# lab06

March 5, 2020

## 1 Lab 6: Introduction to Causality and Causal Models

Welcome to the sixth DS102 lab!

The goals of this lab is to get a hands-on introduction to causality. This lab is based on the chapter on Causality from the Fairness and machine learning book. The lab is adapted from the subsection *A first example* under the section on *Causal models*.

The code you need to write is commented out with a message "TODO: fill in". There is additional documentation for each part as you go along.

### 1.1 Course Policies

**Collaboration Policy**

Data science is a collaborative activity. While you may talk with others about the labs, we ask that you **write your solutions individually**. If you do discuss the assignments with others please **include their names** in the cell below.

**Submission**: to submit this assignment, rerun the notebook from scratch (by selecting Kernel > Restart & Run all), and then print as a pdf (File > download as > pdf) and submit it to Gradescope.

**This assignment should be completed and submitted before Thursday March 5, 2020 at 11:59 PM.**

Write collaborator names here.

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import bernoulli
%matplotlib inline
```

## 2 1) Simulate data from a toy example (heart disease)

We will be using a toy example not intended to capture the real world. Imagine a hypothetical population in which an individual exercises regularly with probability $1/2$. With probability $1/3$, the individual has a latent disposition to develop overweight that manifests in the absence of regular exercise. Similarly, in the absence of exercise, heart disease occurs with probability $1/3$.

Denote by $X$ the indicator variable of regular exercise, by $W$ that of excessive weight, and by $H$ the indicator of heart disease. Below is a structural causal model to generate samples from this hypothetical population.

Let $U_1, U_2, U_3$ be independent Bernoulli random variables, i.e., biased coin flips: $U_1 \sim Bernoulli(1/2)$, $U_2 \sim Bernoulli(1/3)$, $U_3 \sim Bernoulli(1/3)$.

A Bernoulli random variable $Bernoulli(p)$ with bias parameter $p$ is a biased coin toss that takes value 1 with probability $p$ and value 0 with probability $1p$.

We will now define $X$, $W$, and $Z$ as functions of $U_1, U_2, U_3$:

1. $X = U_1$
2. $W = \begin{cases} 0 & \text{if } X = 1 \\ U_2 & \text{otherwise} \end{cases}$
3. $H = \begin{cases} 0 & \text{if } X = 1 \\ U_3 & \text{otherwise} \end{cases}$

### 2.1 1a) Question: Does being overweight cause heart disease in our model?

#TODO: No. Being overweight dose not cause heart disease in our model.

### 2.2 1b) Calculate $X, W, H$ from the samples of $U_1, U_2, U_3$.

Given $n$ independent samples of $U_1, U_2, U_3$, we will now calculate $X, W, H$ from those samples.

First, we will generate $n$ independent samples of $U_1, U_2$, and $U_3$.

[24]:
```
# Generate n independent samples of U1, U2, and U3.
n = 1000
U1 = bernoulli.rvs(1/2, size=n)
U2 = bernoulli.rvs(1/3, size=n)
U3 = bernoulli.rvs(1/3, size=n)
```

Using the $n$ samples of $U_1, U_2, U_3$ generated above, we will now calculate $n$ independent samples of $X, W$ and $H$.

Let $X = [X_1, ..., X_n]$, $W = [W_1, ..., W_n]$, and $H = [H_1, ..., H_n]$.

Each value $X_i$ represents whether or not the $i$th individual exercises. Each value of $W_i$ represents whether or not that same individual is overweight. Each value of $H_i$ represents whether or not the individual has heart disease.

[25]:
```
# No TODOs here, just run this cell and understand what's happening.
def calculate_X(U1_input):
    """Calculates X given U1.

    Args:
      U1_input, a numpy array of length n.

    Returns:
      X, a numpy array of length n.
    """
    return np.copy(U1_input)

X = calculate_X(U1)

print("Number of individuals who exercise: %d out of %d" % (np.sum(X), n))
```

```
Number of individuals who exercise: 512 out of 1000
```

[26]:
```
# TODO: calculate W from X and U2.
def calculate_W(X_input, U2_input):
    """Calculates W given X, U2.

    Args:
      X_input, a numpy array of length n.
      U2_input, a numpy array of length n.

    Returns:
      W, a numpy array of length n.
    """
    # TODO: calculate W based on X_input and U2_input.
    W = []
    for i in np.arange(len(X_input)):
        if X_input[i] ==1:
            W = np.append(W, 0)
        else:
            W = np.append(W,U2_input[i])

    return W

W = calculate_W(X, U2)

print("Number of individuals who are overweight: %d out of %d" % (np.sum(W), n))
```

```
Number of individuals who are overweight: 149 out of 1000
```

[27]:
```
# TODO: calculate H from X and U3.
def calculate_H(X_input, U3_input):
    """Calculates H given X, U3.

    Args:
      X_input, a numpy array of length n.
      U3_input, a numpy array of length n.

    Returns:
      H, a numpy array of length n.
    """
    # TODO: calculate H based on X_input and U3_input.
    H = []
    for i in np.arange(len(X_input)):
        if X_input[i] ==1:
            H = np.append(H, 0)
        else:
            H = np.append(H,U3_input[i])
    return H
```

```
H = calculate_H(X, U3)

print("Number of individuals with heart disease: %d out of %d" % (np.sum(H), n))
```

```
Number of individuals with heart disease: 176 out of 1000
```

### 2.3 1c) Calculate the baseline probability that an individual has heart disease.

While we know that being overweight does not cause heart disease in our model (since $H$ does not depend on $W$), we will now observe how the data reflects that relationship.

First, we will calculate the baseline probability that an individual has heart disease, $P(H = 1)$. To empirically calculate this probability for $n$ samples, we have

$$P(H = 1) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(H_i = 1)$$

where $\mathbf{1}(W_i = 1)$ is an indicator function that takes value 1 if $H_i = 1$ and 0 otherwise.

[28]:
```
# TODO: calculate the probability P(H=1).
def prob_H(H_input):
    """Calculates the baseline probability of heart disease, P(H = 1)

    Args:
      H_input, a numpy array of length n.

    Returns:
      a float probability between 0 and 1.
    """
    prob = np.mean(H_input)
    # TODO: calculate the probability P(H=1) using H_input.
    return prob

prob = prob_H(H)

print("Probability that an individual has heart disease: %f" % (prob))
```

```
Probability that an individual has heart disease: 0.176000
```

### 2.4 1d) Calculate the conditional probability that an individual has heart disease given that they are overweight.

The conditional probability that an individual has heart disease given that they are overweight is equal to

$$P(H = 1|W = 1) = \frac{P(H = 1, W = 1)}{P(W = 1)}.$$

To empirically calculate this conditional probability for $n$ samples, we have

$$P(W = 1) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(W_i = 1)$$

and

$$P(H = 1, W = 1) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(H_i = 1, W_i = 1),$$

Combining these with the formula for the conditional probability,

$$P(H = 1|W = 1) = \frac{P(H = 1, W = 1)}{P(W = 1)} \approx \frac{\sum_{i=1}^{n} \mathbf{1}(H_i = 1, W_i = 1)}{\sum_{i=1}^{n} \mathbf{1}(W_i = 1)}.$$

Next, we will empirically calculate $P(H = 1|W = 1)$ for the $n$ samples that we drew.

[31]:
```
# TODO: calculate the conditional probability P(H = 1 | W = 1) from n samples.
def probability_H_given_W(H_input, W_input):
    """Calculates P(H = 1 | W = 1) from n samples.

    Args:
      H_input, a numpy array of length n.
      W_input, a numpy array of length n.

    Returns:
      a float probability between 0 and 1.
    """
    # TODO: calculate P(H = 1 | W = 1) using H_input and W_input.
    prob_overweight = np.mean(W_input)
    overweight_disease = 0
    for i in np.arange(len(H_input)):
        if H_input[i] == 1 and W_input[i] == 1:
            overweight_disease = overweight_disease + 1
    prob_Overweight_Disease = overweight_disease / len(H_input)
    #prob_H_given_W = prob_Overweight_Disease / prob_overweight
    prob_H_given_W = overweight_disease / sum(W_input)
    # 1/n term cancle out each other, leaving only the sum
    return prob_H_given_W

prob = probability_H_given_W(H, W)
print("Probability that an individual has heart disease given that they are␣
 ↪overweight: %f" % (prob))
```

```
Probability that an individual has heart disease given that they are overweight:
0.322148
```

### 2.5 1e) Question: Which was greater, the baseline probability of heart disease P(H=1), or the conditional probability of heart disease given overweight, P(H=1 | W=1)? Can you explain why one of them was greater?

#TODO: The conditional probability of heart disease given overweight, P(H=1 | W=1) is greater. The conditional probability might be greater because although heart diseases are not cause by overweight, heartdisease is related to exercise(no enough exercise). So being overweight indicate that people might not exerice regularly. That's why the conditional probability of heart disease is higher.

# 3 2) Comparison to a hypothetical substitution (*do-operator*)

In 1), we observed that even though being overweight does not heart disease in our model, we can still observe that the probability of getting heart disease $H$ conditioned on whether or not one is overweight $W$ is greater than the baseline probability of getting heart disease. If conditional probability is not a good indicator of whether or not being overweight causes heart disease in our model, what else can we do to check whether or not being overweight causes heart disease?

Instead, we could set W = 1, resulting in a new distribution. The active substitution $W = 1$ creates a new hypothetical population in which all individuals are overweight with all that it entails in our model. If this were real life, what we would be doing here is making *all* of the individuals in our study overweight, and observing whether or not they develop heart disease. This substitution of $W = 1$ is sometimes called the *do-operator*.

Specifically, this time, we will generate the data using the following model:

1. $X = U_1$
2. $W = 1$
3. $H = \begin{cases} 0 & \text{if } X = 1 \\ U_3 & \text{otherwise} \end{cases}$

This model has the exact same functions as 1) for $X$ and $H$, but now it has $W = 1$ instead of the previous function for $W$.

## 3.1 2a) Calculate the new $X, W, H$ samples.

[33]:
```
# TODO: calculate W, a vector of all 1s.
def calculate_W_all_ones():
    """Calculates W, where W is a vector of all 1s.

    Returns:
      W, a numpy array of length n where all entries are 1.
    """
    length = len(W)
    return np.ones(length)
# TODO: calculate the hypothetical W, a numpy array of length n where all␣
 ↪entries are 1.

W_do = calculate_W_all_ones()

print("Number of individuals who are overweight: %d out of %d" % (np.sum(W_do),␣
 ↪n))
```

```
Number of individuals who are overweight: 1000 out of 1000
```

[34]:
```
# Calculate X and H using the same functions from 1).
# No TODOs here, we're reusing the same functions for X and H from 1).
X_do = calculate_X(U1)
print("Number of individuals who exercise: %d out of %d" % (np.sum(X_do), n))
```

```
H_do = calculate_H(X_do, U3)
print("Number of individuals with heart disease: %d out of %d" % (np.sum(H_do),␣
 ↪n))
```

```
Number of individuals who exercise: 512 out of 1000
Number of individuals with heart disease: 176 out of 1000
```

### 3.2 2b) Calculate the probability that an individual has heart disease with the hypothetical substitution W = 1.

[35]:
```
# No TODOs here, we will reuse the same prob_H function that we wrote in 1).
prob = prob_H(H_do)

print("Probability that an individual has heart disease: %f" % (prob))
```

```
Probability that an individual has heart disease: 0.176000
```

#### 3.2.1 Question: Is probability that an individual has heart disease under the active substitution W=1 (from 2b) the same as the probability that an individual has heart disease conditioned on W=1 (from 1d)? Why might they be different?

TODO: The probability is not the same.Since our model assume there is no casual relationship between overweight and heart diseases, also the overweight is related(or cause by not enough exercise), not the other way around, forcing all W to be 1, then fing the exercise, it's different from giving exercise condition then finding overweight. Thus, the populations from forcing every data to be overweight is different from conditioning on W=1, which will results in different datas.

## 4 3) Another hypothetical distribution where exercise habits are driven by body weight.

In the distributions from 1) and 2), heart disease $H$ was not directly caused by weight $W$. This time, we will observe a different hypothetical model where overweight individuals choose to exercise with some probability, but that's the only reason anyone would exercise. Heart disease develops in the absence of exercise.

We will now define $X$, $W$, and $Z$ as functions of $U_1, U_2, U_3$ for this new model:

1. $W = U_2$
2. $X = \begin{cases} 0 & \text{if } W = 0 \\ U_1 & \text{otherwise} \end{cases}$
3. $H = \begin{cases} 0 & \text{if } X = 1 \\ U_3 & \text{otherwise} \end{cases}$

### 4.1 3a) Question: Is there a causal relationship between heart disease and weight in this model?

TODO: Yes, there is a casual relationship between heart disease and weight in this model. The model create a casual chain(gateway) as W –> X –> W. So W and H has a casual relationship.

### 4.2 3b) Calculate $X, W, H$ in this new model.

[39]:
```
# Generate n independent samples of U1, U2, and U3.
n = 1000
U1 = bernoulli.rvs(1/2, size=n)
U2 = bernoulli.rvs(1/3, size=n)
U3 = bernoulli.rvs(1/3, size=n)
```

[40]:
```
# No TODOs here, calculate the new W.
def calculate_W_new(U2_input):
    """Calculates W given U2 for the new model.

    Args:
      U2_input, a numpy array of length n.

    Returns:
      W, a numpy array of length n.
    """
    return np.copy(U2_input)

W_new = calculate_W_new(U2)

print("Number of individuals who are overweight: %d out of %d" % (np.
 ↪sum(W_new), n))
```

```
Number of individuals who are overweight: 346 out of 1000
```

[43]:
```
# TODO: calculate the new X based on W and U1.
def calculate_X_new(W_input, U1_input):
    """Calculates X given W, U1 for the new model.

    Args:
      W_input, a numpy array of length n.
      U1_input, a numpy array of length n.

    Returns:
      X, a numpy array of length n.
    """
    X = []
    for i in np.arange(len(W_input)):
        if W_input[i] == 0:
            X = np.append(X,0)
```

```
        else:
            X = np.append(X,U1_input[i])

    # TODO: calculate X based on W_input and U1_input.
    return X

X_new = calculate_X_new(W_new, U1)

print("Number of individuals who exercise: %d out of %d" % (np.sum(X_new), n))
```

```
Number of individuals who exercise: 170 out of 1000
```

[44]:
```
# No TODOs here, we can reuse the function from 1) for H.
H_new = calculate_H(X_new, U3)
print("Number of individuals with heart disease: %d out of %d" % (np.
 ↪sum(H_new), n))
```

```
Number of individuals with heart disease: 288 out of 1000
```

### 4.3 Calculate the baseline probability that an individual has heart disease.

We will now calculate the baseline probability that an individual has heart disease in this new model, $P(H = 1)$.

[45]:
```
# No TODOs here, we just reuse the function we implemented in 1).
prob = prob_H(H_new)

print("Probability that an individual has heart disease: %f" % (prob))
```

```
Probability that an individual has heart disease: 0.288000
```

### 4.4 Calculate the conditional probability that an individual has heart disease given that they are overweight.

As done in 1), we will now empirically calculate $P(H = 1|W = 1)$ for the $n$ samples that we drew for the new model.

[46]:
```
# No TODOs here, just reuse the function we implemented in 1).
prob = probability_H_given_W(H_new, W_new)
print("Probability that an individual has heart disease given that they are␣
 ↪overweight: %f" % (prob))
```

```
Probability that an individual has heart disease given that they are overweight:
0.176301
```

### 4.5 3c) Comparison to a hypothethical substitution (*do-operator*) for this new model

As we did for the model in 1, we will no observe what happens to $H$ when we set W = 1, resulting in a new hypothetical population in which all individuals are overweight with all that it entails in our model.

Specifically, this time, we will generate the data using the following model:

1. $W = 1$
2. $X = \begin{cases} 0 & \text{if } W = 0 \\ U_1 & \text{otherwise} \end{cases}$
3. $H = \begin{cases} 0 & \text{if } X = 1 \\ U_3 & \text{otherwise} \end{cases}$

This model has the exact same functions as for $X$ and $H$, but now it has $W = 1$ instead of the previous function for $W$.

[47]:
```
# No TODOs here, just reuse the function from part 1).
W_do = calculate_W_all_ones()

print("Number of individuals who are overweight: %d out of %d" % (np.sum(W_do),␣
 ↪n))
```

```
Number of individuals who are overweight: 1000 out of 1000
```

[48]:
```
# No TODOs here, use the same X and H from part b).
X_do = calculate_X_new(W_do, U1)
print("Number of individuals who exercise: %d out of %d" % (np.sum(X_do), n))

H_do = calculate_H(X_do, U3)
print("Number of individuals with heart disease: %d out of %d" % (np.sum(H_do),␣
 ↪n))
```

```
Number of individuals who exercise: 487 out of 1000
Number of individuals with heart disease: 172 out of 1000
```

#### 4.5.1 Calculate the probability that an individual has heart disease with the hypothetical substitution W = 1.

[49]:
```
# No TODOs here, we will reuse the same prob_H function that we wrote in 1).
prob = prob_H(H_do)

print("Probability that an individual has heart disease: %f" % (prob))
```

```
Probability that an individual has heart disease: 0.172000
```

#### 4.5.2 Question: Compared to your answer to 2b), is the probability that an individual has heart disease under the active substitution W=1 (from 3c) closer to probability that an individual has heart disease conditioned on W=1 (from 3b)?

The probability that an individual has heart disease under the active substitution is closer to the conditional probablity that having heart disease given overweight condition. Compare to previous one(2b), where there's no causal realtionship between W and H, this new model is refeling a casual chain of W–>X–>H.

### 4.6 Summary

The active hypothethical substitution of $W = 1$ for the new model in 3) leads to an increased probability of exercise, hence lowering the probability of heart disease. In this case, the conditioning on $W = 1$ has the same affect. Both lead to a probability of about $1/6$.

What we see is that fixing a variable by substitution may or may not correspond to a conditional probability. This is a formal rendering of our earlier point that observation isn't action. A substitution corresponds to an action we perform. By substituting a value we break the natural course of action our model captures. This is the reason why the substitution operation is sometimes called the do-operator, written as $\text{do}(W := 1)$.

For more information on this example and for next steps in the studying causality, see the chapter on Causality from the Fairness and machine learning book.

[ ]: