**iRCT**

Garrett Barber, Xia Jiang

**Abstract:**

**Background:**

**1. Introduction**

This paper is focused on the introduction of iRCT (intelligent Randomized Control Trial) – A causal learning method package for python based on many different methodologies, primarily with a focus on the use of matching estimators, combined with average treatment effect. iRCT currently supports 6 different methods being 1) iRCT – the current main method of iRCT which uses propensity scoring and weighting in order to determine an average treatment effect; 2) Multi-covariate iterative iRCT – this method is a model of iRCT that uses a simple iteration method in order to match based on a propensity score that is produced from multiple covariates; 3) Single-covariate iRCT – based on Alberto Abadie’s work [1], using a simple method of matching on a single covariate and then computing the average treatment effect; 4) Inverse probability treatment weighting (IPTW) – similarly to the base iRCT it calculates a propensity score for all entries and then based on the balance of treated versus untreated entries an inverse probability weight is derived and applied to each propensity score to create an unbiased average treatment effect; 5) Parametric G-Formula – using the parametric G-Formula on the dataset given a treatment and outcome, then determining the effect if all were treated then determining the effect if none were treated, and subtracting the two in order to determine the effect of the treatment on the outcome. There are plans to add two more methods being the java implementation and python implementation of MBIL respectively. There are also plans to increase the efficiency of the Multi-covariate iterative iRCT and Single-covariate iRCT.

We will introduce it from 4 parts being main technology tools, propensity scoring, detailed description of each method and limitations. The main purpose of the iRCT package is to congregate multiple single variable causal learning methods in one package in order to allow for ease of use.

**2. Main Technology Tools**

The package was developed in Python due to the lack of packages for Python with multiple causal learning methods. One of the main features is reading the dataset into a pandas dataframe before any calculations are done [2]. In order to do any propensity scoring calculations the sklearn package was utilized and the logistic regression method was called [3]. For individual methods such as IPTW, there was a package already developed known as causalinference which was implemented [4]. The same applies to the G-Formula method which was once again already developed and was implemented using the zepid package [5].

**3. Propensity Scoring**

As used in multiple methods, propensity scoring is defined as “a conditional probability of being exposed given a set of covariates” [6]. Propensity scoring is implemented in the iRCT method, Multi-covariate iterative iRCT method, and IPTW, however for IPTW it is done within the causalinference methods provided by the package. For the iRCT method and Multi-covariate iterative iRCT method the sklearn logistic regression method is used with the inputs being the treatment column and every other column other than the treatment and outcome columns. The purpose of using propensity scoring is in order to allow for matching and comparison based upon multiple covariates.

**4. Description of each of the methods**

iRCT is a method that uses propensity scoring for each entry. After determining the propensity score the weight is calculated by comparing the treatment effects divided by the propensity scores. This weight is then applied to the outcome of each entry and is then averaged in order to determine the average treatment effect.

Multi-covariate iterative iRCT also uses propensity scoring in order to determine what to match upon. The dataframe is then iteratively processed and a match(es) is found. The outcomes are then subtracted from each other and all values are added together and averaged in order to determine an overall average treatment effect.

Single-covariate iRCT uses a single given covariate to match upon and then performs the same iterative and outcome calculation process as Multi-covariate iRCT in order to determine the average treatment effect.

IPTW or inverse probability treatment weighting works the same as iRCT except the weight effects are calculated differently. The weight is determined as 1/propensity score for exposed individuals and 1/(1 – propensity score) for unexposed individuals [7]. This weighting is then used to determine the average treatment effect.

G-Formula is an analytical tool to standardize outcome distributions using the covariates present in order to determine an average treatment effect [8]. This allows for the removal of bias coming from time-varying covariates and other confounding factors.

**5. Limitations**

Currently there are only 4 different types of files that may be uploaded being txt, csv, xlsx, and dat. The dataset must already be completely numerical, as well as the treatment column must be binary in nature. There are functions currently in the works in order to automatically transform datasets and give the mapping for the transformation. The iRCT package is available for install via (insert installation method here) and accessible via the iMedCausal website using the URL *http://imedcausal.odpac.net/.*

**6. Comparison of Methods**

As the Multi-covariate iterative iRCT and Single-covariate iRCT are still under development due to inefficiency, this section will be comparing iRCT, IPTW, G-Formula, and MBIL-py. For this comparison, all algorithms were run using the COVID3\_4Nodes3 dataset. This means for iRCT, IPTW, and G-Formula, all possible combinations of treatment and outcome were tested, and for MBIL-py every variable was set as the target. Shown in table 1 is the results of this experiment.

**7. Conclusion**

The iRCT package allows for ease of access of multiple single target causal learning methods. Through the package users will be able to compare results of different methods and gain new insight into the effects of different variables from previously conducted randomized control trials.

**Reference**

[1] *Pandas*. pandas. (n.d.). Retrieved March 27, 2023, from https://pandas.pydata.org/

[2] Abadie, A., Drukker, D., Herr, J. L., & Imbens, G. W. (2004). Implementing matching estimators for average treatment effects in stata. *The Stata Journal: Promoting Communications on Statistics and Stata*, *4*(3), 290–311. https://doi.org/10.1177/1536867x0400400307

[3] *Sklearn.linear\_model.logisticregression*. scikit. (n.d.). Retrieved March 27, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html

[4] *Causal inference in python¶*. Causal Inference in Python - Causalinference 0.1.3 documentation. (n.d.). Retrieved March 27, 2023, from https://causalinferenceinpython.org/

[5] *Zepid¶*. zEpid. (n.d.). Retrieved March 27, 2023, from https://zepid.readthedocs.io/en/latest/index.html

[6] *Propensity score analysis*. Columbia University's Mailman School of Public Health. (2022, October 3). Retrieved March 27, 2023, from https://www.publichealth.columbia.edu/research/population-health-methods/propensity-score-analysis

[7] Chesnaye, N. C., Stel, V. S., Tripepi, G., Dekker, F. W., Fu, E. L., Zoccali, C., & Jager, K. J. (2021). An introduction to inverse probability of treatment weighting in observational research. *Clinical Kidney Journal*, *15*(1), 14–20. https://doi.org/10.1093/ckj/sfab158

[8] Keil, A. P., Edwards, J. K., Richardson, D. B., Naimi, A. I., & Cole, S. R. (2014). The parametric G-formula for time-to-event data. *Epidemiology*, *25*(6), 889–897. https://doi.org/10.1097/ede.0000000000000160