CS7290 Causal Modeling in Machine Learning: Homework 3

Submission guidelines

Use a Jupyter notebook and/or R Markdown file to combine code and text answers. Compile your solution to a static PDF document(s). Submit both the compiled PDF and source files. If you use Google Collab, send the link as well as downloaded PDF and source files.

Background/Reference

Causal Inference in Statistics - A Primer, Chapter 3, Chapter 1

Question 1: Section 3.1, 3.2, 3.3

Question 2: Section 3.3, 3.4

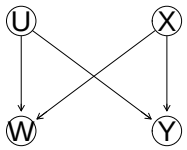
Question 3: Section 3.6

Question 4: Section 1.5

Question 1: Valid adjustment sets (27 points)

1.1 (9 points)

The following DAG represents a causal model of user behavior in an app.



U represents the user specific preferences. X represents the introduction of a feature designed to make users make certain in-app purchases, Y was whether or not the user made the purchase, W represents app usage after the feature is introduced.

1.1.a

You are interested in estimating the causal effect of X on Y. List all the valid adjustment sets. A valid adjustment set is the set of variables that if you adjust, you will get the unbiased results. (For a formal definition of valid adjustment set, see "ELements of Causal Inference", Definition 6.38, Proposition 6.41) (3 points)

1.1.b

What would happen if you adjusted for W? (2 points)

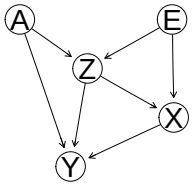
1.1.c

Suppose you want to assess the effect of X on Y for users who have a high amount of app usage. Fill in the blanks on the right-hand-side for the adjustment formula of interest. (4 points)

$$P(Y = y|do(X = x), W = high) = \sum_{?} P(Y = y|?)P(?|?)$$

1.2 (6 points)

Consider the following DAG.



You are interest in estimating the causal effect of X on Y.

1.2.a

Is the set containing only Z a valid adjustment set? Why or why not? (2 points)

1.2.b

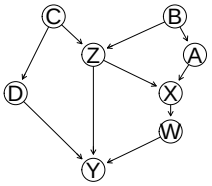
List all of the adjustment sets that blocks all the back doors(there are three) and write the adjustment formula for each adjustment set. (3 points)

1.2.c

Suppose that E and A are both observable, but observing E costs \$10 per data point and observing A costs \$5 per data point. Which conditioning set do you go with? (1 point)

1.3 (12 points)

Consider the following DAG:



1.3.a

List all of the sets of variables that satisfy the backdoor criterion to determine the causal effect of X on Y. (3 points)

1.3.b

List all of the minimal sets of variables that satisfy the backdoor criterion to determine the causal effect of X on Y (a minimal valid adjustment set here means if you removed any one of the variables from the set, it would no longer be a valid adjustment set). (3 points)

1.3.c

List all the minimal sets of variables that need to be measured in order to identify the effect of D on Y. (3 points)

1.3.d

Now suppose we want to know the causal effect of intervening on 2 variables. List all the minimal sets of variables that need to be measured in order to identify the effect of set {D, W} on Y, i.e., $P(Y = y|do(D = d), do(W = w))$. (3 points)

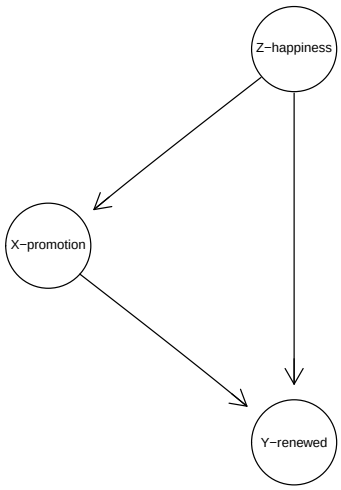
Question 2: Covariate adjustment (20 points)

2.1 (10 points)

You are a data scientist at a prominent tech company with paid subscription entertainment media streaming service. You come across some data on a promotional campaign. The campaign targeted 70K subscibers users who were coming to a subscription renewal time and were at high risk of not renewing. They were targeted with two types of promotions, call them promotion 0 and promotion 1. You do some digging and find out the promotions the users were offered dependended on how happy the users were (quantified from user behavior and customer service interactions). The following table shows the percentage of users renewing for happy users and unhappy users after receiving promotion 0 and promotion 1.

	Overall	Unhappy	Happy
Promotion 0	77.9% (27272/35000)	93.2% (8173/8769)	73.3% (19228/26231)
Promotion 1	82.6% (28902/35000)	86.9% (23339 / 26872)	68.7% (5582/8128)

You assume the following causal DAG:



You are interested in the average causal effect $P(Y = 1|\text{do}(X = 0)) - P(Y = 1|\text{do}(X = 1))$, where Y=1 represents renewed, X=0 represents promotion 0 and X=1 represents promotion 1

2.1.a

Build the model with Pyro using the values in the table. Use `pyro.condition` to calculate the causal effect of promotion on renew by adjusting for happiness. (5 points)

2.1.b

Verify your result of Q2.1.a using `pyro.do`. (5 points)

2.2 (10 points)

You are a data scientist investigating the effects of social media use on purchasing a product. You assume the dag shown below. User demographic information here is unobserved. One of the team members argues

that social media usage does not drive purchase based on Table 1. Only 15% social media user made the purchase, while 90.25% non social media users made the purchase. Moreover, within each group, no-adblock and adblock, social media users show a much lower rate of purchase than non social media users. However, another team member argues that social media usage increases purchases. When we look at each group, social media user and non social media user as show in Table 2 (Table 1 and Table 2 both represent the same dataset), advertisement increases purchases in both groups. Among social media users, purchases increases from 10% to 15% for people who have seen advertisement. Among non social media users, purchases increases from 90% to 95% for people who have seen advertisement. Which view is right? To asnwer this question, you want to calculate the average causal effect of social media on product purchase $P(Y = 1|\text{do}(X = 0)) - P(Y = 1|\text{do}(X = 1))$, where Y=1 represents purchase, X=1 represents social media user, and X=0 represents non social media user.

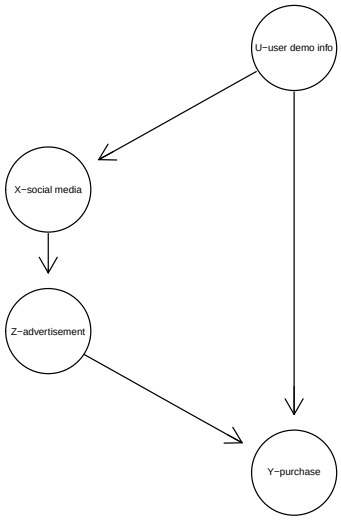


Table 1:

	advertisement (50%)		no ad (50%)		all subjects(800)	
	social	no social	social	no social	sociale	no social
Total	380	20	20	380	400	400
No Purchase	323 (85%)	1 (5%)	18 (90%)	38 (10%)	341 (85.25%)	39 (9.75%)
Purchase	57 (15%)	19(95%)	2 (10%)	342 (90%)	59 (14.75%)	361 (90.25%)

Table 2:

	social (50%)		no social (50%)		all subjects(800)	
	advertisement	no ad	advertisement	no ad	advertisement	no ad
Total	380	20	20	380	400	400
No Purchase	323 (85%)	18 (90%)	1 (5%)	38 (10%)	324 (81%)	56 (14%)
Purchase	57 (15%)	2 (10%)	19 (95%)	342 (90%)	76 (19%)	344 (86%)

2.2.a

Suppose you don't have any data on user demographic information, so U is unobserved. Use `pyro.condition` to calculate the causal effect of social media on product purchase using front-door adjustment.(5 points)

2.2.b

Verify your result of Q2.2.a using `pyro.do`. (5 points)

Question 3: Inverse probability weighting with a propensity score (13 points)

Probabilistic programming generally works by executing the program many times, and then reasoning on the ensemble of *program executions*, which vary because the program is probabilsitic. A program execution is typically called an *execution trace*, or just *trace*. The data structure representing a trace stores the values of the variables in the program, the log-probability of the trace, as well as other useful items. Pyro has a class called `Trace` that serves as a trace data structure. Given the following model:

```
def model():
  x = sample('x', Normal(0, 1))
  y = sample('y', Normal(x, 1))
  return x, y
```

Suppose you wanted to generate 3 samples from the model as well as the probability of each sample. You can use the following approach to handle and generate traces.

```
import numpy as np
trace_handler = pyro.poutine.trace(model)
for i in range(3):
  trace = trace_handler.get_trace()
  x = trace.nodes['x']['value']
  y = trace.nodes['y']['value']
  log_prob = trace.log_prob_sum()
  p = np.exp(log_prob)
  print(x, y, p)
```

3.1

Use the data in Question 2.1 to create the following propensity score function. (3 points)

```
def propensity(x, z):
    # returns P(X = x | Z = z)
    ...
```

3.2

Use the model from Question 2.1 to generate 1000 samples, along with the sample probabilities. Print the first 10 samples. (3 points)

3.3

Compute weighted joint probabilities for each possible combinations of X, Y, Z. Hint: Use your `propensity` function to create a list of weights for each combination, and multiplying the original joint probability of each combination by this weight. Normalize the weighted probabilities if they don't sum up to 1. (3 points)

3.4

Sample with replacement 1000 samples from the weighted probabilty distribution obtained in Question 3.3. (1 point)

3.5

Call this new set of samples Ω. Let $p^{\Omega}(X = x)$ be the proportion of times $X == x$ in Ω and $p^{\Omega}(X = x|Y = y)$ be the proportion of the Ω samples where $X == x$ after filtering for samples where $Y == y$. If you performed the above inverse probability weighting procedure correctly, then $P^{\text{model}}(Y = y|\text{do}(X = x)) \approx p^{\Omega}(Y = y|X = x)$ (the LHS and RHS are equal as the sample size goes to infinity). Confirm this by recalculating the causal effect from Question 2.1 using this method. (3 points)

Question 4: Structural Causal Models (23 points)

4.1 (3 points)

Consider the SCM $\mathbb{M}$:

$$\begin{aligned} X &:= N_X \\ Y &:= X^2 + N_Y \\ N_X, N_Y &\overset{\text{i.i.d}}{\sim} N(0,1) \end{aligned}$$

Write this model in Pyro and generate 10 samples of X and Y. (3 points)

4.2 (20 points)

Consider the SCM $\mathbb{M}$:

$$\begin{aligned} X &:= N_X \\ Y &:= 4X + N_Y \\ N_X, N_Y &\overset{\text{i.i.d}}{\sim} N(0,1) \end{aligned}$$

Hint: You need to create a sample name for each random variable in the model using pyro.sample. The reason the `sample` function has you name a variable (e.g. `"A"` in `sample("A", ...)`) is so you can store it by name in the trace object, and refer to that item later with expressions like `condition(model, {"A": a})` and `do(model, {"A": a})`. To create sample name for a continuous variable whose value depends on other variables, there are two ways. Supppose we want to create sample name for continuous random varianble Y=k*X+Ny (where X is another continous variable, k is a scalar parameter, and Ny is Gaussian noise with mean 0 and variance 1).

Method 1: Use a Normal distribution with very small variance (such as 0.01) to approximate a Delta distribution, which only has nonzero probability at one value. (Using pyro.dist.Delta.() in the model tend to cause computational issues in most approximate inference algorithms). Y can be written as follows using this method:

```
Y = k*X + pyro.sample('Ny', dist.Normal(0.0, 1.0))
Y = pyro.sample('Y', dist.Normal(Y, 0.01))
```

Method 2: Using AffineTransform, Y can be written as follows:

```
Ny_dist = dist.Normal(0.0, 1.0)
Y_dist = TransformedDistribution(Ny_dist, AffineTransform(a*X, tensor(1.0)))
Y = pyro.sample('Y', Y_dist)
```

4.2.a

Draw a picture of the model's DAG.(1 point)

4.2.b

$P_Y^{\mathbb{M}}$ is a normal distribution with what mean and variance? (2 points)

4.2.c

$P_Y^{\mathbb{M}:do(X=2)}$ is a normal distribution with what mean and variance? (2 points)

4.2.d

How and why does $P_Y^{\mathbb{M}:X=2}$ differ or ot differ from $P_Y^{\mathbb{M}:do(X=2)}$? (2 points)

4.2.e

$P_X^{\mathbb{M}:Y=2}$ is a normal distribution with what mean and variance? Note: Need explanation (2 Points)

4.2.f

$P_X^{\mathbb{M}:do(Y=2)}$ is a normal distribution with what mean and variance? (2 points)

4.2.g

Write model $P_{X,Y}^{\mathbb{M}}$ in code and generate 10 samples.(3 points)

4.2.h

Use the `do` operator to generate 100 samples from model $P_Y^{\mathbb{M}:do(X=2)}$ and visualize the results in a histogram.(3 points)

4.2.i

Use the `condition` operator and a Pyro inference algorithm to generate 10 samples from $P_X^{\mathbb{M}:Y=2}$. Use one of the Bayesian inference procedures described in the lecture notes.(3 points)