Assignment 2: Diabetes Interventions

NWI-IMC012: Bayesian Networks and Causal Inference

Authors:

Daan Kersten
S1053750

Andrew Schroeder
S1111686

Sven Meijboom
S1054374

Radboud University, Nijmegen

December 20, 2024

Contents

1 Introduction

2 Methods

2.1 Structure Learning

In order to automate the process of creating a DAG from the dataset, the PC-algorithm was used in Rstudio. The PC-algorithm is a straightforward algorithm which makes use of black- and white-lists. The white-list consist of all edges between nodes which certainly should end up in the final DAG. The black-list consists of all edges between nodes that certainly should not end up in the final DAG. Based on the black- and white-lists and the data, the PC-algorithm learns a DAG.

To ensure that the PC-algorithm learns the DAG solely from the data and not from pre-existing knowledge, the minimal number of edges were added to the black- and white-lists. The main focal relationship for this paper is between high blood pressure and diabetes, it was therefore chosen to only add this edge to the white list. For the black-list the choice was made to add all edges coming into the variables Sex and Age. The reason for this is that nothing can influence a person's Sex or Age during their lifetime. Apart from these edges, none were further added to the lists such that all further relationships where learned solely from the dataset.

Now that the structure learning algorithm, together with the black- and white-lists have been defined, the next step is to run the algorithm on the dataset. The initial idea was to run the PC-algorithm on all 253000 samples in the dataset but there are some major flaws to this approach. The first one is the time it takes to run. An initial attempt was made to run the algorithm on the full dataset, but this attempt was aborted after 15 hours where there was still no outcome. The second flaw is the potential for overfitting. When running the algorithm on a very large dataset, even small insignificant relations in the network become significant enough for the edges to be included in the final DAG. To prevent these flaws, the choice was made to only use a small subset of 5000 samples.

2.2 Propensity Score Matching

The first step involved performing a logistic regression to estimate the unadjusted relationship between `HighBP` and `Diabetes_binary`. This provided a baseline estimate of the effect without accounting for potential confounders. To evaluate covariate imbalance between the treatment (`HighBP` $= 1$) and control groups (`HighBP` $= 0$), a t-test was conducted, and standardized differences for each covariate were calculated. A forest plot was created to systematically visualize these imbalances.

Next, propensity score matching was applied to reduce confounding. Covariates for estimating the propensity score were selected based on the DAG of assignment 1, where we selected the parent nodes of `HighBP`. A logistic regression model was used to calculate propensity scores, which represent the probability of receiving 'treatment' (`HighBP`) given the selected covariates. The distribution of propensity scores across treatment groups was visualized using boxplots, and inverse probability weights were calculated to conduct a weighted regression analysis. To further refine the dataset, a caliper matching approach was implemented, pairing treatment and control observations within a specified distance of 0.02 times the standard deviation of the propensity scores (I found online that this was a rule of thumb. Is it an idea to test different 'caliper'-values and look for the least sum of standardized difference?). Unmatched observations were removed, and the regression analysis was repeated on the matched dataset. The new dataset's covariate balance was reassessed using standardized differences and visualized with a forest plot.

For the DAG that is created by the structure learning algorithm, we will redo the steps we have done in the previous paragraph but change the covariates to the parents of `HighBP` of the DAG that is created by the structure learning algorithm.

2.3 Covariate Adjustment

3 Results

3.1 Structure Learning

As mentioned in the methods section, the DAG was created with the PC-algorithm using a subset of 5000 samples from the dataset. The following DAG consisting of 68 edges is what came out as a result. Below this DAG, the DAG from assignment 1 is placed for comparison.

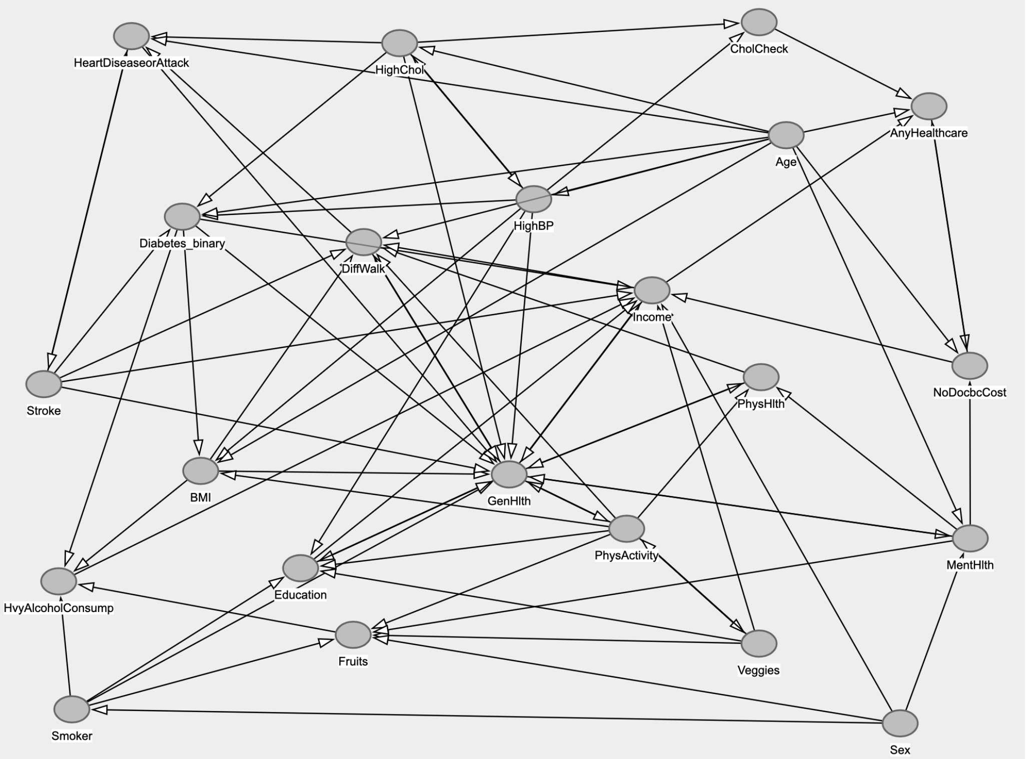


Figure 1: DAG created by PC-algorithm

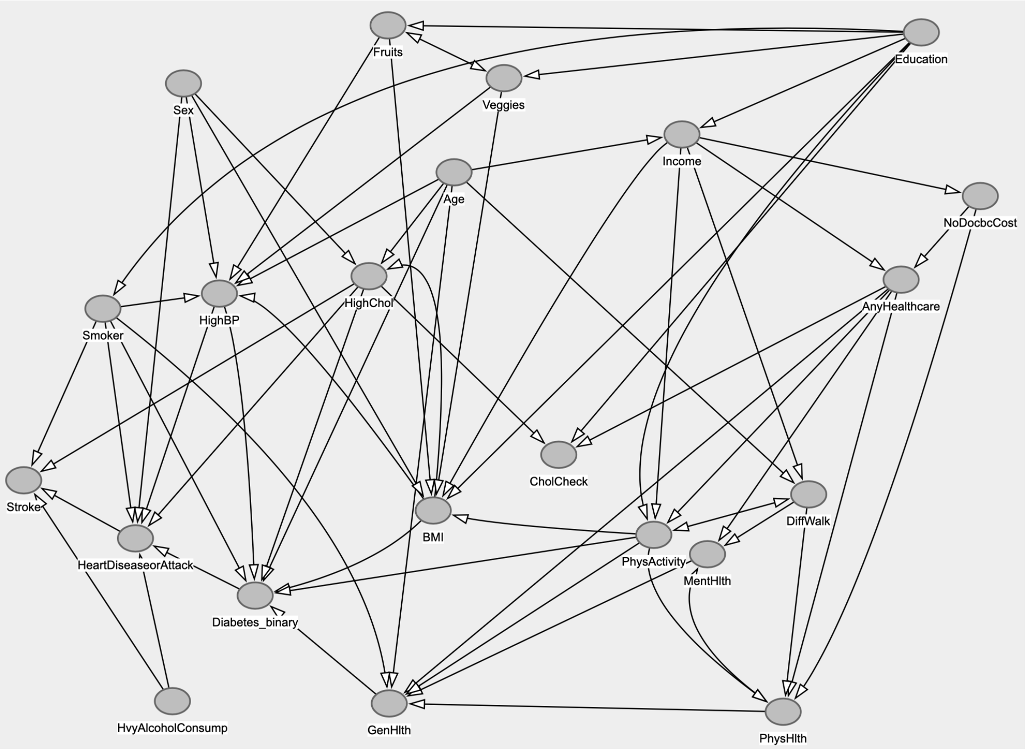
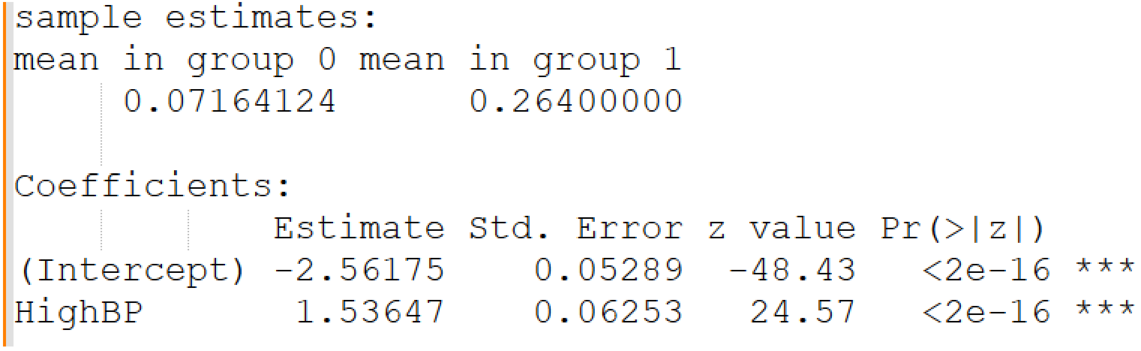


Figure 2: Handmade DAG

3.2 Propensity Score Matching

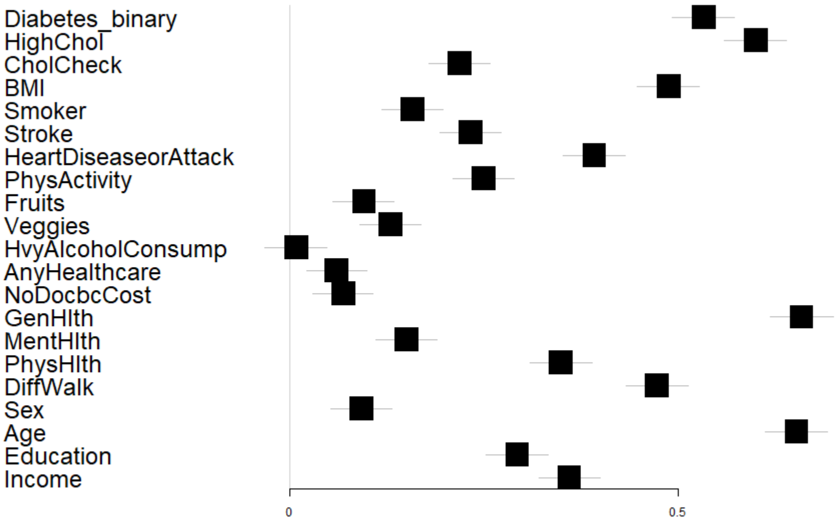
Quick overview of the results we've got until now:

Before propensity matching:

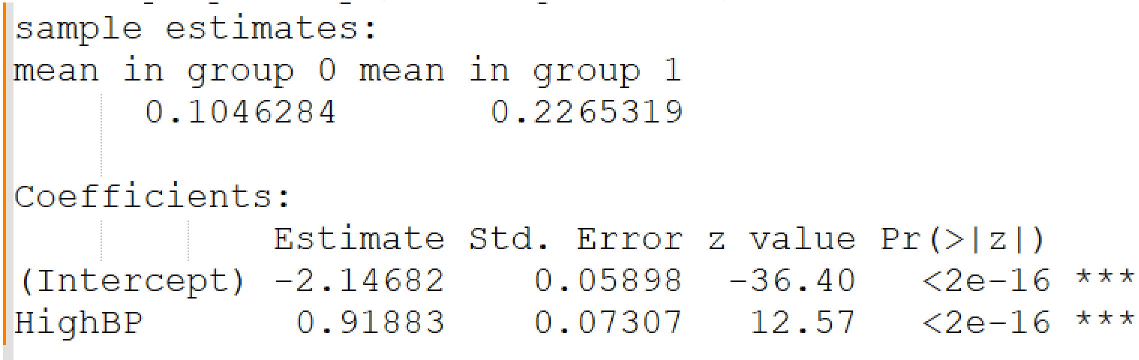


```
sample estimates:
mean in group 0 mean in group 1
     0.07164124      0.26400000

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.56175    0.05289  -48.43   <2e-16 ***
HighBP       1.53647    0.06253   24.57   <2e-16 ***
```

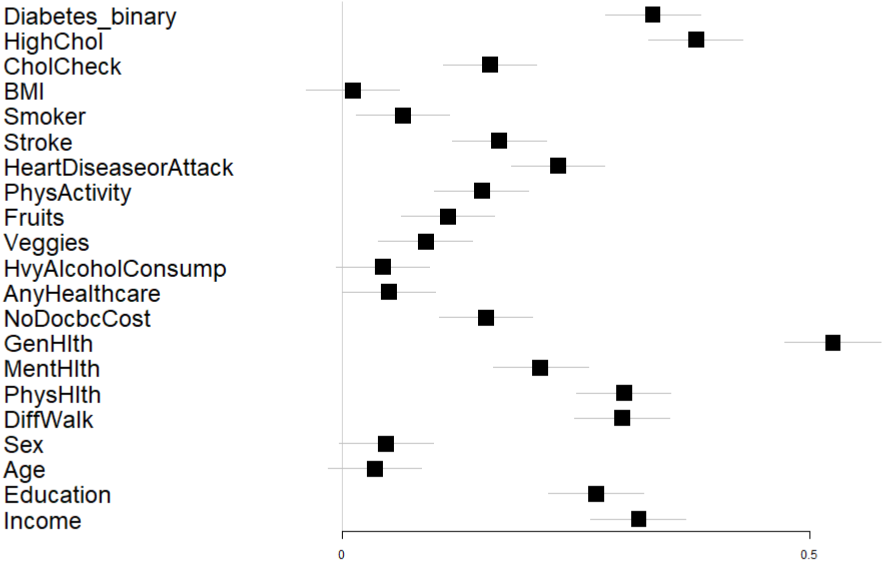


After propensity matching (on DAG of assignment 1):



```
sample estimates:
mean in group 0 mean in group 1
      0.1046284       0.2265319

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.14682    0.05898  -36.40   <2e-16 ***
HighBP       0.91883    0.07307   12.57   <2e-16 ***
```



This does not seem like good results because the difference in mean between group 0 and group 1 is still quite high after propensity matching. Besides that, the forest plot of the standardized difference did show some improvements for some variables, but not as much as we expected. We're still figuring out how we can improve this.

3.3 Covariate Adjustment

4 Discussion

5 Conclusion