Introduction to `IPTWtrim` R package

Xiaoya Wang

July 31, 2023

Contents

Welcome to the `IPTWtrim` package handbook! This vignette provides examples and usage instructions for the functions available in the `IPTWtrim` package. The package is designed to facilitate causal inference with inverse probability of treatment weighting (IPTW) and weight trimming. It enables users to estimate the Average Treatment Effect (ATE) using weight trimming and evaluate the performance of trimming for different trimming percentile based on simulation studies for a specific setting. We focus on the setting where three confounding variables $\mathbf{X} = (X_1, X_2, X_3)$ follows a multivariate normal distribution, treatment A is generated by a logistic regression model including all confounding variables $\mathbf{X}$, and the response variable Y is a linear combination of confounding variables $\mathbf{X}$, treatment A and a standard normal distributed error term.

1 Brief Introduction to IPTW Trimming

IPTW trimming is a statistical method used for estimating the Average Treatment Effect (ATE) in observational studies with confounding variables. It involves assigning weights to observations based on the probability of treatment given observed confounders. However, sometimes extreme weights can lead to biased estimates. Some researchers mentioned that IPTW trimming may address this by capping extreme weights at specific percentiles, improving estimation reliability (Stuart, 2010).

2 Introduction to `IPTWtrim` R package

2.1 Overview

The `IPTWtrim` R package is a powerful tool designed for causal inference using Inverse Probability of Treatment Weighting (IPTW) with symmetric weight trimming. It provides a set of functions to estimate the Average Treatment Effect (ATE) in observational studies, particularly when dealing with confounding variables that may affect both the treatment assignment and the outcome of interest. Also, it can assess the performance of different weighting percentile.

The `IPTWtrim` package has the following architecture:

```
## /Users/yaya/Desktop/IPTWtrim
## +-- DESCRIPTION
## +-- IPTWtrim.Rproj
## +-- LICENSE
```

```
## +-- LICENSE.md
## +-- NAMESPACE
## +-- R
## |   +-- IPTWtrim-package.R
## |   +-- dataGenFun.R
## |   +-- measureATEfun.R
## |   \-- trimSummary.R
## +-- README.md
## +-- man
## |   +-- dataGenFun.Rd
## |   +-- measureATEfun.Rd
## |   \-- trimSummary.Rd
## +-- tests
## |   +-- testthat
## |   \-- testthat.R
## \-- vignettes
##       +-- IPTWtrim_handbook.Rmd
##       +-- IPTWtrim_handbook.html
##       \-- references.bib
```

2.2 Basic examples for data generation

2.2.1 Installation

First, install and load the required R packages (`MASS`, `ggplot2`, `gridExtra` packages).

```r
install.packages("MASS")
install.packages("ggplot2")
install.packages("gridExtra")
library(MASS)
library(ggplot2)
library(gridExtra)
```

First, install and load the R package.

```r
library(devtools)
install_github("Dorayaya/IPTWtrim",
               build_vignettes = TRUE)
library(IPTWtrim)
```

2.2.2 `dataGenFun()`: Generate Data for Our Specific Setting

Now, we can use `dataGenFun()` function to generate data for a binary treatment (A) and continuous confounding variables (x1, x2, x3). The confounding variables are assumed to be generated from a multivariate normal distribution with specified mean and covariance matrix. The binary treatment A is modeled using a logistic link function with specified coefficients. The continuous outcome Y is generated based on the treatment and confounders using a linear model with normally distributed errors.

This table gives the description for each argument, and the default values are set according to the need of my STAT900 final project.

Argument	Description	Default Value
n	The number of observations to generate from the normal distribution. Should be a positive integer.	500
mu	The mean vector of the multivariate normal distribution for predictors x1, x2, x3.	c(10, 1, 11)

Argument	Description	Default Value
`sigma.mat`	The covariance matrix of the multivariate normal distribution for predictors x1, x2, x3.	`matrix(c(11^2, 4, 4, 4, 3^2, 3, 4, 3, 8^2), 3)`
`eta`	The coefficients of the logistic regression model for the binary outcome A.	`c(-1.5, 0.1, 0.3, 0.2)`
`theta`	The coefficients of the linear regression model for the continuous outcome Y.	`c(110, -12, -0.3, -0.8, -0.2)`

```
dataExample <- dataGenFun()
head(dataExample)
```

References

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1.