Econometrics I

Lecture 10: Nonparametric Estimation with Kernels

Paul T. Scott
NYU Stern

Fall 2018

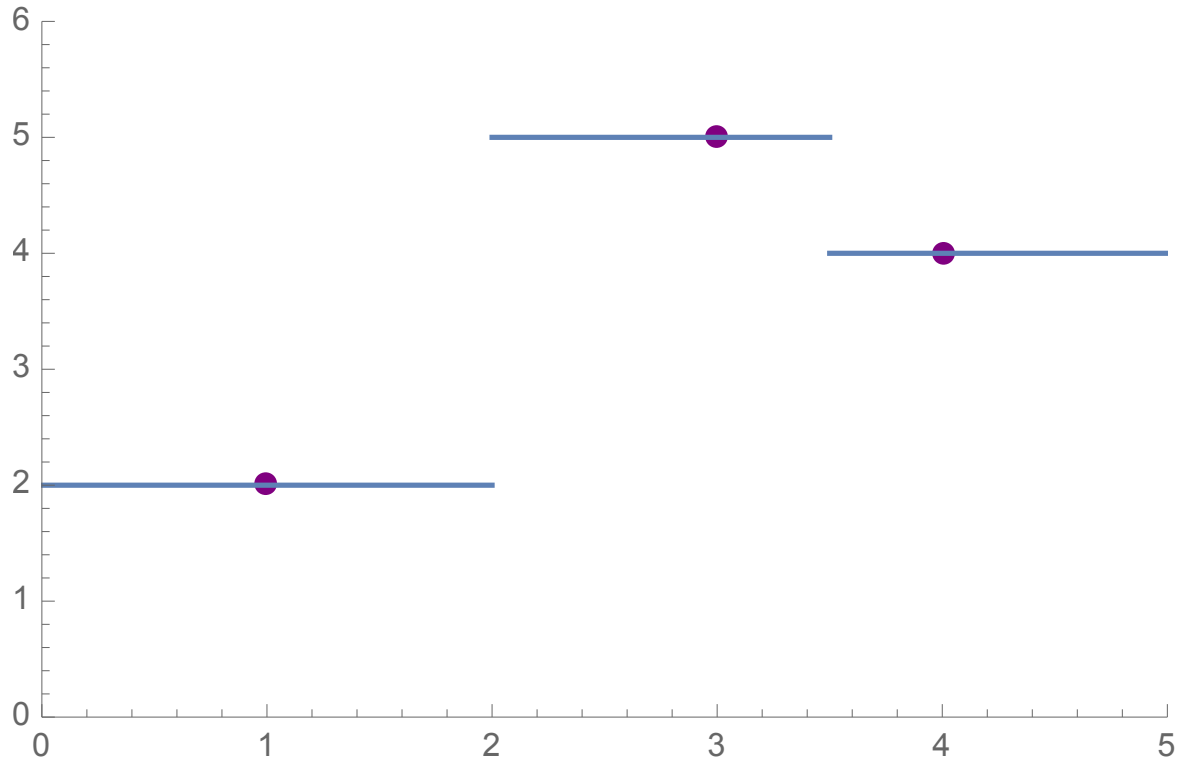
Nonparametric Regression: Intuition

- Let's get back to conditional means and consider a general functional form:

$$y = \mu(x) + \varepsilon$$

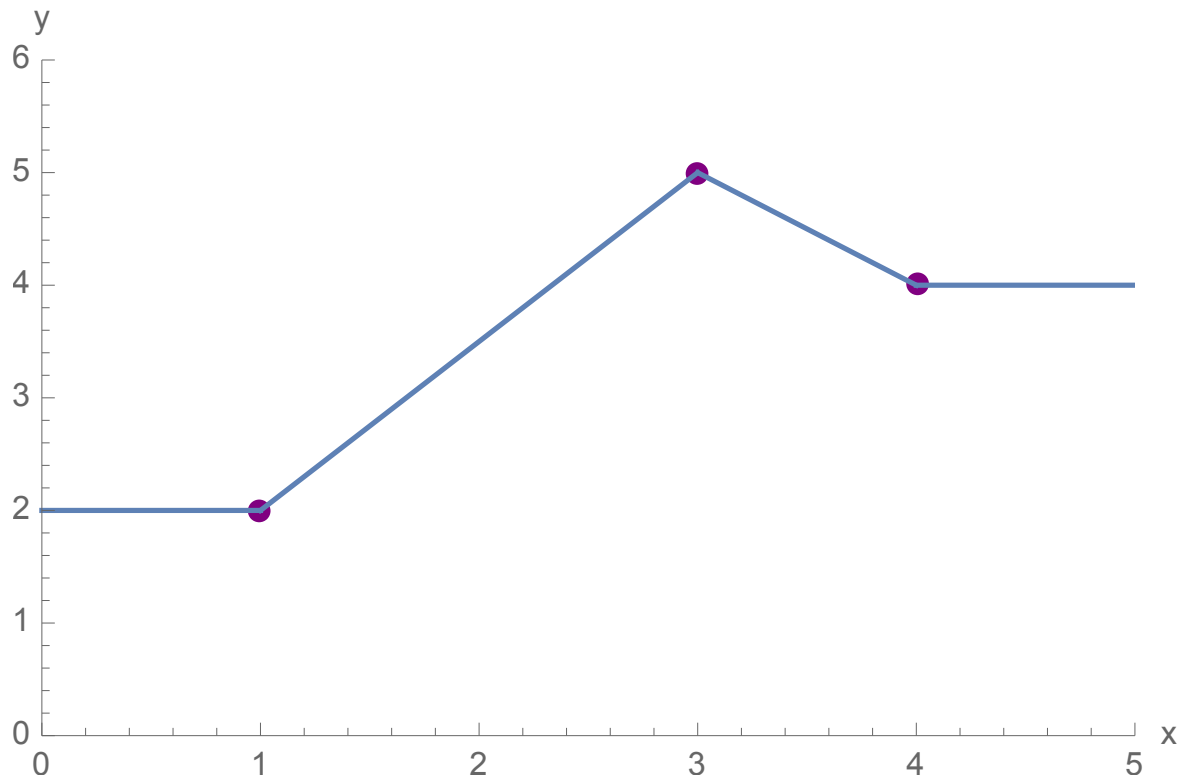
- The basic idea behind non-parametric estimation is to avoid imposing any functional form on the relationships between variables.
- How do we go about this in practice? On board: what if x is discretely distributed?

Nearest Neighbor Interpolation



- **Nearest neighbor interpolation**: $\mu(x)$ is the the value of y associated with the nearest observed value of x

Linear Interpolation



- **Linear interpolation**: $\mu(x)$ is a weighted average of two points
- There are more sophisticated interpolation techniques, especially when it comes to multidimensional x. See kriging.

Local Averaging: Intuition

$$y = \mu(x) + \varepsilon$$

- The above interpolation techniques were just about "filling in" the value of the function between observations.
- When we have lots of data, interpolation makes less sense.
  - When observations are close together, the difference between them might have more to do with noise in ε than an actual change in $\mu(x)$
  - $\mu(x)$ is likely to start looking very jagged if we interpolate between lots of close-together observations.
- This brings us to nonparametric estimators that still fit $\mu(x)$ based on the nearby observations, but average over observations to some extent.

Local Averaging

- Our estimate of $\mu(\cdot)$ can be thought of as weighted mean functions:

$$\hat{\mu}(x^*) = \sum_i^n w_i(x^*|\mathbf{x})\, y_i$$

  with $\sum_i^n w_i(x^*|\mathbf{x}) = 1$.

- Note that this is not just a weighted average, but a function that takes a different weighted average at different points. The weights depend on at what value of x^* we are evaluating the function.
- What's the w function for the linear interpolation example above?

Kernel

- Kernel estimation starts with a **kernel function**

$$K\left(x^{*}|x_{i},h\right)$$

that allows you to generate the weights from the data. Note: the kernel just depends on the point in question x^{*} and a single observation x_{i}, but the weights ultimately depend on all the observations.

- Kernels involve a bandwidth h that, rougly speaking, determines how close x_{i} should be to x^{*} for x_{i} to get some weight in $w_{i}\left(x^{*}|\mathbf{x}\right)$. The bandwidth can be adjusted (and optimally selected).

Logistic Kernel

- The **logistic kernel function is** is

$$K\left(x^{*}|x_{i},h\right)=\Lambda\left(v_{i}\right)\left(1-\Lambda\left(v_{i}\right)\right)$$

where

$$v_{i}=\frac{x_{i}-x^{*}}{h}$$

and

$$\Lambda\left(v\right)=\frac{\exp\left(v\right)}{1+\exp\left(v\right)}$$

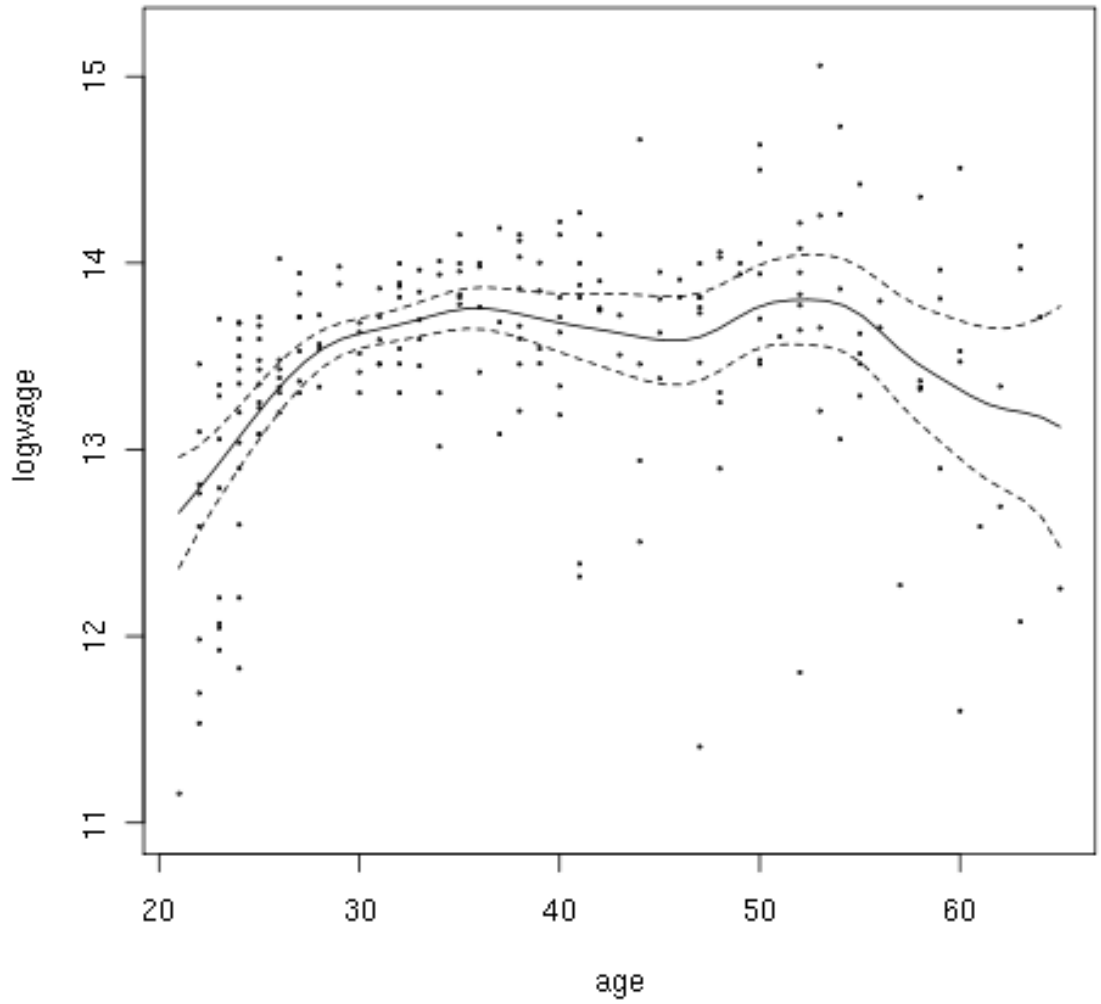
- Note that as x_i becomes far from x^*, either $\Lambda\left(v_{i}\right)\rightarrow 0$ or $\left(1-\Lambda\left(v_{i}\right)\right)\left(v_{i}\right)\rightarrow 0$, but either way the kernel goes to zero as the observations become far apart.
- Other kernels have $K\left(x^{*}|x_{i},h\right)=0$ when $|x_{i}-x^{*}|>h/2$.

From Kernel to Weights

$$\hat{\mu}\left(x^{*}\right)=\sum_{i}^{n} w_{i}\left(x^{*}|\mathbf{x}\right) y_{i}$$

- Given a kernel function $K\left(x^{*}|x_{i}, h\right)$, the weights are

$$w_{i}\left(x^{*}|\mathbf{x}, h\right)=\frac{K\left(x^{*}|x_{i}, h\right)}{\sum_{i=1}^{n} K\left(x^{*}|x_{i}, h\right)}$$



Source: Wikipedia

Bandwidth Selection

- The bandwidth h needs to be set somehow, and it's important.
- If the bandwidth is too large, we risk estimating $\mu(x^*)$ based on using observations x_i where $\mu(x^*) \neq \mu(x_i)$. **BIAS**
- If the bandwidth is too small, we risk averaging over very few observations in which case our estimate of $\hat{\mu}(x^*)$ will be very imprecise. **VARIANCE**
- There is a large literature on bandwidth selection. The main idea is to minimize the expected gap between the fitted function and true function (mean square error). This takes some work to do formally, but in some cases there are straightforward formulas for the optimal bandwidth.
- In practice, the "eyeball test" can provide a good starting point. When plotting your data and $\hat{\mu}(x)$, does seem to wiggle around to try to fit individual observations? Are there some patterns in the data that $\hat{\mu}(x)$ fails to capture because it smooths them out?

Further Comments

- Kernels become difficult-to-impossible to implement with high-dimensional data. Other nonparametric techniques are better suited (splines, LASSO).
- Any version of non-parametric estimation should do something to balance bias and variance. The broader issue is model selection, which is important for parametric as well as non-parametric estimation.