i

$$\widehat{m}(X_i) = \frac{1}{h} \sum_{j=a}^b Y_j$$

where

$$a = i - \frac{h-1}{2}$$
$$b = i + \frac{h-1}{2}$$

Moving Average Estimators - Bandwidth ($h = 5, 10, 25$)



Moving Average Estimators

Given the definition of the regression function, the moving average estimator can be written as

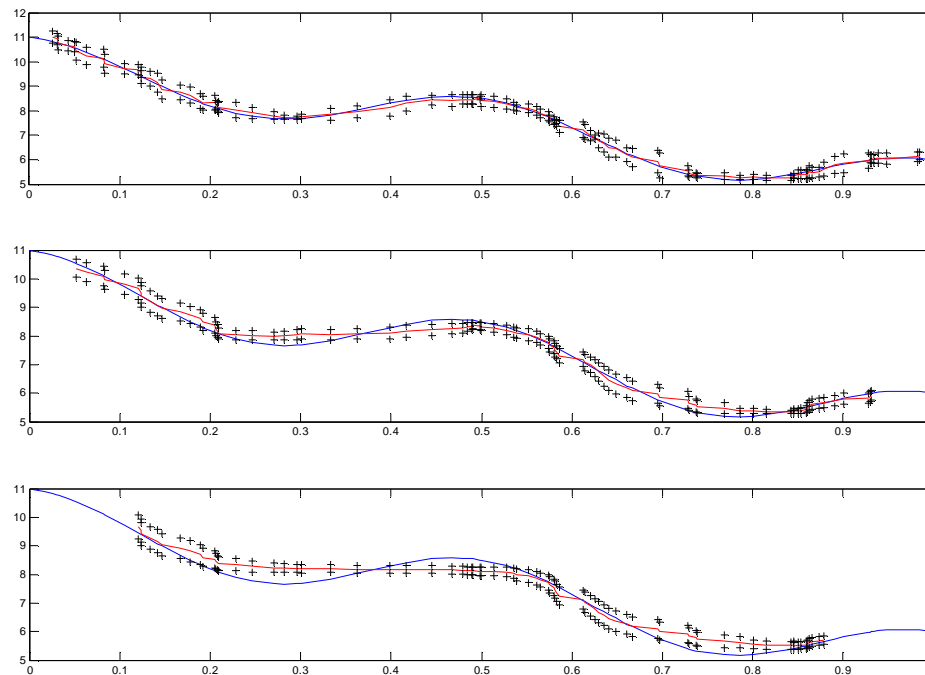
$$\widehat{m}(X_i) = \frac{1}{h} \sum_{j=a}^b m(X_i) + \frac{1}{h} \sum_{j=a}^b e_i$$

so that if the number of observations being averaged increases as the number of observations increase, then the 2nd term on the rhs will be approximately normal with mean 0 and variance σ_e^2/h . And as these observations cluster closer to X_i the first term converges to $m(X_i)$. Further

$$h^{1/2} (\widehat{m}(X_i) - m(X_i)) \sim N(0, \sigma_e^2)$$

Moving Average Estimators - Bandwidth ($h = 5, 10, 25$)

As usual the MSE can be minimised by widening the neighbourhood $[a, b]$ until the increase in bias is offset by the reduction in variance.



Moving Average Estimators

The moving average smoother is consistent.

It is asymptotically normal.

It illustrates the familiar bias/variance trade off wrt bandwidth.

It's easy to calculate.

But it's discontinuous.

A more general formulation of the local averaging smoother modifies the standard MA structure as follows.

$$\widehat{m}(x) = \sum_{i=1}^n w_i(x) Y_i$$

This is the estimate of the regression function at an arbitrary point (x) and it uses all of the data locally weighted by

$$w_i(x)$$

As one would expect that observations close to x would have conditional means close to $m(x)$ it is natural to place more weight on these observations and less on those (symmetrically) further away.

We can think of the standard MA as applying a rectangular weight function around the point of interest.

Kernel Smoothers

We are forming the regression function at the point x as a weighted sum of the y_i data, where the weights depend on the point of interest.

A conceptually simple/natural way to specify those weights is to use a unimodal function centered at x which declines in either direction at a rate which is controlled by a scale parameter.

i.e. *kernel functions*

Kernel Smoothers

Using kernel functions we can define the weights to be

$$w_i(x) = \frac{\frac{1}{nh} K\left(\frac{X_i - x}{h}\right)}{\frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right)}$$

where

1. the numerator is the familiar kernel weight function evaluated at x
2. the denominator is the kernel density estimate.

As usual the shape of the weights are determined by K and their magnitude by the bandwidth h

Kernel Regression

Using

$$\widehat{m}(x) = \sum_{i=1}^n w_i(x) Y_i$$

and

$$w_i(x) = \frac{\frac{1}{nh} K\left(\frac{X_i - x}{h}\right)}{\frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right)}$$

gives the Kernel Regression function estimator (Nadaraya (1964), Watson (1964)):

$$\widehat{m}(x) = \frac{\frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) Y_i}{\frac{1}{nh} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right)}$$

An Alternative Motivation

Given the regression model

$$y = m(x) + e$$

then

$$E(y|x) = m(x)$$

By the definition of a conditional expectation

$$E(y|x) = \int y f(y|x) dy = \int y \frac{f(y, x)}{f(x)} dy = m(x)$$

We can plug in kernel density estimates for the various terms.

An Alternative Motivation

For the bivariate density we can use the bivariate product kernel:

$$\hat{f}(x, y) = \frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right) \frac{1}{g} K\left(\frac{Y_i - y}{g}\right)$$

So

$$\begin{aligned} \int y \hat{f}(x, y) dy &= \frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right) \int \frac{y}{g} K\left(\frac{Y_i - y}{g}\right) dy \\ &= \frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right) Y_i \end{aligned}$$

Also, for the marginal we use the univariate density

$$\hat{f}(x) = \frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right)$$

An Alternative Motivation

Plugging in the estimators gives the Nadaraya-Watson kernel regression estimator once more

$$\widehat{m}(x) = \int \frac{y \widehat{f}(x, y)}{\widehat{f}(x)} dy = \frac{\frac{1}{nh} \sum K\left(\frac{X_i - x}{h}\right) Y_i}{\frac{1}{nh} \sum K\left(\frac{X_i - x}{h}\right)}$$

Here we view the problem as essentially that of estimating an unknown conditional expectation function

Of course it's easy to see the connection if you write

$$\begin{aligned} \frac{\frac{1}{nh} \sum K\left(\frac{X_i-x}{h}\right) Y_i}{\frac{1}{nh} \sum K\left(\frac{X_i-x}{h}\right)} &= \sum \left(\frac{\frac{1}{nh} K\left(\frac{X_i-x}{h}\right)}{\frac{1}{nh} \sum K\left(\frac{X_i-x}{h}\right)} \right) Y_i \\ &= \sum w_i(x) Y_i \end{aligned}$$

you can see that the Nadaraya-Watson kernel regression estimator is a weighted (local) average of the response variable again.

The Nadaraya-Watson kernel regression estimator shares this with other smoothing techniques such as spline-smoothing .

Kernel Regression - remarks

If the denominator of $w_i(x)$ is zero so is the numerator and the estimator is not defined - this happens (quite rightly) when the data is sparse/absent i.e. $\hat{f}(x) = 0$.

Bandwidth determines the degree of smoothness

Choosing the bandwidth to trade off over- and under-smoothness is the key problem.

If the Kernel estimator is evaluated at $\{X_i\}_{i=1,\dots,n}$ then as $h \rightarrow 0$

$$\widehat{m}(X_i) \rightarrow Y_i$$

and as $h \rightarrow \infty$

$$\widehat{m}(X_i) \rightarrow \frac{1}{n} \sum Y_i$$

So small bandwidth gives interpolation, and wide bandwidths gives you an over-smoothed curve.

Inference

Confidence intervals for the Nadarya-Watson estimator can be constructed based upon

$$h^{1/2}n^{1/2} \left(\widehat{m}(x) - m(x) - \frac{1}{2}d_K h^2 \left(m''(x) - 2m'(x) \frac{f'(x)}{f(x)} \right) \right) \rightarrow N \left[0, \frac{c_K \sigma^2}{f(x)} \right]$$

(Wand and Jones (1995, p. 176) give the values for d_K, c_K for different Kernels.

But, as usual, these depend on lots of things we don't know.

And as usual we can use asymptotic approximations to remove those terms

Inference

If we do this the *pointwise* standard error of the regression is defined as

$$s_{\widehat{m}(x)} = \left[\frac{c_K \widehat{\sigma}^2}{nh \widehat{f}(x)} \right]^{1/2}$$

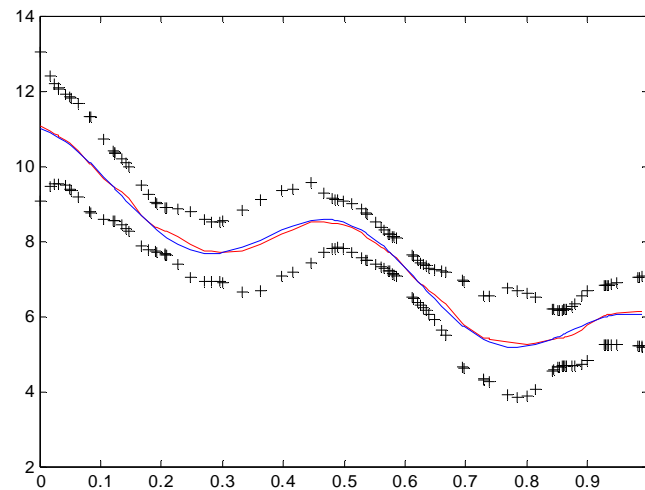
where c_K is a constant which depends on the kernel (Gaussian $c_K = 1/2\pi^{1/2}$, see Yatchew (2003) or Wand and Jones (1995, p. 176) for the values for other kernels),

$$\widehat{\sigma}^2 = 1/n \sum_{i=1}^n (y_i - \widehat{m}(x))^2$$

The 95% pointwise confidence interval is then

$$\widehat{m}(x) \pm 1.96 s_{\widehat{m}(x)}$$

Kernel Regression and 95% asymptotic confidence interval



Note that the confidence interval is not uniform, it's pointwise and depends on the (local) density of the data.

Implementation - Gaussian kernel regression estimator

$$\widehat{m}(x) = \frac{\frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right) Y_i}{\frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right)} \quad \widehat{m}(x) \pm 1.96s \left[\frac{c_K \widehat{\sigma}^2}{nh \widehat{f}(x)} \right]^{1/2}$$

```
x = X; h = 0.02;
mx = NaN*ones(J,1); ci = NaN*ones(J,1);
for j=1:n
K = (1/sqrt(2.*pi).*exp(-(((X(j)-x)/h).^2)./2));
mx(j) = (mean((1/h)*K.*Y))./(mean((1/h)*K));
s2 = mean((Y-mx(j)).^2);
ci(j) = 1.96*sqrt((0.282094792*s2)/(n*h*mean((1/h)*K)));;
end;
```

Kernel Regression - Properties

The kernel regression estimate is consistent (see Hardle (1990, p 39)). But the precise speed of convergence is (very) hard to derive (Gaser & Muller (1984) have it for the fixed design case).

However approximately as $h \rightarrow 0$ and $nh \rightarrow \infty$

$$MSE \approx h^4 d_K^2 (m''(x))^2 / 4 + \frac{1}{nh} \sigma^2 c_K$$

where $d_K = \int u^2 K(u) du$, $c_K = \int K^2(u) du$ are constants and

$$\begin{aligned} bias^2 &\approx h^4 d_K^2 (m''(x))^2 / 4 \\ var &\approx \frac{1}{nh} \sigma^2 c_K \end{aligned}$$

so increasing bandwidth increases bias but lowers variance

Minimising this with respect to h gives

$$h \sim n^{-1/5}$$

which is slower than the rate for OLS in linear regression and the same as that for density estimators.

This is unsurprising perhaps (kernel regressions are ratios of densities)

Bandwidth selection

The same issues arise with bandwidth selection in regression models as arose with bandwidth selection in density estimation.

Deriving the MSE/MISE minimising bandwidth requires we know features of the true regression function.

$$MISE(h) = E \int [\widehat{m}(x, h) - m(x, h)]^2 dx$$

Of course we do not observe the truth so we cannot minimise this directly.

Incidentally, minimising the residual variance won't work - why?

Bandwidth selection

... because

$$\hat{\sigma}^2(h) = \frac{1}{n} \sum [Y_i - \hat{m}(X_i, h)]^2$$

will be minimised by interpolation. But cross-validation will work

$$CV(h) = \frac{1}{n} \sum [Y_i - \hat{m}_{-i}(X_i, h)]^2$$

which uses the data to “predict itself”.

Bandwidth selection

Cross-validation has some very handy properties (as we have seen). In particular (Hardle and Marron (1985)) if we select the bandwidth (h^{CV}) which minimises

$$CV(h) = \frac{1}{n} \sum [Y_i - \widehat{m}_{-i}(X_i, h)]^2$$

then asymptotically

$$MISE(h^{CV}) / MISE(h^*) = 1$$

where h^* is the optimal MISE-minimising value. So if we know h^{CV} that's good enough in large samples.

The bottom line - there are no good “rules of thumb” for bandwidth so CV is the best practical method.

Local Adaptation of the Smoothing Parameter

So far we have considered the basic idea of kernel smoothing from two perspectives.

We've looked at bandwidth selection - cross-validation

Asymptotic inference.

As with density estimation, globally optimal bandwidth selection is difficult with iid data, so adaptive kernel estimation is a reasonable response.

This proceeds in an entirely analogous way to density estimation.

Local Adaptation of the Smoothing Parameter

Step 1. Find a *pilot estimate* $\tilde{f}(x)$ which satisfies $\tilde{f}(X_i) > 0$ for all i

Define the *local bandwidth factors* λ_i by

$$\lambda_i = \left(\frac{\tilde{f}(X_i)}{\prod_{i=1}^n \tilde{f}(X_i)^{1/n}} \right)^{-\alpha}$$

where $\alpha \in [0, 1]$ is a sensitivity parameter.

Step 2. Define the adaptive kernel regression

$$\widehat{m}(x) = \frac{\frac{1}{n} \sum \frac{1}{h\lambda_i} K\left(\frac{X_i - x}{h\lambda_i}\right) Y_i}{\frac{1}{n} \sum \frac{1}{h\lambda_i} K\left(\frac{X_i - x}{h\lambda_i}\right)}$$

Local Adaptation of the Smoothing Parameter

The local factors scale the bandwidth inversely according to how dense the data are.

The sensitivity parameter controls this. Having $\alpha = 0$ simply returns the standard estimator as this will set $\lambda_i = 1$.

The first step requires a pilot kernel. As with density estimation, the consensus is that the method is insensitive to the fine detail of the pilot estimate.

A time-consuming approach like cross-validation is not worthwhile.

Kernel Methods and Local Regression

The Nadaraya-Watson estimator can be seen as a special case of a larger class of kernel regression estimators.

Take the true regression function and do a Taylor series expansion for t in the neighbourhood of x

$$m(t) \approx m(x) + m'(x)(t-x) + \dots + \frac{m^{(p)}(x)}{p!}(t-x)^p$$

This suggests a local polynomial in the neighbourhood of x where we include kernel weights into the minimisation problem

$$\min_{\beta_0, \dots, \beta_p} \sum \left[Y_i - \beta_0 - \beta_1 (X_i - x) - \dots - \beta_p (X_i - x)^p \right]^2 \frac{1}{h} K \left(\frac{X_i - x}{h} \right)$$

The result is a locally weighted least squares estimator with weights given by the kernel.

$$\begin{aligned}
\mathbf{X} &= \begin{bmatrix} 1 & X_1 - x & \cdots & (X_1 - x)^p \\ 1 & X_2 - x & \cdots & (X_2 - x)^p \\ \vdots & \vdots & & \vdots \\ 1 & X_n - x & & (X_n - x)^p \end{bmatrix} \\
W &= \begin{bmatrix} \frac{1}{h}K \left(\frac{X_1 - x}{h} \right) & 0 & \cdots & 0 \\ 0 & \frac{1}{h}K \left(\frac{X_2 - x}{h} \right) & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & & & \frac{1}{h}K \left(\frac{X_n - x}{h} \right) \end{bmatrix} \\
\mathbf{Y} &= \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}
\end{aligned}$$

Then the $\hat{\beta}(x)$ which minimises this is (of course)

$$\hat{\beta}(x) = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1} \mathbf{X}'\mathbf{W}\mathbf{Y}$$

Note that in contrast to OLS the parameters depend on x - they are *local*.

The local polynomial estimation of the regression function is then

$$\hat{m}_p(x) = \beta_0$$

Obviously (perhaps) if $\beta = \beta_0$ this reduces to a local constant which is the Nadaraya-Watson estimator

$$\hat{m}_0(x) = \frac{\frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right) Y_i}{\frac{1}{n} \sum \frac{1}{h} K\left(\frac{X_i - x}{h}\right)}$$

Bottom line - this makes the ends of the regression more sensitive to boundaries - why?

Local Polynomials

For estimating local polynomials the order p is usually taken to be one (local linear) or or three (local cubic regression).

The local linear fit performs (asymptotically) better than the Nadaraya-Watson estimator (local constant). This holds generally.

As the Nadaraya-Watson estimator, the local polynomial estimator is a weighted (local) average of the response variables.

As for all other kernel methods the bandwidth determines the degree of smoothness.

An infinitely large h makes all weights equal, thus we obtain a parametric p th order polynomial fit in that case.