Program Evaluation (c)- k-NN Matching

Chris Conlon

March 30, 2021

Applied Econometrics

Matching Solution to Fundamental Problem

We don't observe the counterfactual $Y_i(T_i)$.

- Find observations with similar X_i and opposite T_i and hope they can be used as counterfactuals.
- Idea: Conditional on X_i, T_i is as good as randomly assigned.

$$Y_i = T_i \cdot Y_i(1) + (1 - T_i) \cdot Y_i(0)$$

i	$Y_i(1)$	$Y_i(0)$	T_i	X_i
1	1	?	1	1
2	0	?	1	2
3	?	0	0	1
		$\vdots$		
n	?	1	0	3

Matching

- Compare treated individuals to un-treated individuals with identical observable characteristics X_i.
- Key assumption: everything about $Y_i(1) - Y_i(0)$ is captured in X_i; or u_i is randomly assigned conditional on X_i.
- Basic idea: The treatment group and the control group don't have the same distribution of observed characteristics as one another.
- Re-weight the un-treated population so that it resembles the treated population.
- Once distribution of X_i is the same for both groups $X_i|T_i \sim X_i$ then we assume all other differences are irrelevant and can just compare means.
- Matching assumes all selection is on observables.

Matching

- Formally the key assumption is the Conditional Independence Assumption (CIA)

$$\{Y_i(1), Y_i(0)\} \perp T_i | X_i$$

- Once we know X_i allocation to treatment T_i is as if it is random.
- The only difference between treatment and control is composition $f(X_i|T_i)$ of the sample.

Nonparametric k-NN Matching: Abadie and Imbens (2002)

For each observation T_i, we observe $Y_i(T_i)$ compute a counterfactual $\hat{Y}_i(1 - T_i)$:

$$\widehat{Y}_i(0) = \begin{cases} Y_i & \text{if } T_i = 0 \\ \frac{1}{\#\mathcal{J}_M(i)} \sum_{l \in \mathcal{J}_M(i)} Y_l & \text{if } T_i = 1 \end{cases}$$

$$\widehat{Y}_i(1) = \begin{cases} \frac{1}{\#\mathcal{J}_M(i)} \sum_{l \in \mathcal{J}_M(i)} Y_l & \text{if } T_i = 0 \\ Y_i & \text{if } T_i = 1 \end{cases}$$

- $\#\mathcal{J}_M(i)$ is the number of matches for i of opposite treatment assignment $T_l = 1 - T_i$.
- M is the "number of matches" within some distance of $|X_l - X_i| < d_M(i)$.
- If there are ties $\#\mathcal{J}_M(i) > M$.
- This is just k-NN matching.

Nonparametric k-NN Matching: Abadie and Imbens (2002)

Each observation i gets a weight based on how often it is used as a match for other observations l:

$$K_M(i) = \sum_{l=1}^{N} 1\left\{i \in \mathcal{J}_M(l)\right\} \frac{1}{\#\mathcal{J}_M(l)}$$

Observations used in lots of matches get more weight. $\sum_{i=1}^{N} K_M(i) = N$.
This is just a weighted average of Y_i values (aka a kernel!):

$$ATE_M = \frac{1}{N} \sum_{i=1}^{N} \left[\widehat{Y}_i(1) - \widehat{Y}_i(0)\right] = \frac{1}{N} \sum_{i=1}^{N} \underbrace{(2T_i - 1)\left[1 + K_M(i)\right]}_{w_{i,M}} Y_i$$

Nonparametric k-NN Matching: Alternatives

Different weighting schemes give different parameters:

$$ATE_M = \frac{1}{N}\sum_{i=1}^{N}(2T_i - 1)\left[1 + K_M(i)\right]\cdot Y_i$$

$$ATT_M = \sum_{i=1}^{N_1}\left[T_i - (1 - T_i)\cdot K_M(i)\right]\cdot Y_i$$

$$ATUT_M = \sum_{i=1}^{N_0}\left[T_i \cdot K_M(i) - (1 - T_i)\right]\cdot Y_i$$

Nonparametric k-NN Matching: Bias Correction

We can use weighted least squares to adjust the predictions $\hat{Y}_i(T_i)$:

$$\left(\widehat{\beta}_{t,0}, \widehat{\beta}_{t,1}\right) = \operatorname{argmin}_{\{\beta_{t,0}, \beta_{t,1}\}} \sum_{i:T_i=t} K_M(i) \left(Y_i - \beta_{t,0} - \beta_{t,1}' X_i\right)^2$$

Where $\widehat{\mu}_1(X_i), \widehat{\mu}_0(X_i)$ are the regression functions for treatment and control.

$$\tilde{Y}_i(0) = \begin{cases} Y_i & \text{if } T_i = 0 \\ \frac{1}{\#\mathcal{J}_M(i)} \sum_{l \in \mathcal{J}_M(i)} \left\{Y_l + \widehat{\mu}_0\left(X_i\right) - \widehat{\mu}_0\left(X_l\right)\right\} & \text{if } T_i = 1 \end{cases}$$

$$\tilde{Y}_i(1) = \begin{cases} \frac{1}{\#\mathcal{J}_M(i)} \sum_{l \in \mathcal{J}_M(i)} \left\{Y_l + \widehat{\mu}_1\left(X_i\right) - \widehat{\mu}_1\left(X_l\right)\right\} & \text{if } T_i = 0 \\ Y_i & \text{if } T_i = 1 \end{cases}$$

So that the ATE is given by: $ATE_M = \frac{1}{N} \sum_{i=1}^{N} \left\{\tilde{Y}_i(1) - \tilde{Y}_i(0)\right\}$

What about higher dimensions?

- We know that nearest neighbor is cursed in high dimensions.
  - Usual caveats apply: may be doing extrapolation.
  - Even more reason to use regression/bias adjustment.
- Given two vectors $\mathbf{x}$ and $\mathbf{y}$, how to choose $d(\mathbf{x}, \mathbf{y})$ the distance function?
- Papers mostly use Mahalanobis distance: $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y}) S^{-1} (\mathbf{x} - \mathbf{y})'$.
  - Quadratic distance with inverse covariance matrix as “weights”.
  - Generalizes Euclidean distance (diagonal S).
- Older papers use caliper matching anything within $\|\mathbf{x_s} - \mathbf{x_t}\| < b$ is match
  - Now number of matches varies from observation to observation.
  - Variance can be unpredictable: some $(\mathbf{y_i}, \mathbf{x_i})$ have many of matches, others have none
  - Some obs may have nothing within $\|\mathbf{x_s} - \mathbf{x_t}\| < b_w$? Drop these?
  - Probably avoid this unless you have a good reason...

Implementation

- Base case is easy enough to do on your own using a canned k-NN package.
- Even regression/bias adjustment is pretty easy
- Computing ATE, ATT, ATUT is relatively straightforward on the matched sample
- `MatchIt` in `R` (by Gary King et. al) or `nnmatch` in Stata (by Abadie and Imbens et al)
- Documentation here is pretty good: `https://scholar.harvard.edu/files/imbens/files/sjpdf-1.html_.pdf`.