Heterogeneous Treatment Effects II

Ye Wang
University of North Carolina at Chapel Hill

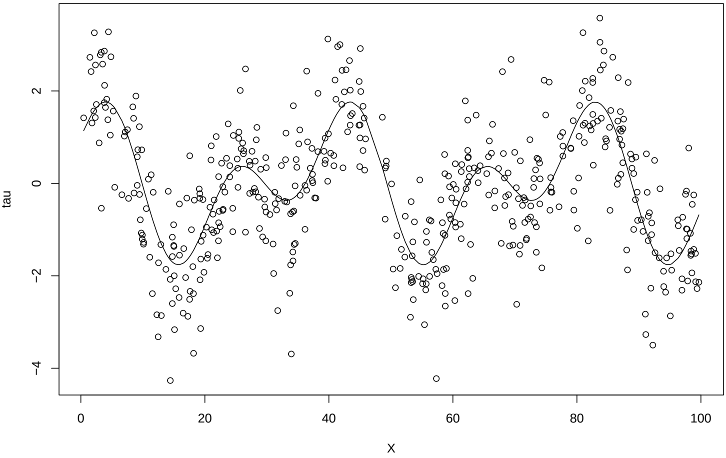
Linear Methods in Causal Inference
POLI784

Review

- We can examine the heterogeneity in treatment effects by estimating the CATE: $\tau(\mathbf{x}) = E[\tau_i | \mathbf{X} = \mathbf{x}]$.
- With these estimates, we can design assignment mechanisms that maximize social welfare or generalize our results to other contexts.
- When $\mathbf{X}$ take a few discrete values, the CATE can be estimated by conditioning on units with the same covariates values.
- It is equivalent to fitting a saturated interactive regression model.
- Relying on the interactive regression model leads to biases if the CATE is not linear in $\mathbf{X}$.
- One solution is to use the binscatter estimator.

From bins to kernels

- Again, let's first assume that τ_i is known.

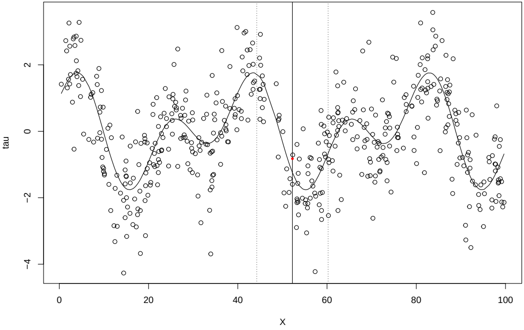


Problems with the binscatter estimator

- The binscatter estimator requires researchers to specify the bins.
- We usually assume that the bins have the same width (known as the bandwidth) and are equidistantly distributed over the support of X.
- Detecting the optimal partition of X is computationally challenging.
- $\hat{\tau}$ is the same for units in the same bin.
- This may not be very accurate if the variation is large within some bins.

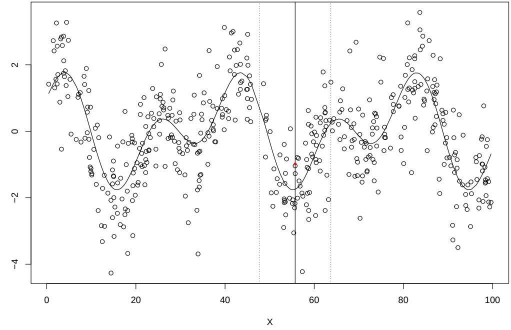
The kernel estimator

- Naturally, we can create a bin around each x and estimate $\tau(x)$ with the average of τ_i in this bin.
- We randomly pick a bandwidth of 8.



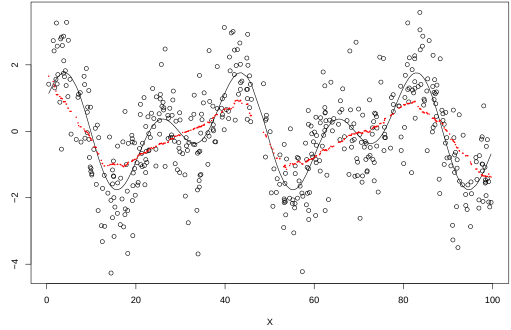
The kernel estimator

- Naturally, we can create bins around each x and estimate $\tau(x)$ with the average of τ_i in this bin.
- We randomly pick a bandwidth of 8.



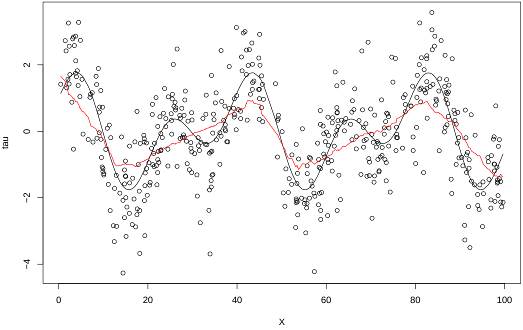
The kernel estimator

- Naturally, we can create bins around each x and estimate $\tau(x)$ with the average of τ_i in this bin.
- We randomly pick a bandwidth of 8.



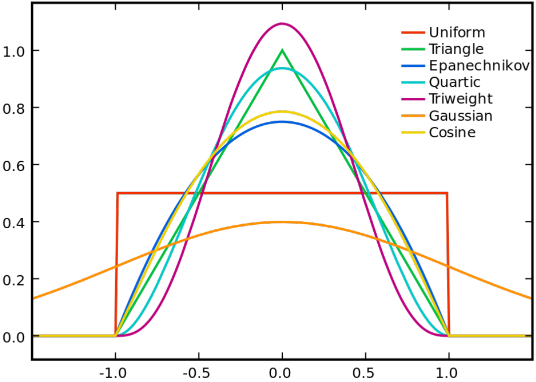
The kernel estimator

- Naturally, we can create bins around each x and estimate $\tau(x)$ with the average of τ_i in this bin.
- We randomly pick a bandwidth of 8.



The kernel estimator

- This is known as the kernel estimator with the uniform kernel.
- Points closer to x provide more information about $\tau(x)$ hence might be up-weighted.
- It leads to other choices of the kernel:

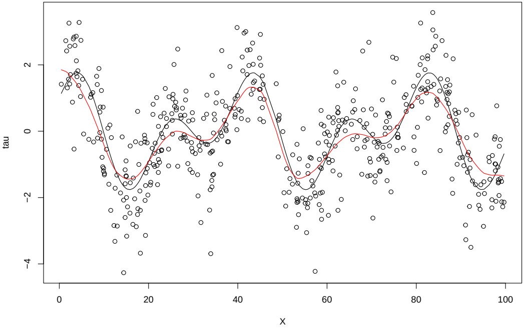


The kernel estimator

- We usually denote the kernel function as $K\left(\frac{|X_i - x|}{h}\right)$.
- Its value at X_i is determined by x and the bandwidth h.
- For the uniform kernel with a bandwidth of 8, $K\left(\frac{|X_i - x|}{h}\right) = \mathbf{1}\left\{\frac{|X_i - x|}{8} \leq 1\right\}$.
- For the triangular kernel with a bandwidth of 8, $K\left(\frac{|X_i - x|}{h}\right) = \mathbf{1}\left\{\frac{|X_i - x|}{8} \leq 1\right\}\left[1 - \frac{|X_i - x|}{8}\right]$.
- Note that the kernel function's value is always between 0 and 1.
- Its integral over the support of X equals h.
- Hence, we can see the kernel as weights for the units.
- Different kernels weigh the units differently.
- For the triangular kernel with a bandwidth of 8, units with $X_i = x$ has a weight of 1, while those with $X_i = x + 8$ has a weight of 0.

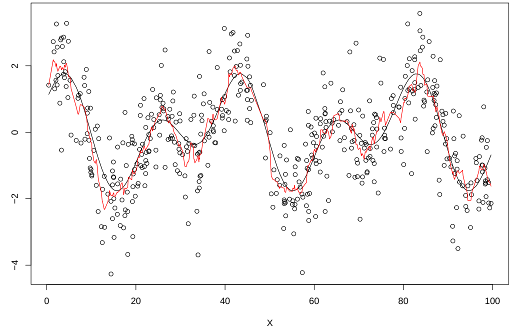
The kernel estimator

- In large sample, the choice of the kernel should not matter.



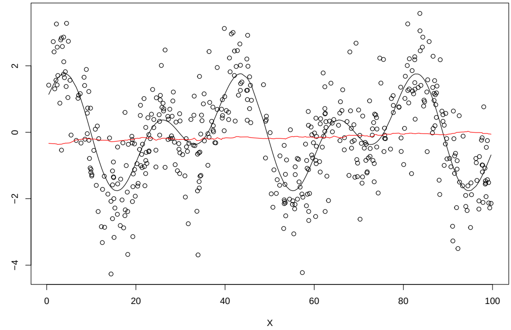
The kernel estimator

- But the bandwidth is crucial.
- A small bandwidth leads to a smaller bias but a larger variance (overfitting).



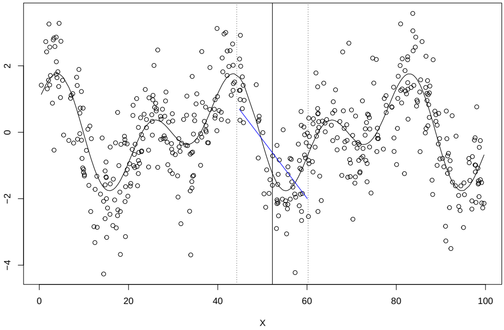
The kernel estimator

- But the bandwidth is crucial.
- A larger bandwidth leads to a smaller variance but a larger bias (underfitting).



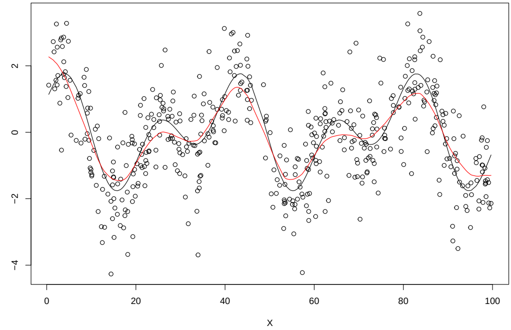
Kernel regression

- We can make the estimation more precise by replacing the average in each bin with the regression prediction (the “local regression”).



Kernel regression

- We can make the estimation more precise by replacing the average in each bin with the regression prediction (the “local regression”).



Kernel regression

- We are still running regression, with

$$\mathbf{X} = \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_N \end{pmatrix}.$$

- The difference is that we are weighting each unit i with the kernel $K\left(\frac{|X_i - x|}{h}\right)$.
- Let's denote the matrix of kernel weights as

$$\mathbf{W} = \begin{pmatrix} K\left(\frac{|X_1-x|}{h}\right) & 0 & \dots & 0 \\ 0 & K\left(\frac{|X_2-x|}{h}\right) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & K\left(\frac{|X_N-x|}{h}\right) \end{pmatrix}.$$

Kernel regression

- Now, the minimization problem becomes

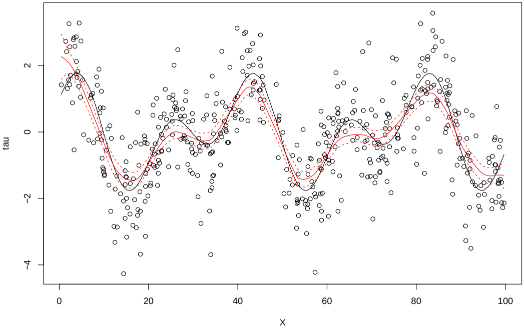
$$\hat{\beta}_x = \arg\min_{\beta} \sum_{i=1}^{N} K\left(\frac{|X_i - x|}{h}\right)(Y_i - \mathbf{X}_i'\beta_x)^2.$$

- We can show that the solution will be

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

- Therefore, the kernel regression estimator is essentially a weighted least squares (WLS) estimator.
- But $\hat{\beta}_x$ represents estimated coefficients for the local regression rather than those for the global regression.
- We predict $\tau(x)$ with $\hat{\tau} = (1, x)\hat{\beta}_x$.
- The variance of $\hat{\beta}_x$ takes the familiar sandwich form.

Kernel regression



Kernel regression

- We can make the model more complicated by setting

$$\mathbf{X} = \begin{pmatrix} 1 & X_1 & X_1^2 & \cdots & X_1^K \\ 1 & X_2 & X_2^2 & \cdots & X_2^K \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_N & X_N^2 & \cdots & X_N^K \end{pmatrix}.$$

- The WLS estimator has the same form but the approximation will be more precise.
- We refer to it as the local polynomial regression.
- Kernel regression can be extended to the multivariate case with the weight $K\left(\frac{|X_{1i}-x_1|}{h_1}\right)K\left(\frac{|X_{2i}-x_2|}{h_2}\right)\cdots K\left(\frac{|X_{Pi}-x_1|}{h_P}\right)$.
- But selecting the optimal bandwidth will be an impossible mission (curse of dimensionality).
- Machine learning is more effective in this case.

Kernel regression for estimating the CATE

- We have been assuming that τ_i is known.
- When it is not, we can fit a local regression with

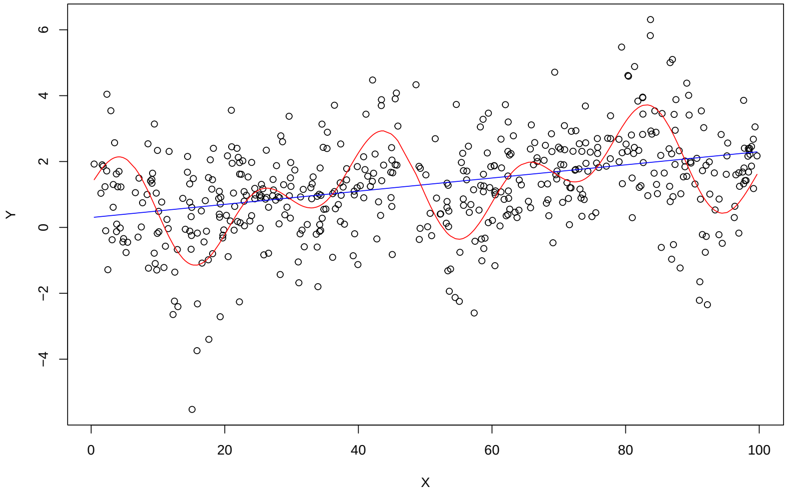
$$\mathbf{X} = \begin{pmatrix} 1 & D_1 & X_1 - x & D_1 * (X_1 - x) \\ 1 & D_2 & X_2 - x & D_2 * (X_2 - x) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & D_N & X_N - x & D_N * (X_N - x) \end{pmatrix}.$$

- The minimization problem becomes

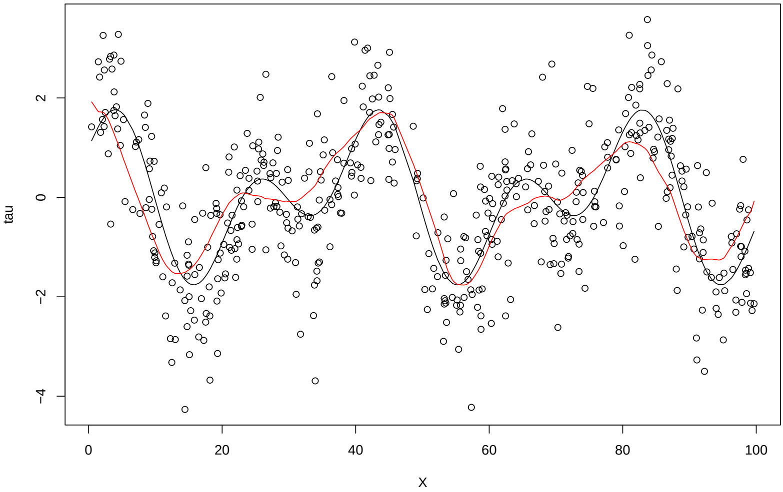
$$\arg\min_{\tau,\beta,\delta} \sum_{i=1}^{N} K\left(\frac{|X_i - x|}{h}\right)(Y_i - \tau D_i - \beta(X_i - x) - \delta D_i * (X_i - x))^2.$$

- $\hat{\tau}$ is our estimate of $\tau(x)$.
- Repeat this process for each x, we have an estimated curve $\hat{\tau}(x)$.

Kernel regression for estimating the CATE



Kernel regression for estimating the CATE



Bandwidth selection

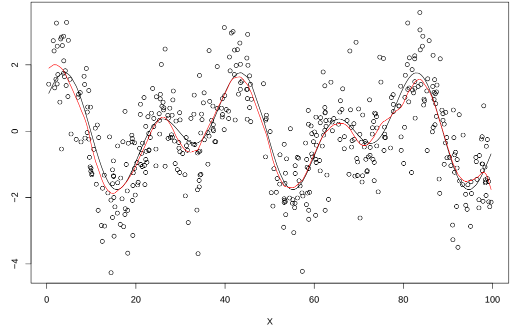
- For simplicity, let's return to the scenario where τ_i is known.
- Different bandwidths lead to different estimate $\hat{\tau}_h(x)$.
- We can find the optimal bandwidth h^* through cross-validation.
- Step 1, set a sequence of possible bandwidths, $h \in \{h_1, h_2, \ldots, h_H\}$ (e.g., $\{2, 4, \ldots, 20\}$).
- Step 2, randomly divide the sample into K folds (usually 5 or 10), denoted as $\{\mathcal{I}_k\}_{k=1}^K$.
- Step 3, for any unit $i \in \mathcal{I}_k$ and any h, fit the kernel regression model and generate the predicted value $\hat{\tau}_h(X_i)$ using units from $\cup_{k' \neq k} \mathcal{I}_{k'}$.
- Step 4, calculate the mean squared error (MSE):

$$\frac{1}{N} \sum_{i=1}^{N} (\tau_i - \hat{\tau}_h(X_i))^2$$

- Step 5, repeat Steps 1-4 for each h and find h^* that minimizes the MSE.

Bandwidth selection (*)

```
## The optimal bandwidth is 4
```



Bandwidth selection (*)

- In the cross-validation algorithm, we call $\mathcal{I}_k$ the test set and $\cup_{k' \neq k} \mathcal{I}_{k'}$ the training set.
- We fit the model on the latter and examine its performance on the former.
- Define $\varepsilon_i = \tau_i - \tau(X_i)$, which captures variation in τ_i that cannot be explained by $\tau(X_i)$ (the irreducible error).
- We can think $\tau(X_i)$ as the signal and ε_i the noise.
- Note that for $i \in \mathcal{I}_k$ and $j \in \cup_{k' \neq k} \mathcal{I}_{k'}$, ε_i is independent to ε_j.
- Therefore, we have

$$
\begin{aligned}
E\left[(\tau_i - \hat{\tau}_h(X_i))^2\right] =& E\left[(\tau(X_i) + \varepsilon_i - \hat{\tau}_h(X_i))^2\right] \\
=& E\left[(\tau(X_i) - \hat{\tau}_h(X_i))^2\right] + E[\varepsilon_i^2]
\end{aligned}
$$

- With cross-validation, the MSE measures how well $\hat{\tau}_h(X_i)$ approximates $\tau(X_i)$.

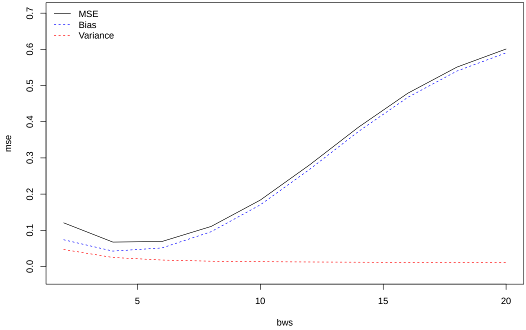
Bandwidth selection (*)

- Then,

$$
\begin{aligned}
& E\left[(\tau(X_i) - \hat{\tau}_h(X_i))^2\right] \\
=& E[\tau^2(X_i) - 2\tau(X_i)\hat{\tau}_h(X_i) + \hat{\tau}_h^2(X_i)] \\
=& \tau^2(X_i) - 2\tau(X_i)E[\hat{\tau}_h(X_i)] + E[\hat{\tau}_h^2(X_i)] \\
=& \tau^2(X_i) - 2\tau(X_i)E[\hat{\tau}_h(X_i)] + (E[\hat{\tau}_h(X_i)])^2 \\
& + E[\hat{\tau}_h^2(X_i)] - (E[\hat{\tau}_h(X_i)])^2 \\
=& (\tau(X_i) - E[\hat{\tau}_h(X_i)])^2 + Var[\hat{\tau}_h(X_i)]
\end{aligned}
$$

- The MSE equals the square of the bias plus the variance of $\hat{\tau}_h(X_i)$.
- It is typically a U-shaped function of h.
- h^* achieves the optimal trade-off between bias and variance.

Bandwidth selection (*)



Adaptive kernels

- In kernel regression, we select one bandwidth for all units.
- But it makes more sense to allow the bandwidth to vary.
- Classic methods can hardly do this.
- But we have machine learning now!
- One example is random forest, which can be interpreted as an adaptive kernel estimator (Athey et al. 2019).
- We randomly draw sub-samples from data and generate K bins to minimize the SSR.
- The collection of the K bins is known as a tree and each bin is a leave.
- We repeat this process to grow $1,000$ trees, which compose a forest.
- For any point x, the weight assigned to observation i equals the proportion of trees in which X_i and x are in the same leave.

References I

Athey, Susan, Julie Tibshirani, Stefan Wager, et al. 2019. "Generalized Random Forests." *The Annals of Statistics* 47 (2): 1148–78.