HW 2: Average Treatment Effects and Heterogeneous Treatment Effects

See the course syllabus for more instructions about working in teams. Students should turn in individual write-ups but may collaborate on code.

Getting Data and R Packages

For the first part of this assignment, please continue with the "altered" experimental dataset you used in the last assignment. For the second part, go back to the original "un-altered" experimental dataset. (Or you can switch datasets for the second part—but you should work with an experimental dataset.)

To install packages:

- You need to install the package "devtools"
- On Windows, you also need RTools installed. That is available here: https://cran.r-project.org/bin/windows/Rtools/
    - R will prompt you to install if you have not. I found that I had to copy the files that were installed from c:\rbuildtools\3.4\ into a different directory, c:\Rtools (with no version number, e.g. no 3.4 subdirectory, below it)
- There is a bug in R 3.4 that prevents installing packages. To get the patch, find it here: https://cran.r-project.org/bin/windows/base/rpatched.html

For causalTree:

- To install:
    - install_github("susanathey/causalTree")
- To see some examples of how to use causalTree, see this:
    - https://github.com/susanathey/causalTree/blob/forestCode/test/test_causalTree.R

Specific Assignment

For your assignment:

Part I: Heterogeneous Treatment Effects in Observational Studies

- In the first part of this HW, you may re-use the data set from your first homework, including the artificial confounding. We start by considering different random forest based strategies for estimation heterogeneous treatment effects (HTE).
    - Use the regression_forest function in grf to estimate HTEs via the S-learner strategy.
    - Use two calls to regression_forest to estimate HTEs via the T-learner strategy [see the optimal policy portion of the tutorial for code to estimate two separate prediction forests, one for treated and one for control, and make predictions on the whole training dataset].
    - Use several calls to regression forest to estimate HTEs via the X-learner strategy.
    - Estimate HTEs using the causal_forest function.
    - Compare HTE estimates obtained via the 4 strategies, and plot them using some of the methods discussed in lecture 4. Which ones do you trust?
- We'll now change the size of the dataset used above, and see how the different methods are affected by this. For each bullet point below, re-run the 4 methods from above, and check whether their answers are consistent with what you found above.
    - Draw a random subset of 20% of your data (or 400 observations, whichever is greater), and re-run all methods.
    - Subset the data such that there are exactly the same number of treated and control units (e.g., if you started with 1442 treated and 2056 controls, now just keep 1442 from both categories), and re-run all methods.
    - Subset the data such that there are 5x more control units than treated units (unless that leaves you with less than 400 observations, in which case you should modify the ratio), and re-run all methods.
    - Which methods are most stable when we alter data size?

Part II: Heterogeneous Treatment Effects in Randomized Experiments

- Return to the un-altered randomized experiment. Use random sampling to divide the dataset into three datasets, call them df_tr, df_est, and df_test as in the tutorial.
- Using https://github.com/susanathey/causalTree, use the command honest.causalTree to build and prune an honest Causal Tree. See the tutorial for sample code.
    - Create a factor variable for the leaves in samples df_tr, df_est and df_test, and run linear regressions that estimate the treatment effect magnitudes and standard errors (see tutorial code) in each sample.
    - Compare your results in samples df_tr, df_est and df_test. How do your results in sample df_tr differ from those in samples df_est and df_test?
- Estimate heterogeneous treatment effects using causal_forest from the grf package.
    - Pick a few values for the covariates (you may select a few points from the test set, or create specific covariate values useful for visualizing heterogeneity). Report the

    estimated treatment effects and standard errors from grf. (Note that eliminating Monte Carlo errors in confidence intervals may require a larger number of trees than simply getting stable point estimates.)
  - Reduce the size of the dataset to 90%, 70%, and 50% of its original size. Report the estimated treatment effects and standard errors for each test point, for the different data set sizes. Interpret your results, particularly what they tell you about the reliability of the confidence intervals in your particular dataset.

Part III: Estimating optimal policies

- For this part, can follow the sample code in the tutorial closely.
- Continuing with a randomized trial, use causal_forest in GRF to estimate the CATE function. Then:
  - Estimate the simple sample average treatment effect overall in the training and test data (the tutorial does this with a simple regression on dummy variables, so that you also get the standard errors neatly).
  - Group the units into two groups, those with positive CATE and those with negative CATE. Define a treatment assignment policy that assigns the treatment to those with a positive CATE. (If most of the units have a positive CATE or a negative CATE, shift the CATE's by a constant to continue the exercise, e.g. imagine that the treatment has a cost about equal to the mean treatment effect, as in the tutorial. See the tutorial for code to define a factor variable equal to the sign of a variable.).
  - In both the training and test data, estimate the sample average treatment effect for the units assigned to be treated by your policy; and estimate the sample average treatment effect for the units assigned to be control by your policy (the tutorial does this using a simple regression of outcomes on dummy variables). Interpret each of these (adjusting the sign to be opposite in the group assigned to control) as the difference between using your policy and a random policy (see the tutorial for details). Estimate the value of your policy relative to a random policy as the weighted sum of the (weighting the first group by the fraction assigned to be treated, and the second group by the fraction assigned to be control, and being careful to take the negative treatment effect on the group assigned to control.)
  - Use a classification tree (e.g. from rpart) to estimate a simpler policy. See the tutorial for implementation. In particular:
    - Use GRF to estimate $\hat{\tau}$ (causal_forest) and $\hat{\mu}_w$ (regression_forest). Create a new, augmented dataset as in the tutorial and estimate the policy using the double-robust scores as the weights. Use a fixed depth tree. Visualize the resulting policy assignments.
    - Estimate the value of your policy relative to a random policy, following the approach from above and details in the notes.
    - Estimate this value on both the training data and the test data.
  - Interpret your results, including differences between training and test.

Your write-up should include code with output (preferably generated by “knitting” as per the R instructions provided in the tutorial, although sometimes that has bugs), as well as an electronic document (submitted individually) that discusses the results. Try to make your document self-contained by pasting in figures and referring to specific numbers/standard errors in the text where relevant. If you worked with group members on your code, indicate the group members on the assignment, but your write-ups should be done individually, and each member should submit the code/knit file.