Causal Inference in Python: A Vignette

Laurence Wong

May 20, 2015

Installation

CausalInference can be installed using `pip`, and will run provided the necessary dependencies are in place. On Ubuntu systems, the following commands should take care of all the essential steps if you are starting from scratch:

```
$ sudo apt-get update
$ sudo apt-get install python-pip python-numpy python-scipy
$ sudo pip install causalinference
```

Minimal Example

The main object of interest in CausalInference is the class `CausalModel`. It takes as inputs three NumPy arrays: `Y`, an N-vector of observed outcomes; `D`, an N-vector of treatment status indicators; and `X`, an N-by-K matrix of covariates. The following code snippet illustrates how the class can be invoked using random data simulated from NumPy's `random` module.

```
>>> import numpy as np
>>> from causalinference import *
>>> Y = np.random.rand(1000)
>>> D = np.random.randint(0, 2, 1000)
>>> X = np.random.rand(1000, 3)
>>> causal = CausalModel(Y, D, X)
```

Class Attributes and Methods

Once an instance of the class `CausalModel` has been created, it will contain a number of attributes and methods that are relevant for conducting a causal analysis. Tables 1 and 2 contain a brief description of these attributes and methods.

Attribute	Description
`N, N_c, N_t, K`	Integers indicating sample sizes and number of covariates.
`covariates`	Dictionary-like object containing summary statistics for the covariate variables.
`pscore`	Dictionary-like object containing propensity score data, including estimated logistic regression coefficients, predicted propensity score, maximized log-likelihood, and the lists of the linear and quadratic terms that are included in the regression.
`cutoff`	Floating point number specifying the cutoff point for trimming on propensity score.
`blocks`	Either an integer indicating the number of equal-sized blocks to stratify the sample into, or a list of ascending numbers specifying the boundaries of the strata.
`strata`	List-like object containing the list of stratified propensity bins.
`est`	Dictionary-like object containing treatment effect estimates for each estimator used.

Table 1: Attributes of the class CausalModel. Invoking `print` on any of the dictionary- or list-like attribute above yields customized summary tables. Note that some attributes are only created after the relevant methods have been called.

Using CausalInference Interactively

```
>>> print causal.covariates
Covariates Summary
                     Controls (N_c=267)          Treated (N_t=233)
      Covariate        Mean          S.d.         Mean          S.d.     Nor-diff
---------------------------------------------------------------------------------
             X0      -0.507         0.936        0.390         0.907        0.972
             X1      -0.295         0.902        0.380         0.905        0.748
             X2      -0.514         0.950        0.367         0.871        0.967
```

Estimate propensity score.

```
>>> causal.propensity_s()
```

Method	Description
`restart`	Reinitializes data to original inputs, and drop any estimated results.
`propensity`	Estimates via logit the propensity score using specified linear and quadratic terms.
`propensity_s`	Estimates via logit the propensity score using the covariate selection algorithm of Imbens and Rubin (2015).
`trim`	Trims data based on propensity score using the threshold specified by the attribute `cutoff`.
`trim_s`	Trims data based on propensity score using the cutoff selected by the procedure of Crump, Hotz, Imbens, and Mitnik (2008).
`stratify`	Stratifies the sample based on propensity score as specified by the attribute `blocks`.
`stratify_s`	Stratifies the sample based on propensity score using the bin selection procedure suggested by Imbens and Rubin (2015).
`blocking`	Estimates average treatment effects using regression within blocks.
`matching`	Estimates average treatment effects using matching with replacement.
`weighting`	Estimates average treatment effects using the Horvitz-Thompson weighting estimator modified to incorporate covariates.
`ols`	Estimates average treatment effects using least squares.

Table 2: Methods of the class CausalModel. Invoke `help` on any of the above methods for more detailed documentation.

```
>>> print causal.pscore
Estimated Parameters of Propensity Score
                    Coef.       S.e.          z      P>|z|      [95% Conf. int.]
--------------------------------------------------------------------------------
     Intercept     -0.018      0.136     -0.133      0.447     -0.284      0.248
            X0      1.785      0.192      9.305      0.000      1.409      2.161
            X2      1.789      0.195      9.177      0.000      1.407      2.171
            X1      1.765      0.203      8.678      0.000      1.366      2.163
```

Trimming based on propensity score.

```
>>> causal.trim_s()
>>> causal.N, causal.N_c, causal.N_t
(281, 143, 138)
>>> print causal.covariates
```

```
Covariates Summary
                       Controls (N_c=143)         Treated (N_t=138)
      Covariate          Mean          S.d.          Mean          S.d.     Nor-diff
--------------------------------------------------------------------------------
             X0        -0.171         0.770         0.163         0.893        0.401
             X1        -0.046         0.848         0.066         0.765        0.139
             X2        -0.178         0.780         0.125         0.802        0.383
```

Stratifying sample based on propensity score.

```
>>> causal.stratify_s()
>>> print causal.strata
Stratification Summary
              Propensity score                           Ave. p-score    Within
   Stratum       Min.       Max.        N_c        N_t  Controls   Treated      Est.
--------------------------------------------------------------------------------
         1      0.085      0.513        104         37     0.230     0.267     2.274
         2      0.518      0.631         18         17     0.570     0.589     2.996
         3      0.632      0.753          7         28     0.678     0.699     3.447
         4      0.757      0.915         14         56     0.835     0.842     3.367
```

Computing the blocking estimator.

```
>>> causal.blocking()
>>> print causal.est
Treatment Effect Estimates
Blocking
                     Est.       S.e.          z      P>|z|      [95% Conf. int.]
--------------------------------------------------------------------------------
           ATE      2.782      0.127     21.941      0.000      2.534      3.031
           ATT      3.045      0.152     20.046      0.000      2.747      3.342
           ATC      2.530      0.134     18.925      0.000      2.268      2.792
```