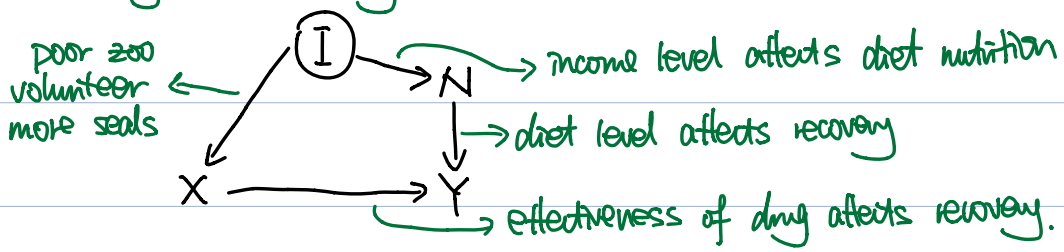
hw4.

1.

a. → X = drug, Y = recovery, I = income level, N = diet.

⇒ 

b.

→ backdoor criteria = block all confounding passway in X & Y.

→ ordered pair (X, Y) & backdoor variable set Z.

→ no node in Z is decendent of X, then Z blocks confounder.

⇒ minimal set Z = variable I

⇒ I statisfied the backdoor criteria because it block all the confounders in this casual model (income affects diet and the seals receiving the drug, which is the only two direct parent of our recovery rate Y.

c. → adjustment formula for effects of the drugs on recovery.

⇒ treatment effects = $\mathbb{E}[Y \mid do(X := 1)]$

$= \mathbb{P}(Y = y \mid do(X := \text{received}))$

$$= \sum_I \mathbb{P}(Y = \text{recovered} \mid X = \text{received}, I = i) * \mathbb{P}(I = i)$$

2.

a. → prove $\mathbb{E}[Y \mid do(X:=0)] = \mathbb{E}\left[\frac{Y(1-X)}{1-e(Z)}\right]$

$$\Rightarrow \mathbb{E}[Y \mid do(X:=0)] = \sum_y y * \mathbb{P}(Y=y \mid do(X:=0))$$

$$= \sum_y y * \left(\sum_z \mathbb{P}(Y=y \mid X=0, Z=z) * \mathbb{P}(Z=z)\right)$$

$$= \sum_y y * \left(\sum_z \frac{\mathbb{P}(Y=y \mid X=0, Z=z) * \mathbb{P}(Z=z) * \mathbb{P}(X=0 \mid Z=z)}{\mathbb{P}(X=0 \mid Z=z)}\right)$$

$$= \sum_y y * \left(\sum_z \frac{\mathbb{P}(Y=y, X=0, Z=z)}{\mathbb{P}(X=0 \mid Z=z}\right)$$

$$= \sum_y y * \left(\sum_{z, x \in (0,1)} \frac{\mathbb{1}\{X=0\}\mathbb{P}(Y=y, X=x, Z=z)}{\mathbb{P}(X=0 \mid Z=z)}\right)$$

$$= \sum_{y,z,x \in (0,1)} \frac{y * \mathbb{1}\{X=0\} * \mathbb{P}(Y=y, X=x, Z=z)}{\mathbb{P}(X=0 \mid Z=z)}$$

$$\Rightarrow \mathbb{P}(X=0 \mid Z=z) = 1 - \mathbb{P}(X=1 \mid Z=z) = 1 - e(z)$$

$$\Rightarrow \mathbb{1}\{X=0\} \text{ for each } X_i = 1 - X_i$$

$$\Rightarrow \text{thus, } \mathbb{E}[Y \mid do(X:=0)] = \mathbb{E}\left[\frac{Y(1-X)}{1-e(Z)}\right].$$

b.

$$\Rightarrow \text{IPSWE } \mathbb{E}[Y \mid do(X:=0)] = \frac{1}{n}\sum_{i=1}^{n} \frac{y_i(1-x_i)}{1-\hat{e}(z_i)}.$$

c.

$$\Rightarrow \text{the treatment effect } \mathbb{E}[Y \mid do(X:=1)] - \mathbb{E}[Y \mid do(X:=0)].$$

$$= \frac{1}{n}\sum_{i=1}^{n} \frac{x_i y_i}{\hat{e}(z_i)} - \frac{1}{n}\sum_{i=1}^{n} \frac{y_i (1 - x_i)}{1 - \hat{e}(z_i)}$$

3. → IHDP & treated deficient infants with trained child-care.

→ estimate casual effect on outcome = cognitive test score.

→ may have confounder. btw treatment & outcome.

c.

⇒ treatment effect for IHDP dataset ≈ 3.6957.

4.

a. $\rightarrow T_i(t) =$ #times i^{th} arm pulled (up to $t=n$)

$\rightarrow T_i(n) = \sum_{t=1}^{n} \mathbb{1}\{A_t = i\}$

$\rightarrow$ prove $R(n) = \sum_{i=1}^{k} \Delta_i \mathbb{E}[T_i(n)]$.

$$\Rightarrow n\mu^* - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right]$$

$$= \mathbb{E}\left[\sum_{t=1}^{n} (\mu^* - X_t)\right]$$

$$= \mathbb{E}\left[\sum_{i=1}^{k}\sum_{t=1}^{n} \mathbb{1}\{A_t = i\}(\mu^* - Y_{i,t})\right] \leftarrow A_t \text{ independent of } Y_{i,t}$$

$$= \sum_{i=1}^{k}\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\{A_t = i\}(\mu^* - Y_{i,t})\right]$$

$$= \sum_{i=1}^{k}\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\{A_t = i\}\right] * \mathbb{E}\left[\mu^* - Y_{i,t}\right]$$

$$= \sum_{i=1}^{k}\sum_{t=1}^{n} \left[\mathbb{E}\left[\mathbb{1}\{A_t = i\}\right] * \mathbb{E}(\mu^*) - \mathbb{E}\left[\mathbb{1}\{A_t = i\}\right]\mathbb{E}\left[Y_{i,t}\right]\right]$$

$$= \sum_{i=1}^{k}\left[\mathbb{E}(\mu^*) * \mathbb{E}[T_i(n)] - \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\{A_t = i\}\right]\mathbb{E}\left[Y_{i,t}\right]\right]$$

$$= \sum_{i=1}^{k}\left[\mathbb{E}(\mu^*) * \mathbb{E}[T_i(n)] - \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{1}\{A_t = i\}(Y_{i,t})\right]\right]$$

$$= \sum_{i=1}^{k}\left[\mathbb{E}(\mu^*) * \mathbb{E}[T_i(n)] - \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}\{A_t = i\}(Y_{i,t})\right]\right]$$

← definition from last semes

$\hat{\mu}_{i,T_i(t)} = \frac{1}{T_i(t)} \sum_{s=1}^{t} X_s \mathbb{1}\{A_s = a$

$$= \sum_{i=1}^{k}\left[\mathbb{E}(\mu^*) * \mathbb{E}[T_i(n)] - \mathbb{E}\left[\hat{\mu}_{i,T_i(n)} * T_i(n)\right]\right]$$

$$= \sum_{i=1}^{k}\left[\mathbb{E}(\mu^*) * \mathbb{E}[T_i(n)] - \mathbb{E}\left[\hat{\mu}_{i,T_i(n)}\right] * T_i(n)\right]$$

$$= \sum_{i=1}^{k}\left[\mathbb{E}(T_i(n)) * \left[\mathbb{E}(\mu^*) - \mathbb{E}(\hat{\mu}_i)\right]\right]$$

$$= \sum_{i=1}^{k} \left[\mathbb{E}(T_i(n)) * (\mu^* - \mu_i) \right]$$

$$= \sum_{i=1}^{k} \Delta_i \, \mathbb{E}[T_i(n)].$$

$\Rightarrow$ thus, $R(n) = \sum_{i=1}^{k} \Delta_i \, \mathbb{E}[T_i(n)]$

b. $\rightarrow$ if $n > ck$

$\rightarrow \mathbb{E}[T_i(n)] = c + (n - kc)\, \mathbb{P}\left(\hat{\mu}_i > \max_{j=1,\dots,k, j \neq 1} \hat{\mu}_j\right)$.

$\rightarrow n > ck$, every arm is pulled c times (at least)

$\rightarrow$ afterward, arm only pull if $\hat{\mu}_i = \operatorname{argmax}_i \hat{\mu}_i$

$\Rightarrow$ when $n > ck$, every A_i will deterministically be pulled c times.

$\Rightarrow T_i(n) = c + \mathbb{1}\{\hat{\mu}_i = \mu^*\} * (n - kc)$

$\Rightarrow \mathbb{E}(T_i(n)) = \mathbb{E}\left[c + \mathbb{1}\{\hat{\mu}_i = \mu^*\} * (n - kc) \right]$

$$= c + (n - kc) * \mathbb{E}\left[\mathbb{1}\{\hat{\mu}_i = \mu^*\}\right]$$

$\Rightarrow$ definition of $\mu^* = \max_{i \in \{1,\dots,k\}} \mu_i$

$\Rightarrow$ so $\mathbb{1}\{\hat{\mu}_i = \mu^*\} = \mathbb{1}\{\hat{\mu}_i >$ all the other arms mean$\}$

$\Rightarrow$ so $\mathbb{E}\left[\mathbb{1}\{\hat{\mu}_i = \mu^*\}\right] = \mathbb{P}\{\hat{\mu}_i >$ all the other arms mean$\}$

$$= \mathbb{P}\left(\hat{\mu}_i > \max_{j=1,\dots,k, j \neq 1} \hat{\mu}_j\right)$$

$\Rightarrow$ thus, $\mathbb{E}[T_i(n)] = c + (n - kc)\, \mathbb{P}\left(\hat{\mu}_i > \max_{j=1,\dots,k, j \neq i} \hat{\mu}_j\right)$.

c. $\rightarrow$ optimal arm (μ^*) = the first arm μ_1

→ any sub-optimal arm i

→ $\mathbb{P}(\hat{\mu}_i > \max_{j=1,\dots,k, j\neq i} \hat{\mu}_j) \leq \mathbb{P}(\hat{\mu}_i > \hat{\mu}_1)$.

→ one event subset of another.

⇒ $\mathbb{P}(\hat{\mu}_i > \max_{j=1,\dots,k, j\neq i} \hat{\mu}_j)$ & μ_i is the sub-optimal arm

⇒ since we know μ_1 is the μ^*

⇒ $\mathbb{P}(\hat{\mu}_i > \max_{j=1,\dots,k, j=i} \hat{\mu}_j)$ where $i \in \{2, 3, \dots, k\}$.

⇒ however for $\mathbb{P}(\hat{\mu}_i > \hat{\mu}_1)$, since $\mu_1 = \mu^*$

here $i \in \{1, 2, \dots, k\}$.

⇒ which includes arm 1

⇒ so if we let $\hat{\mu}_i > \max_{j=1,\dots,k, j\neq i} \hat{\mu}_j$ be event E_1

and $\hat{\mu}_i > \hat{\mu}_1$ be event E_2

⇒ $E_1 \subseteq E_2$

⇒ $\mathbb{P}(E_2) > \mathbb{P}(E_1)$

⇒ thus, $\mathbb{P}(\hat{\mu}_i > \max_{j=1,\dots,k; j\neq i} \hat{\mu}_j) \leq \mathbb{P}(\hat{\mu}_i > \hat{\mu}_1)$.

d. → $\mathbb{E}[T_i(n)] \leq c + (n - kc)\mathbb{P}(\hat{\mu}_i > \hat{\mu}_1)$.

→ prove $\mathbb{P}(\hat{\mu}_i > \hat{\mu}_1)$

$= \mathbb{P}\left(\frac{1}{c}\sum_{s=1}^{c} Y_{i,K(s-1)+i} > \frac{1}{c}\sum_{s=1}^{c} Y_{1,K(s-1)+i}\right) \leq \exp\left(-\frac{c\Delta_i^2}{2(b-a)^2}\right)$.

→ hoeffding bounds: for $z_1, \ldots, z_m$ & $z_j \in (l_j, u_j)$ for $j = 1, \ldots, m$.

$$P\left(\sum_{j=1}^{m} z_j - E\left[\sum_{j=1}^{m} z_j\right] > t\right) \leq \exp\left(-\frac{2t^2}{\sum_{j=1}^{m} (u_j - l_j)^2}\right).$$

→ $Y_{i,K(s-1)+i}$ = reward for i^{th} arm that is pulled s^{th} time

→ $Y_{i,K(s-1)+i} - Y_{1,K(s-1)+i} \in (b-a, a-b)$

→ mean $\mu_i - \mu_1$

$$\Rightarrow P(\hat{\mu}_i > \hat{\mu}_1) = P\left(\frac{1}{C}\sum_{s=1}^{C} Y_{i,K(s-1)+i} > \frac{1}{C}\sum_{s=1}^{C} Y_{1,K(s-1)+i}\right)$$

$$= P\left(\frac{1}{C}\sum_{s=1}^{C} Y_{i,K(s-1)+i} - \frac{1}{C}\sum_{s=1}^{C} Y_{1,K(s-1)+i} > 0\right)$$

$$= P\left(\sum_{s=1}^{C} Y_{i,K(s-1)+i} - \sum_{s=1}^{C} Y_{1,K(s-1)+i} > 0\right).$$

$$= P\left(\sum_{s=1}^{C} \left[Y_{i,K(s-1)+i} - Y_{1,K(s-1)+i}\right] > 0\right)$$

⇒ let $Y_{i,K(s-1)+i} - Y_{1,K(s-1)+i}$ = variable z

⇒ $b-a < z < a-b$ & $z \sim$ mean $(\mu_i - \mu_1)$

we have total of C^{th} z.

⇒ our original equation become $P\left(\sum_{s=1}^{C} z_s > 0\right)$

⇒ using hoeffding bound.

$$\Rightarrow P\left(\sum_{j=1}^{C} z_j - E\left[\sum_{j=1}^{C} z_j\right] > t\right) \leq \exp\left(-\frac{2t^2}{\sum_{j=1}^{C} (u_j - l_j)^2}\right)$$

$$\leq \exp\left(-\frac{2t^2}{\sum_{j=1}^{C} [(b-a-a+b)]^2}\right).$$

$$\Rightarrow E\left[\sum z_j\right] - E\left[\sum Y_{i,K(s-1)+i} - Y_{1,K(s-1)+i}\right]$$

$$= \mathbb{E}[c\hat{\mu}_i - c\hat{\mu}_1]$$

$$= c * \mathbb{E}[\hat{\mu}_i - \hat{\mu}_1].$$

$\Rightarrow$ since we need $\mathbb{P}(\sum_{s=1}^{c} z_s > 0)$

$\Rightarrow \mathbb{P}(\sum_{j=1}^{c} z_j - \mathbb{E}[\sum_{j=1}^{c} z_j] > \mathbb{E}[\sum_{j=1}^{c} z_j])$.

$\Rightarrow \mathbb{P}(\sum_{j=1}^{c} z_j - \mathbb{E}[\sum_{j=1}^{c} z_j] > c * \mathbb{E}[\hat{\mu}_i - \hat{\mu}_1])$

$\leq \exp\left(-\frac{2[c * \mathbb{E}[\hat{\mu}_i - \hat{\mu}_1]]^2}{c * [2(b-a)]^2}\right)$ ← μ^* is larger than any μ_i

$\leq \exp\left(-\frac{2[c * \mathbb{E}[\mu^* - \mu_i]]^2}{c * [2(b-a)]^2}\right)$ ← prove in 4c without lose of generality.

$\leq \exp\left(-\frac{2c^2\Delta_i^2}{4c(b-a)^2}\right)$

$\leq \exp\left(-\frac{c\Delta_i^2}{2(b-a)^2}\right)$.

$\Rightarrow \mathbb{P}(\sum_{j=1}^{c} z_j)$ = our original equation

$\Rightarrow$ thus, $\mathbb{P}(\hat{\mu}_i > \hat{\mu}_1) = \mathbb{P}\left(\frac{1}{c}\sum_{s=1}^{c} Y_{i,K(s-1)+i} > \frac{1}{c}\sum_{s=1}^{c} Y_{i,K(s-1)+i}\right)$

$\leq \exp\left(-\frac{c\Delta_i^2}{2(b-a)^2}\right)$.

e.

$\Rightarrow \mathbb{E}[T_i(n)] \leq c + n * \exp\left(\frac{c\Delta_i^2}{2(b-a)^2}\right)$.

$\Rightarrow$ goal = value of c for main hoeffding bound $\leq \frac{1}{n}$

$\Rightarrow$ set $\exp(\frac{c\Delta_i^2}{2(b-a)^2}) = \frac{1}{n}$ & solve for c.

$\Rightarrow$ from part A: $R(n) = \sum_{i=1}^{k} \Delta_i \, \mathbb{E}[T_i(n)]$

$\mathbb{E}[T_i(n)]$ → substitute in upper bound.

$$\leq \sum_{i=1}^{k} \Delta_i \left(C^* + n \right)$$

$$\leq \sum_{i=1}^{k} \Delta_i \left(C^* + n * \frac{1}{n}\right).$$

Untitled

April 7, 2020

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats
%matplotlib inline
```

[2]:
```
dataframe = pd.read_csv("ihdp.csv", header = None)
dataframe
```

[2]:
```
            0                  1                    2   \
0    treatment            outcome             feature0
1          0.0    2.35891557538093   0.23012360624317396
2          1.0    5.86106115988978   0.12204291849488699
3          0.0   0.9771483357774841  -0.28866369494860494
4          0.0    4.36721066102877   -0.0725023194520303
..         ...                 ...                  ...
743        0.0   3.6496843910405294  -0.00765390680305797
744        0.0    3.02022901834076    1.3973950339246801
745        0.0    5.679426224076151   0.31658815644180394
746        1.0    7.02702037229374   -2.5151258625633197
747        1.0   7.6900243405027195     0.921840007832212

                     3                  4                    5   \
0               feature1           feature2             feature3
1       0.596582186338594  -0.360897988611247    0.161702527138546
2    -0.20294594178963002  -0.360897988611247     1.20301104241867
3     0.19681812227448198   0.0114649913509099  -0.8796059881415771
4       0.596582186338594  -0.7332609685734041     2.24431955769879
..                    ...                 ...                  ...
743  -0.20294594178963002   0.7561909512752241    0.161702527138546
744     1.39611031446682    -1.47798692849772    0.161702527138546
745   0.19681812227448198  -0.7332609685734041     2.24431955769879
746   -3.4010584543025297     2.99036883104816    0.161702527138546
747     1.39611031446682   -0.7332609685734041    0.161702527138546

                     6                  7         8         9    ...  \
0               feature4           feature5  feature6  feature7  ...
```

```
1    0.24605163988208498   -0.526555605465598       0.0       0.0  ...
2     0.0585000327474753    0.632753557052133       1.0       0.0  ...
3      0.308568842260288   -1.1890179840471597       0.0       0.0  ...
4       1.80898169933717    0.632753557052133       1.0       0.0  ...
..                   ...                  ...       ...       ...  ...
743  0.24605163988208498   -0.0297088215294272       1.0       1.0  ...
744   0.0585000327474753    0.9639847463429141       1.0       0.0  ...
745    0.683672056529507    0.632753557052133       0.0       0.0  ...
746  -1.5669472290858102     1.46083153027908       1.0       0.0  ...
747  0.24605163988208498    0.7983691516975241       0.0       1.0  ...

             17         18         19         20         21         22  \
0    feature15  feature16  feature17  feature18  feature19  feature20
1          1.0        0.0        1.0        0.0        0.0        1.0
2          1.0        1.0        1.0        0.0        0.0        0.0
3          1.0        0.0        1.0        1.0        0.0        0.0
4          1.0        0.0        1.0        1.0        0.0        0.0
..         ...        ...        ...        ...        ...        ...
743        1.0        1.0        1.0        0.0        0.0        0.0
744        1.0        1.0        1.0        0.0        1.0        0.0
745        0.0        0.0        1.0        0.0        0.0        0.0
746        1.0        1.0        1.0        0.0        0.0        0.0
747        1.0        1.0        1.0        0.0        0.0        0.0

             23         24         25         26
0    feature21  feature22  feature23  feature24
1          0.0        0.0        0.0        0.0
2          0.0        0.0        1.0        0.0
3          0.0        0.0        0.0        0.0
4          0.0        0.0        0.0        0.0
..         ...        ...        ...        ...
743        0.0        0.0        0.0        0.0
744        0.0        0.0        0.0        0.0
745        0.0        0.0        0.0        1.0
746        0.0        0.0        0.0        1.0
747        0.0        0.0        0.0        0.0

[748 rows x 27 columns]
```

[3]:
```
X_var = np.asarray(dataframe[0][1:])
Y_var = np.asarray(dataframe[1][1:])
Z_confounders = []
for i in np.arange(1,len(dataframe)):
    single = np.asarray(dataframe.loc[i][2:])
    Z_confounders.extend([single])
```

[4]:
```
from sklearn.linear_model import LogisticRegression as LR
lr = LR(penalty="none", max_iter=200,solver ='lbfgs',random_state=0)
```

[5]:
```
lr.fit(Z_confounders,X_var)
```

[5]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=200,
                   multi_class='warn', n_jobs=None, penalty='none',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

[6]:
```
attemp = lr.predict_proba(Z_confounders)
lr.classes_
```

[6]:
```
array(['0.0', '1.0'], dtype=object)
```

[7]:
```
def estimate_treatment_effects(trained_model,X,Y,Z):
    e_zi = trained_model.predict_proba(Z)
    first_total = 0
    second_total = 0
    for i in np.arange(0,len(e_zi)):
        first = (float(X[i]) * float(Y[i])) / e_zi[i][1]
        first_total = first_total + first
        second = ((1 - float(X[i])) * float(Y[i])) / (1 - e_zi[i][1])
        second_total = second_total + second
    score = first_total/len(X) - second_total/len(X)
    return score
```

[8]:
```
estimate_treatment_effects(lr,X_var,Y_var,Z_confounders)
```

[8]:
```
3.6956988430209456
```

3c. Since the treatment effects score is approximate 3.6956988, which is positive, we can obserive that the treatement cause conitive test scores to increase. Thus, according to the estimate, the treatment did have a beneficial causal effect on the outcome.

[9]:
```
naive_first = 0
first_track = 0
naive_second = 0
second_track = 0
for i in np.arange(0,len(X_var)):
    first_n = float(Y_var[i]) * float(X_var[i])
    first_track = first_track + float(X_var[i])
    naive_first = naive_first + first_n
    second_n =  float(Y_var[i]) * (1 - float(X_var[i]))
    second_track = second_track + (1 - float(X_var[i]))
    naive_second = naive_second + second_n
naive_score = naive_first/first_track - naive_second/second_track
naive_score
```

[9]:
```
4.021121012430834
```

This naive treatment effect estimator is higher than the one we compute is the previous part. This is because during this naive estimator, we did not take into account of the confounders effects. For instance, baby who received treatment are more likely premature baby, who might naturally just get more attention from the family(to take care of), and thus have earlier brain de-

velopment,which cause thier test score to be higher. The naive treatment did not take into consider the confounder effects, and thus might wrongly amplify the treatment effect. Those two score will be the same, under the condition that there's no confounders between the treatment and the outcome. Which mean, none of the feature will influence whether the baby receive the treatment or the test score the achieve. If there's no other path between X and Y, those two treatment effect estimator would be the same.

[ ]: