Homework 1

CS396-4 Causal Inference

January 20, 2022

Instructions

This assignment is due on Thursday, Jan 20 at 11:59pm CST. Late assignments will be accepted, but with a 14.3% (1/7th) penalty per day late. If your assignment is less than 24 hours late, we'll grade it and you'll receive 85.7% of those points; if it's less than 48 hours late, you'll receive 71.4% of those points. If it's more than 6 days late, you'll receive no points.

Your answers must be uploaded to Canvas as a single pdf document; you should edit the LaTeX source for this pdf to add in your answers. This is an individual assignment – you are welcome to discuss the problems with your classmates, but you must solve each part and write each answer on your own.

0.1 (1 point)

By submitting this assignment, you affirm that you have neither given nor received any unauthorized aid on this assignment, and that the solutions shown here are wholly you own. Any violation of Northwestern's academic integrity policies will result in you receiving a 0 on this assignment and a report to your dean's office. If you're ever worried about whether you are at risk of violating these policies, please ask – we can help you follow the rules, but we can't retract a report of suspected cheating.

1 Simpson's Paradox with synthetic data

For this question, you will need to run the code provided in `lecture2demo.py`. It uses the Python libraries `numpy`, `pandas`, and `statsmodels`. If you have trouble running this code, make a post on CampusWire and we'll provide help getting set up. If you are familiar with either `virtualenv` or `conda`, you may find it helpful to use such an environment to manage dependencies.

1.1 (1 point)

Fill in the tables by running the provided code with the indicated arguments. The code returns the mean and standard deviation of `repeats` samples. The first cell in each table is filled in for you. For example, in the top left table, the first cell's value is computed by running:

```
python lecture2demo.py observed --repeats 10 --c_dim 10 --ols "y ~ a"
```

You will use these tables to answer the next few questions.

observed(ols = "y $\sim$ a", . . .)

repeats	c_dim = 10	c_dim = 500
10	-1.625 ± 0.454	26.627 ± 73.807
100	-1.430 ± 0.472	36.458 ± 66.111
1000	-1.454 ± 0.544	29.842 ± 71.167

observed(ols = "y $\sim$ a + c", . . .)

repeats	c_dim = 10	c_dim = 500
10	0.409 ± 0.625	-0.521 ± 5.410
100	0.532 ± 0.483	-0.018 ± 3.406
1000	0.508 ± 0.534	0.508 ± 4.286

randomized(ols = "y $\sim$ a", . . .)

repeats	c_dim = 10	c_dim = 500
10	0.480 ± 0.387	-5.517 ± 12.106
100	0.473 ± 0.374	1.036 ± 14.663
1000	0.491 ± 0.384	0.558 ± 14.346

randomized(ols = "y $\sim$ a + c", . . .)

repeats	c_dim = 10	c_dim = 500
10	0.517 ± 0.213	0.706 ± 1.436
100	0.501 ± 0.203	0.780 ± 1.539
1000	0.499 ± 0.234	0.547 ± 1.571

1.2 (1 point)

The true causal effect of A on Y is 0.5. Which table has mean results furthest away from that value? Why?

The table based on **observed(ols = "y $\sim$ a")** seems to have mean values consistently furthest from 0.5, with all mean values corresponding to `c_dim = 10` below -1.4 and all mean values corresponding to `c_dim= 500` exceeding 26.6.

Estimated values are furthest from true value because we're not taking into account the entire structure of the data-generating process by omitting C from the model. That is, the parameter n that determines how many trials to run in generating the A data is influenced by the value of C (`n=1 + c_dim - c`) under the "observed" data-generating process, but the variables' dependence is not accounted for here. In addition, the value of C directly determines how many trials to run in generating the Y data (given that `n=a + c`).

observed(ols="y $\sim$ a+c") gives a better estimate because the model accounts for the role of C in determining the value of Y. Then **randomized(ols="y $\sim$ a")** gives a better estimate because under the **randomized** data-generating process the value of C does not affect the value of A (`n=1` for A). Finally, **randomized(ols="y∼a+c")** gives a better estimate both because C is included in the model *and* the value of C does not directly impact that of A.

1.3 (1 point)

How do the mean and standard deviation of the results change as you increase `c_dim` and `repeats`? What explains these trends?

An increase in `c_dim` consistently increases the standard deviation of results, but does *not* yield a consistent change in estimated mean. For example, increasing `c_dim` increases all mean estimates by over 25 under the **observed(ols = "y ∼ a")** model but increases some mean estimates while decreasing others under the **randomized(ols = "y ∼ a")** model. On the whole, however, the estimated means and variances increase because inflating `c_dim` increases the potential range of C and Y in all cases, and the potential range of A as well under observed data-generating processes (this is especially the case when C is not accounted for in the model).

Increases in `repeats` alone do not induce a consistent change in mean or standard deviation across the board (sometimes the statistics increase, sometimes they decrease). However, generally, by the law of large numbers the estimates converge to their expectations and their variances decrease.

However, increasing *both* `c_dim` and `repeats` yields an increase in mean estimates and standard deviation for estimates in all models.

1.4 (1 point)

Compare the `c_dim = 500` columns in the top right table (**observed(ols = "y ∼ a + c", ...)**) and the bottom left table (**randomized(ols = "y ∼ a", ...)**). This is the only comparison (for the same value of `c_dim`) where a column in a **observed** table has lower variance and a mean closer to 0.5 than a **randomized** table. Why does this happen for this comparison? Why doesn't it happen anywhere else?

The table **randomized(ols = "y ∼ a", ...)** fails to take into account the impact of C on Y. Not only that, it fails to consider the impact of C when it has an outsized impact on Y—that is, since `c_dim = 500` in the column, $Y \sim Binom(n = a + c, p = 0.5)$ could range from $n = 0 + 1 = 1$ trial with a corresponding expected value of $E[Y|A = 0, C = 1] = 0.5$ to $n = 1 + 501 = 502$ trials with a corresponding expected value of $E[Y|A = 1, C = 501] = 251$. As we fail to take into account C in this case, the variance and increased impact of C is ascribed to A. Hence both the parameter estimate associated with A and its standard deviation are inflated (with the standard deviation inflated to a much higher degree).

On the other hand, the **observed(ols = "y ∼ a + c", ...)** data-generating process includes C in its model, taking the variable with the greatest standard deviation into account explicitly. In addition, the fact that A depends on C means that both variables reflect the change in `c_dim`; it's not like we're inflating the potential value and variance of one variable and constraining the other (as is the case with the data-generating process with **randomized(ols = "y ∼ a", ...)**). While the value and variance of Y are inflated due to a much higher `c_dim`, both the impact of C on A in the data-generating process and the inclusion of A in the model contribute to a more reasonable estimate and standard deviation.

Thus **randomized(ols = "y ∼ a", ...)** performs poorly and **observed(ols = "y ∼ a + c", ...)** relatively well with `c_dim = 500`.

This doesn't happen elsewhere in `c_dim = 10` columns because the potential impact of C on the value and variance of Y is comparatively lower that with `c_dim = 500`. That impact isn't taken into account in the model with **observed(ols = "y ∼ a", ...)**, so that performs horribly, whereas it is taken into account with **randomized(ols = "y ∼ a+c", ...)**. It's only when we exaggerate the impact on C by setting `c_dim = 500` and compare a **randomized** model that does not take C into account with an **observed** model that does do we get this comparison (for the same value of `c_dim`) where a column in a **observed** table has lower variance and a mean closer to 0.5 than a **randomized** table.

2 Simpson's Paradox in expectation

	A=0	A=1
C=0	$x_1 = 0.93$ (81/87)	$x_2 = 0.87$ (234/270)
C=1	$x_3 = 0.73$ (192/263)	$x_4 = 0.69$ (55/80)
Both	$x_5 = 0.78$ (273/350)	$x_6 = 0.83$ (289/350)

Table 1: Simpson's Paradox, as covered in lecture. C is patient age, A is one of two drugs. Each cell shows average (binary) recovery rate Y. We've named the cells x_i to make them easier to reference below.

Consider Table 1, which we saw in lecture (for example, slide 9 of lecture 1). We said that this table shows Simpson's paradox because if you don't know the causal structure of the data, you can't tell which drug ($A = 0$ or 1) is better. If C were a mediator (like a side effect), we should compare x_5 against x_6 to see which drug is best. But if C were a confounder (like age), we should compare x_1 against x_2 and compare x_3 against x_4.

2.1 (1 point)

For $i = 1 \ldots 5$, define x_i as a conditional expectation involving Y, A, and C. For example, $x_6 = \mathbb{E}[Y \mid A = 1]$.

- $x_1 = E[Y|A = 0, C = 0]$
- $x_2 = E[Y|A = 1, C = 0]$
- $x_3 = E[Y|A = 0, C = 1]$
- $x_4 = E[Y|A = 1, C = 1]$
- $x_5 = E[Y|A = 0]$
- $x_6 = E[Y|A = 1]$

2.2 (1 point)

Consider the return statement in the `observed` function of `lecture2demo.py`:

```
smf.ols(ols, data=df).fit().params['a']
```

For both `ols = "y ~ a"` and `ols = "y ~ a + c"`, write out the value returned by this line in terms of the expectations you wrote out in your answer for 2.1. Explain your answer based on what `smf.ols` is doing.

You may assume that `c_dim = 2` and assume that C is always either 0 or 1. You may find it helpful to reference the `statsmodels` documentation here and here, as well as CampusWire post # 9 here.

The statement `smf.ols(ols="y ~  a", data=df).fit().params['a']` should return the value $x_5 - x_6 = E[Y|A = 0] - E[Y|A = 1]$. In this case, `smf.ols` is evaluating the difference in the expected value of Y given that A has a value of 1 versus a value of 0. That is, the statement aims to estimate the marginal change in Y due to a change in A by finding a value that "explains" the variables' relationship with minimal variance.

On the other hand, the statement `smf.ols(ols="y ~ a + c", data=df).fit().params['a']` should be returning the value $(x_1 - x_2) \cdot p(C = 0) - (x_3 + x_1) \cdot p(C = 1) = p(C = 0)(E[Y|A = 0, C = 0] - E[Y|A = 1, C = 0]) - p(C = 1)(E[Y|A = 0, C = 1] - E[Y|A = 1, C = 1])$. In this case, `smf.ols` should be comparing the difference in the expected value of Y given $A = 1$ versus $A = 0$ when $C = 0$ against the expected value of Y given $A = 1$ versus $A = 0$ when $C = 1$, weighting the context wherein $C = i$ with its marginal probability $p(C = i)$. That is, the statement aims to estimate the marginal change in Y due to a change in A, given the value of C, by finding a value that "explains" the variables' relationship with minimal variance.

2.3 (2 point)

Note that in Table 1, $p(C = 0) = 357/700 = 0.51$ and $p(C = 1) = 343/700 = 0.49$. Define $\Theta = p(C = 0) \cdot (x_1 - x_2) + p(C = 1) \cdot (x_3 - x_4) - (x_5 - x_6)$. What is the relationship between Θ and Simpson's Paradox? Explain.

Θ quantifies the difference in our estimate of the effect of drug taken *A* on recovery rate *Y* if we account for the confounding variable age *C* in our calculations versus if we (incorrectly) treat *C* as a mediator and do not explicitly account for its effect in our calculations. In short, Θ expresses the difference we'll get between treating *C* as a mediator and treating *C* as a confounder. Thus Θ illustrates the bias possible if we "fall for" (don't account for) Simpson's Paradox when analyzing the data by failing to understand the relationship between our variables (age, drug taken, recovery).

2.4 (2 point)

Consider the data-generating process implemented in the `observed()` function in `lecture2demo.py`. Suppose we let `c_dim = 10`; we must update our definition of Θ to include $p(C = 1), p(C = 2), \ldots,$ and $p(C = 10)$. We can also show that $\mathbb{E}[Y \mid A = 1] \approx 3.055$ and $\mathbb{E}[Y \mid A = 0] \approx 4.505$. With `c_dim` $= 10$ and with the new definition of Θ, what is $\mathbb{E}[\Theta]$ for the `observed()` data-generating process?

The "updated" definition of Θ proceeds as follows:

$$\Theta' = \Sigma_{i=1}^{10}[p(C = i) \cdot (E[Y|A = 0, C = i] - E[Y|A = 1, C = i])] - (E[Y|A = 0] - E[Y|A = 1])$$

Then taking the expectation:

$$E[\Theta'] = E[\Sigma_{i=1}^{10}[p(C = i) \cdot (E[Y|A = 0, C = i] - E[Y|A = 1, C = i])]] - E[(E[Y|A = 0] - E[Y|A = 1])]$$

Under the **observed** data-generating process, the variable Y is distributed $Y \sim Binomial(n = a+c, p = 0.5)$. As such, we easily determine that $E[Y|A = 0, C = 1] = 0.5$, $E[Y|A = 0, C = 2] = 1$, $E[Y|A = 0, C = 3] = 1.5$, and so forth until $E[Y|A = 0, C = 10] = 5$. Similarly, $E[Y|A = 1, C = 1] = 1$, $E[Y|A = 1, C = 2] = 1.5$, $E[Y|A = 1, C = 3] = 2$, and so forth until $E[Y|A = 1, C = 10] = 5.5$. Thus for all i, $i = 1, \ldots, 10$, $E[Y|A = 0, C = i] - E[Y|A = 1, C = i] = -0.5$.

We also know, by virtue of the fact that `c_dim = 10` means that $C \sim Uniform(1, 10)$, for all i, $i = 1, \ldots, 10$, $P(C = i) = \frac{1}{10}$. Furthermore, we're given that $\mathbb{E}[Y \mid A = 1] \approx 3.055$ and $\mathbb{E}[Y \mid A = 0] \approx 4.505$.

$$E[\Theta'] = E[\Sigma_{i=1}^{10}[(0.1) \cdot (-0.5)]] - E[(4.505 - 3.055)] = E[-0.5 - 1.45] = -1.95$$

So in this case, $\mathbb{E}[\Theta'] = -1.95$, as desired.

2.5 (2 point)

Let C, A, and Y be binary variables where C is patient age, A is drug assignment, and Y is recovery. Suppose we have some dataset sampled from the distribution $p(C, A, Y)$ that represents a randomized trial. Assume that A is marginally randomized, such that each patient has an equal probability of receiving $A = 0$ or $A = 1$. Using the definition of Θ from 2.3 above, use the rules of probability to show that for this distribution, $\mathbb{E}[\Theta] = 0$.

Recall that in (2.3) we found that:

$$\Theta = p(C = 0) \cdot (E[Y|A = 0, C = 0] - E[Y|A = 1, C = 0]) + p(C = 1)$$
$$\cdot (E[Y|A = 0, C = 1] - E[Y|A = 1, C = 1]) - (E[Y|A = 0] - E[Y|A = 1])$$

Then taking the expectation (and by the linearity property of expectation):

$$E[\Theta] = p(C = 0) \cdot E[(E[Y|A = 0, C = 0] - E[Y|A = 1, C = 0])]$$
$$+ p(C = 1) \cdot E[(E[Y|A = 0, C = 1] - E[Y|A = 1, C = 1])] - E[(E[Y|A = 0] - E[Y|A = 1])]$$

By definition of conditional expectation we can rewrite the above as:

$$E[\Theta] = p(C=0) \cdot E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0, C=0)}{P(A=0, C=0)})$$

$$-(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1, C=0)}{P(A=1, C=0)})]$$

$$+p(C=1) \cdot E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0, C=1)}{P(A=0, C=1)})$$

$$-(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1, C=1)}{P(A=1, C=1)})]$$

$$-E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0)}{P(A=0)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1)}{P(A=1)})]$$

As the problem suggests random assignment of treatment, we know that $A \perp C$ and so:

$$E[\Theta] = P(C=0) \cdot E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0, C=0)}{P(A=0)P(C=0)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1, C=0)}{P(A=1)P(C=0)})]$$

$$+P(C=1) \cdot E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0, C=1)}{P(A=0)P(C=1)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1, C=1)}{P(A=1)P(C=1)})]$$

$$-E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0)}{P(A=0)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1)}{P(A=1)})]$$

Then we find that:

$$E[\Theta] = E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(C=0)P(Y=y, A=0|C=0)}{P(A=0)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(C=0)P(Y=y, A=1|C=0)}{P(A=1)})]$$

$$+E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(C=1)P(Y=y, A=0|C=1)}{P(A=0)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(C=1)P(Y=y, A=1|C=1)}{P(A=1)})]$$

$$-E[(\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=0)}{P(A=0)}) - (\Sigma_{(y \in \mathcal{Y})} y \cdot \frac{P(Y=y, A=1)}{P(A=1)})]$$

Then rearranging and by the properties and definition of conditional expectation:

$$E[\Theta] = E[(\frac{P(C=0)E(Y=y, A=0|C=0) + P(C=1)E(Y=y, A=0|C=1)}{P(A=0)})$$

$$-(\frac{P(C=0)E(Y=y, A=1|C=0) + P(C=1)E(Y=y, A=1|C=1)}{P(A=1)})$$

$$-(\frac{E(Y=y, A=0)}{P(A=0)}) + (\frac{E(Y=y, A=1)}{P(A=1)})]$$

Given the properties of conditional expectation and that $\{C=0, C=1\}$ is a partition of the sample space of *C*:

$$E[\Theta] = E[(\frac{E(Y=y, A=0)}{P(A=0)}) - (\frac{E(Y=y, A=1)}{P(A=1)}) - (\frac{E(Y=y, A=0)}{P(A=0)}) + (\frac{E(Y=y, A=1)}{P(A=1)})]$$

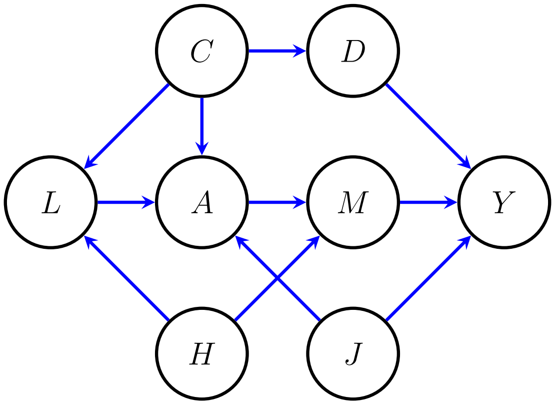
Hence:

$$E[\Theta] = E[0] = 0$$

as desired.

3 d-Separation in graphical models

Consider the following DAG:



3.1 (1 point)

List all collider ("head-to-head") nodes in the graph. For each, list all their descendants in the graph.

1. L (as its parents are H and C), with descendants A, M, and Y
2. A (as its parents are L, C, and J), with descendants M and Y
3. M (as its parents are A and H), with descendant Y
4. Y (as its parents are D, M, and J), with no descendants

3.2 (2 points)

For each of the following parts, we write "Is $A \perp B \mid C$" to mean "Is A d-separated (and therefore conditionally independent) of B given C?" For each question, provide your explanation in terms of blocked and unblocked paths. For example, if we asked, "Is $L \perp D \mid C$?" it would not be enough to say "yes" – you must explain, e.g.:

> Yes, because all paths from L to D go through C or Y, and the path $L \leftarrow C \rightarrow D$ is blocked by conditioning on C, the path $L \rightarrow A \leftarrow C \rightarrow D$ is blocked by conditioning on C, and the path $L \rightarrow A \rightarrow M \rightarrow Y \leftarrow D$ is blocked at Y, a collider.

1. Is $H \perp J \mid Y$? Why?

   No, because the path $H \rightarrow L \rightarrow M \rightarrow Y \leftarrow J$ is an unblocked path since we're conditioning on Y (if we didn't condition on Y, its status as a collider would block the referenced path). So H and J are not conditionally dependent given Y.

2. Is $H \perp J \mid L$? Why?

   Yes, because all paths from H to J must proceed through A or Y. But Y has no descendants and three parents—D, M, J. Thus Y will necessarily be a collider that will block paths from H to J through it (Y is a "dead end" and all paths through it are blocked). Then

$H \to M \leftarrow A$ is blocked as M is a collider, $H \to L \to A$ is blocked by conditioning on L, and in $H \to L \leftarrow C \to A \leftarrow J$, the node A blocks since it's a collider. Thus all paths from H to J are blocked.

3. Is $A \perp Y \mid J, M, C$? Why?

   Yes, because paths from A to Y must proceed through J, M or D. Conditioning on J blocks all paths that proceed through J, as $A \leftarrow J \to Y$. Paths proceeding through M, $A \to M \to Y$ and $A \leftarrow L \leftarrow H \to M \to Y$ are both blocked by conditioning on M. Finally, paths proceeding through D, $A \leftarrow C \to D \to Y$ and $A \leftarrow L \leftarrow C \to D \to Y$ are both blocked by conditioning on C. Therefore all paths from A to Y are blocked.

4. Is $L \perp Y \mid A, M, C, D$? Why?

   No, because the path $L \to A \leftarrow J \to Y$ is an unblocked path since we're conditioning on A (if we didn't condition on A, its status as a collider would block the referenced path). So L and Y are not conditionally dependent given A, M, C, D.