**Summarizing what we did with matching**

1. When should we use any of the methods under the “backdoor criterion / conditional independence” or “selection on observable” methods?

As a rule, these methods are very demanding, not on the data, but on your confidence that you *know* the confounders you need to control for, you know they’re not colliders or bad controls, and they are in your data. Not that you know 7 confounders – all of the confounders. That requires some kind of prior theory about the phenomena you’re studying.

The thing to remember is with people, they are typically thought to be choosing the treatments for reasons that are directly connected to what they think it will do for them. The more correct they are, the better that they are at choosing appropriate treatments, the more selection bias we have baked into the data. Perfect doctor from earlier. A DAG is not perfect, but a DAG is a causal notation that will provide you with guidance as to which variables you need to focus on (and which ones to avoid) when adjusting comparisons for covariates.

Once you have chosen the covariates, though, you typically are in a world of “many covariates” and in a finite sample, adjusting for many covariates necessarily will have a dimensionality problem called the “curse of dimensionality”. Your goal is to either reduce that dimension into something manageable through variable selection (ML approach) or use some of these canonical approaches like propensity scores (propensity score theorem allows you some confidence that the dimension of K reduced to 1 scalar did not lose information).

Advantages of the propensity score are numerous and often don’t even require using propensity score estimation.

* Visualize common support problems with histograms. Look at the distribution of 1000 covariates all interacted across two groups – impossible. But with a propensity score that is the collapsed scalar version of those k^1000 strata it’s actually easy. Common support is one of the key assumptions in your analysis.
* Propensity score can enter as a weight either for direct inverse probability weighting with bootstrapping or OLS with the weights as analytical weights on the observations (getting standard errors is simple).
* Addressing the common support problem. Just trim – just trim in a principled way. Crumb, et al. 2009 suggest trimming off the 0.1,0.9 interval. More recently you’re seeing more work by the machine learning community to look at even more principled ways but at worse, simply trimming bottom part and the top part is a good start. At least you’re not getting a bunch of drops like you get with caliper or radius matching.
* Problem with propensity scores: if you have model misspecification, then it can have major relevance for the bias of estimation.

Covariate matching is an alternative, but covariate matching will have matching discreprancy bias increasing in the number of covariate dimensions *k* (strata). But, there is a “matching bias correction” that you can use to reduce that bias. And at Twitch, you may have so much data that what traditionally plagues social scientists is a trivial or even nonexistent problem for you because the matching bias shrinks to zero as n grows to infinity. But the problem is we know the convergence rate gets slow as *k* increases too. So probably even at Twitch, you’re going to need to do some kind of bias adjustments bc there will be discreprancies, even as you grow n.

But the main thing I want you to be thinking of is this: matching is not a “kitchen sink” approach. When you hear people say “rich controls to match on”, you need to ask yourself are any of these controls confounders in my DAG? If they give you 10,000 covariates and not a single one of them makes sense in a DAG, *you are not gaining anything from including them*. You have to think theoretically if you are going to go this route.

My advice:

* Use DAGs to help you make choices about covariate selection
* Then use propensity scores to help guide you to think about the severity of common support problems
* Consider the 0.1 and 0.9 interval trimming as at least a robustness preferred method for addressing that off support selection problem for which the matching will likely be bad.
* Be very cognizant of which parameter you are estimating. If you are estimating the ATE, hyou have to get counterfactuals for the control group – does your treatment group have people in it that you think “look like” the control group? Remember the CPS example. Oftentimes the data may really only support or justify the ATT.
* Inverse probability weight with propensity scores on a trimmed sample and nearest neighbor covariate matching with bias adjustment involve the fewest subjective researcher judgments. The fewer subjective researcher judgements or what is sometimes called “researcher degrees of freedom” (ie researcher bias), the more principled the approach is, the easier to defend the choices.

Just remember – we don’t *know* the average treatment effect. We *estimate* them and our estimates are only reliable if the identifying assumptions fit the method. The biggest one here is “backdoor criterion”. It leads directly to the conditional independence assumption, and then you need to check common support, explore covariate imbalances, and make decisions abt weighting vs imputation.

ATT = 1/Pr(D=1) x E[Y x (-p)/(1-p)]

Really high propensity scores in the control group can blow up the calculation and the really small ones can evaporate it. You’re partly trimming to address that instability in the way weights are constructed.

When you’re doing nearest neighbor it doesn’t provide that protection. You could say “Well, 15,000 people have propensity scores in the control group of 0.000000000000001. And the lowest propensity score in the treatment of 0.01. You have to decide – is that a close number? Is that a far number? You could drop those but then the same issues show up all throughout the range of the propensity score because what you are doing in NN is matching a close neighbor to each and every treatment unit (for the ATT). Dropping <0.1 and >0.9 does nothing for a unit who has a score of 0.25 and the nearest neighbor is 0.23. Nearest neighbor – the problem is again matching discreprancies are sort of harder to exactly detect compared to the covariate NN matching. There is no matching bias adjustment in NN propensity score matching unfortunately which is why these old papers used “brakes”. Radius, calipers – they’d decide *ex ante* on a number that the researcher said disqualified a unit as being a good control. There is also a paper by Gary King, and coauthors. Read that, make your own decision, I will tell you sort of where my gut leads me.

With propensity score NN matching, you often end up dropping units all over the propensity score bc there isn’t always someone within 0.X distance of someone else. In a finite sample, it’s not non-zero mass at every value of the propensity score. You never find someone with the EXACT propensity score. I think I probably lean towards IPW with trimming bc it isn’t this one after another after another ad hoc choice that you really can’t justify. Why did you say 0.01 and not 0.005? Most of us know we sort of pulled out that number bc it seemed to us “small”.

**To do:**

Comment by ivachau: I have 2 kind of related questions 1) How do we think about variations in treatment effect across propensity scores? 2) what about cases where the overall matched covariate imbalance looks good, but is imbalanced for groups in certain propensity score ranges

So number one is a question about heterogenous treatment effects, or that’s a way to restate it. Heterogeneity across the propensity score.

Is this the propensity score definition for the ATE or the ATT?

Check this: # inverse propensity score weights

df\_nonexp <- df\_nonexp |>

mutate(

inv\_ps\_weight = treat \* 1/pscore + (1-treat) \* 1/(1-pscore)

)