**Stat 992: Causal Inference**

Fall 2019 Syllabus

University of Wisconsin-Madison

Class Project

­­­­**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

For the class project, you can work alone or in a group of two to four. You will be expected to (i) present a talk on your project to the class similar to a JSM-contributed session talk (15 minutes in length) and (ii) write a short paper using the LaTeX template for the *Annals of Applied Statistics* or an R package with a vignette (depending on the project below).

The due dates and talks will be scheduled during the latter half of the class.

I recommend that you choose one of the project topics below. However, should you choose to work on a project different than what’s listed below, please come talk to me before October 1st before writing up a 1-page proposal.

Project 1: Does considerable loss in income late in life cause an increase in all-cause mortality? Is there treatment heterogeneity in this causal effect? See <https://jamanetwork.com/journals/jama/article-abstract/2677445>

This project would replicate the analysis presented in the paper, but by using any causal method of your choice from class. One suggestion that I have is matching-based technique called risk-set matching:

1. Li, Propert, and Rosenbaum <https://amstat.tandfonline.com/doi/abs/10.1198/016214501753208573>
2. Haviland, Nagin, Rosenbaum: <https://psycnet.apa.org/record/2007-12911-001>
3. Chapter 12 of Rosenbaum’s Design of Observational Studies.

You’ll compare the results from using approach to analyzing this causal question to the time-varying IPW weighting that was used in the paper. You’ll also conduct a sensitivity analysis to see if the conclusions are sensitive to unmeasured confounders. You are required to document all of your coding/data cleaning work with Rmarkdown. Visualizing your data analysis for wider consumption (i.e. i.e. grandma can understand it) is strongly encouraged.

If you cannot obtain the full public data (HRS at UMich. and RAND Corporation), come talk to me EARLY in the semester (before Oct. 1st) and we can define a slightly new goal.

You’re also more than welcome to study a different exposure-outcome in the HRS data. For example, there is considerable interest in studying the causal effect of when/how you retire on important health outcomes.

Project 2: Build a fast matching algorithm for big datasets.

If you like building lm() in R from scratch or optimizing lm() for big data, both from an implementation as well as an algorithmic standpoint, this project will be most interesting to you.

Let k = max(n-m,m) where m is the number of treated units and n is the total sample size. From the late 1980s to this year, the fastest matching algorithm (in the worst case) is O(k^3); this is the computational limit for optmatch. In 2019, this was improved to O(k^2 log(k)) by Ruoqi, Paul Rosenbaum’s (now 4th year) student, under some conditions; see her bigmatch package here: <https://cran.r-project.org/web/packages/bigmatch/bigmatch.pdf>

From a practical standpoint, if your sample size is around 200k-ish, most matching algorithms will crash, even on a compute grid, due to memory and speed limits.

Your goal is to implement a fast matching algorithm that can scale to 600k individuals (roughly 100k treated, 500k control) and still achieve good covariate balance. You can approach this in two ways

1. Greedy/stochastic/approximation algorithms. If you can (a) approximate the optimal matched set solution, (b) this approximate optimization can be solved very quickly, and (c) you can characterize the gap between this approximate solution and the original matching problem, this would be the perfect algorithm. For an example, see <https://www.tandfonline.com/doi/full/10.1080/10618600.2019.1584900>

A word of caution: if you go “too” greedy (i.e. do a few-epoch/passes at data, find the closest match per person without looking at the optimization problem globally), you might be sacrificing covariate balance for speed. This problem has been well-documented since the early 1990s and it’s different in structure (I think) than SGD.

1. Improve the underlying implementation of optmatch. My friend Colin Fogarty has told me that if you use an NP-hard algorithm (based on integer programming) to solve matching, it is actually faster than optmatch’s polynomial time algorithm based on network flow.

Finally, I can tell you that the “Pythonic/Julia/C/Go” solution is generally not promising; the current optmatch software is implemented in Fortran.

You are required to produce an R package with a vignette should you choose to do this project; no paper is necessary. The latter is mainly for me (and maybe Ben Hansen, see below) to compare all of your efforts to see which group has the fastest matching algorithm in a “mini” class competition.

If your matching algorithm can do matching in less a minute for a sample size of around a million, this is a huge accomplishment.

Some additional references on algorithms for matching are here:

1. Liz Stuart’s 2010 review on matching: <https://projecteuclid.org/euclid.ss/1280841730>
2. Paul Rosenbaum’s 2019 review on matching: <https://www.annualreviews.org/doi/abs/10.1146/annurev-statistics-031219-041058>
3. Jose Zubizarreta’s mixed integer programming approach to matching: <https://amstat.tandfonline.com/doi/full/10.1080/01621459.2012.703874>
4. Sam Pimentel’s sparse matching algorithm for refined balance: <https://www.tandfonline.com/doi/abs/10.1080/01621459.2014.997879>, R package details here: <https://pdfs.semanticscholar.org/90f2/61e24b19a6a53bf56204831f9831b9ac3214.pdf>

Finally, Ben Hansen, the inventor of optmatch, is coming to give a talk this semester. If you have a working version of your matching algorithm by the time he visits, it could be a great opportunity to get feedback from him ☺.

Project 3: Fast Algorithms for Exact Permutation Testing (Under Non-Sharp Null Hypothesis)

See papers:

1. Rigdon and Hudgens: <https://onlinelibrary.wiley.com/doi/full/10.1002/sim.6384>
2. Li and Hudgens: <https://onlinelibrary.wiley.com/doi/full/10.1002/sim.6764>
3. Ding: <https://projecteuclid.org/euclid.ss/1504253116>
4. Lee and Romano: <https://projecteuclid.org/download/pdfview_1/euclid.aos/1366138199>
5. Tritchler: <https://www.jstor.org/stable/2288357>
6. Bahadur and Savage: <https://www.projecteuclid.org/euclid.aoms/1177728077>

Consider testing the null hypothesis of no average treatment effect (i.e. non-sharp, composite null hypothesis) using common test statistics (e.g. difference in means, rank-based tests, M-statistics). For some time, computing the exact p-value/confidence intervals has been limited to binary outcomes (papers #1,2) or using asymptotic approximations (paper #4); paper #3 summarizes this debate from a philosophical perspective. Paper #5 provides a way to do compute exact p-values for sharp null hypothesis based on characteristics functions and the Fast Fourier transforms. Paper #6 hints at a negative result which suggests that the goal is impossible.

Your task is to do two of the following:

1. Read Tritchler carefully and summarize the results in the paper. Your summary must have working code and a real data example.
2. Come up with a fast way to exactly test the average treatment effect in non-sharp, non binary outcome setting (e.g. discrete, bounded outcomes)
3. Reinterpret Bahadur and Savage’s paper and prove a negative result that states that this is impossible, either statistically or using polynomial time algorithms.

The coding portion must be documented in RMarkdown. You are required to produce a short paper as a part of this project.

Other Project Topics

If you plan on working on topics in the areas specified below, let me know before the proposal deadline on what you plan to do.

1. Negative controls: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3053408/>
2. Diff-in-diff: see any applied microecon paper.
3. Visualizing matching with clustering: see Jiongyi Cao’s efforts to do this here: <https://github.com/jiongyi-cao/tSneClstGRF>
4. Machine learning for heterogeneous treatment effects: This field is getting fairly dense/technical and it’s better to work on a very specific instance of this problem. Talk to Michael Johnson or Chan Park for specific instances of this problem here: <https://arxiv.org/abs/1908.03652>, <https://arxiv.org/abs/1908.04427>
5. Mendelian randomization (MR)/IV projects. IV and MR is vast, but MR is ripe for a good stat student to get into. Thankfully, the problems in MR are parametric in nature and requires no more than MLE theory. Talk to Sheng Wang for some example problems in this area.
6. Matching for heterogeneous treatment effects: Michael had to recently deal with this issue in his recent project on instruments and heterogeneous treatment effects. Talk to him for specifics.
7. Personalized medicine, policy learning: a recent book summarizing this literature is here <https://epubs.siam.org/doi/book/10.1137/1.9781611974188>.
8. Synthetic controls: this is the “newest” causal method in town. See Abadie, Diamond, Hainmueller: <https://amstat.tandfonline.com/doi/abs/10.1198> and <https://onlinelibrary.wiley.com/doi/full/10.1111/ajps.12116> and Ding and Li: <https://www.cambridge.org/core/journals/political-analysis/article/bracketing-relationship-between-differenceindifferences-and-laggeddependentvariable-adjustment/2FDDA72EBC5979561F1D0414329F003E>
9. Deep learning and causal inference: see Farrell, Liang, Misra: <https://arxiv.org/abs/1809.09953> and Shi, Blei, and Veitch: <https://arxiv.org/pdf/1906.02120.pdf>
10. Peer effects in causal inference: This literature is fairly new, yet quite dense. First, read these applied papers by Sacerdote: <https://academic.oup.com/qje/article-abstract/116/2/681/1904199>, Nickerson: <https://www.cambridge.org/core/journals/american-political-science-review/article/is-voting-contagious-evidence-from-two-field-experiments/8C2E64552D946C87FD062DD2CCD9054E>, Fafchamps, Vaz, and Vicente <https://www.journals.uchicago.edu/doi/abs/10.1086/700634?journalCode=edcc>, Coppock, Guess, and Ternovski: <https://link.springer.com/article/10.1007/s11109-015-9308-6>. Afterwards, read the causal papers on networks by Hudgens and Halloran: <https://www.ncbi.nlm.nih.gov/pubmed/19081744>, Ogburn and Vanderweele: <https://projecteuclid.org/euclid.ss/1421330547>, Basse, Ding, Feller, Toulis: <https://arxiv.org/abs/1904.02308>, Li, Ding, Lin, Yang, Liu: <https://arxiv.org/abs/1807.01635>. Let me know if you are interested in working on something related to this.