Simpson's Paradox

Bookmark this page

Exercise: Simpson's Paradox

15 points possible (graded)

This problem looks at a real-life situation in which a school was accused of gender bias, seemingly admitting more men than women.

We have the data encoded as a 3D joint probability table across three random variables G for gender (either female or male), D for the department someone applied to (there are 6 departments in consideration, which for simplicity we will just call A, B, C, D, E, and F), and A for the admissions decision (either admitted or rejected).

Thus, $p_{G,D,A}(\text{female}, \text{C}, \text{admitted})$ refers to the probability that a woman applying to department C gets admitted.

The joint probability table is provided in the file simpsons_paradox_data.py. Please download that and make sure it's in the same working directory as your IPython prompt (the same way you were able to import `comp_prob_inference.py`).

Now let's load in everything from `simpsons_paradox_data.py`:

```
from simpsons_paradox_data import *
```

Please open up `simpsons_paradox_data.py` to see what Python variables you have access to, and to see how we can turn a probability space represented as a Python dictionary into the 3D joint probability table, stored as a 3D NumPy array.

For example, to get the probability that a woman applied to department C and got admitted, you can enter into the prompt:

```
joint_prob_table[gender_mapping['female'], department_mapping['C'],
admission_mapping['admitted']]
```

Some terminology: In Python, especially when we deal with NumPy arrays higher than 2 dimensions, talking about rows and columns is confusing. We will instead refer to the different axes as axis 0 (in this case: gender), axis 1 (in this case: department), axis 2 (in this case: admission decision), and so forth if we have more than 3 dimensions.

Let's see what the paradox is. First, let's look at the probability that women were admitted vs the probability that men were admitted. This means that we are going to marginalize out the department (again, this is axis 1).

Marginalization is easy to do with NumPy:

```
joint_prob_gender_admission = joint_prob_table.sum(axis=1)
```

On the right-hand side, `.sum(axis=1)` says to sum across axis 1 so that we no longer have axis 1! This means that the Python variable `joint_prob_gender_admission` stores a 2D joint probability table for random variables G and A.

Now, for example, the probability that a woman applies and is admitted is given by:

```
joint_prob_gender_admission[gender_mapping['female'], admission_mapping['admitted']]
```

Now let's do some conditioning to see what the probability is of being admitted given that the applicant is female. Remember, what we want is the following conditional probability:

$$\begin{aligned} p_{A|G}(\textbf{admitted}|\textbf{female}) &= \frac{p_{A,G}(\textbf{admitted}, \textbf{female})}{p_G(\textbf{female})} \\ &= \frac{p_{A,G}(\textbf{admitted}, \textbf{female})}{\sum_a p_{A,G}(a, \textbf{female})} \\ &= \frac{p_{A,G}(\textbf{admitted}, \textbf{female})}{p_{A,G}(\textbf{admitted}, \textbf{female}) + p_{A,G}(\textbf{rejected}, \textbf{female})}. \end{aligned}$$

Let's restrict the joint probability table of G and A so that we only look at when $G = \textbf{female}$:

```
female_only = joint_prob_gender_admission[gender_mapping['female']]
```

Now this corresponds to a vector that we have to normalize to be 1 to get a valid conditional probability table!

```
prob_admission_given_female = female_only / np.sum(female_only)
```

This is the right conditional probability table, represented as an array. To get it into the dictionary format we've been dealing with earlier in the course, we do:

```
prob_admission_given_female_dict = dict(zip(admission_labels, prob_admission_given_female))
print(prob_admission_given_female_dict)
```

What is $\mathbb{P}(A = \textbf{admitted}|G = \textbf{female})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

Answer: 0.30333514986376009

What is $\mathbb{P}(A = \textbf{admitted} | G = \textbf{male})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

Answer: 0.44519509476031227

So it looks like there's some gender bias going on! Let's investigate by looking at how things differ by each department.

But before we do so, we just wanted to say a word about conditioning in code on axes aside from axis 0. With our 2D joint probability table of G and A, to condition on admission decision instead, there's a slightly different syntax in Python. Let's condition on being admitted:

```
admitted_only = joint_prob_gender_admission[:, admission_mapping['admitted']]
```

Notice that we had to put a "`:,`". That is to indicate that we want to keep everything in the 0-th axis. We didn't have to do this when we conditioned on a value in the 0-th axis, since it is implied that you want everything in the axis 1 in that case.

Thus, the conditional probability table of gender given admitted is:

```
prob_gender_given_admitted = admitted_only / np.sum(admitted_only)
prob_gender_given_admitted_dict = dict(zip(gender_labels, prob_gender_given_admitted))
print(prob_gender_given_admitted_dict)
```

All right, now let's look at which departments are favoring men over women.

For the following part, we will condition on both G and D taking on specific values together. For example, to only look at the entries in the 3D joint probability table for when $G = \textbf{female}$ and, at the same time, $D = \textbf{A}$, then we can do the following:

```
female_and_A_only = joint_prob_table[gender_mapping['female'], department_mapping['A']]
```

Now let's determine the probability of getting admitted given each gender and each department.

- **Department A:**

  What is $\mathbb{P}(A = \text{admitted} | G = \text{female}, D = \text{A})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.82

  What is $\mathbb{P}(A = \text{admitted} | G = \text{male}, D = \text{A})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.62

- **Department B:**

  What is $\mathbb{P}(A = \text{admitted} | G = \text{female}, D = \text{B})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.68

  What is $\mathbb{P}(A = \text{admitted} | G = \text{male}, D = \text{B})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

Answer: 0.63

- **Department C:**

  What is $\mathbb{P}(A = \text{admitted} | G = \text{female}, D = \text{C})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.34

  What is $\mathbb{P}(A = \text{admitted} | G = \text{male}, D = \text{C})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.37

- **Department D:**

  What is $\mathbb{P}(A = \text{admitted} | G = \text{female}, D = \text{D})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.35

  What is $\mathbb{P}(A = \text{admitted} | G = \text{male}, D = \text{D})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

Answer: 0.33

- **Department E:**

  What is $\mathbb{P}(A = \textbf{admitted} | G = \textbf{female}, D = \textbf{E})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.24

  What is $\mathbb{P}(A = \textbf{admitted} | G = \textbf{male}, D = \textbf{E})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.28

- **Department F:**

  What is $\mathbb{P}(A = \textbf{admitted} | G = \textbf{female}, D = \textbf{F})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

  Answer: 0.07

  What is $\mathbb{P}(A = \textbf{admitted} | G = \textbf{male}, D = \textbf{F})$? (Please be precise with at least 3 decimal places, unless of course the answer doesn't need that many decimal places.)

Answer: 0.06

How many of these departments have a higher probability of admitting women than of admitting men?

Answer: 4

Somehow, it seems that when we marginalized out the department, the gender bias is going one direction, yet when looking at the specific departments, most departments seem to be having the bias go the other direction!

Take-away message: We have to be very careful when interpreting conditional probabilities! Also, marginalization (which lumps different groups of data together, where here the groups are departments) can reverse trends that appear in specific groups!

Solution:

To compute $\mathbb{P}(A = \mathbf{admitted} | G = \mathbf{female})$, we do (we actually provided most of this code above):

```
from simpsons_paradox_data import *
joint_prob_gender_admission = joint_prob_table.sum(axis=1)
female_only = joint_prob_gender_admission[gender_mapping['female']]
prob_admission_given_female = female_only / np.sum(female_only)
prob_admission_given_female_dict = dict(zip(admission_labels, prob_admission_given_female))
print(prob_admission_given_female_dict['admitted'])
```

The answer that gets printed out is **0.303335149864**.

To compute $\mathbb{P}(A = \mathbf{admitted} | G = \mathbf{male})$, we do:

```
male_only = joint_prob_gender_admission[gender_mapping['male']]
prob_admission_given_male = male_only / np.sum(male_only)
prob_admission_given_male_dict = dict(zip(admission_labels, prob_admission_given_male))
print(prob_admission_given_male_dict['admitted'])
```

The answer that gets printed out is **0.44519509476**.

Now to get the probability of being admitted given each of the genders and for each of the departments (i.e., to answer the rest of the exercise), we can actually just code a few lines that prints out all the answers:

```
for department in department_labels:
    for gender in gender_labels:
        restricted = joint_prob_table[gender_mapping[gender], department_mapping[department]]
        print(department, gender, dict(zip(admission_labels, restricted / np.sum(restricted)))
['admitted'])
```

The output of the above code is:

```
A female 0.82
A male 0.62
B female 0.68
B male 0.63
C female 0.34
C male 0.37
D female 0.35
D male 0.33
E female 0.24
E male 0.28
F female 0.07
F male 0.06
```

For each line, the first two values are the department and gender that we are conditioning on, and the third value is the conditional probability of being admitted.

Submit You have used 0 of 5 attempts

Discussion

Topic: Jointly Distributed Random Variables / Exercise: Simpson's Paradox

Hide Discussion

Add a Post

another very poor section with no background given before one does this hard exercise

discussion posted 2 months ago by anonymous

very poor instruction

This post is visible to everyone.

+ Expand discussion

code readability in Simpson's Paradox exercise

discussion posted 2 months ago by **Greg1950**

As a person with a (quite old) math degree, I've seen all sorts of arcane notation and am reasonably comfortable with such. But I found the...

This post is visible to everyone.

+ Expand discussion

[Paper] Understanding Simpson's Paradox

discussion posted 2 months ago by **antonms**

HI,

Here's an interesting paper on the topic.

Best, Anton

This post is visible to everyone.

+ Expand discussion

Precision in numpy arrays

discussion posted 2 months ago by **DJamesW**

Suggesting that the precision be 3 decimal places enticed me to learn a lot about adjusting the precision of numpy arrays, but it took some time...

This post is visible to everyone.

+ Expand discussion

Help with normalization with two conditionals

question posted 2 months ago by **jpdjeredjian**

I followed from the example how to normalize over one variable (gender), but I still cant get how to normalize over two (gender and department,...

This post is visible to everyone.

+ Expand discussion

Failure to explain notation

discussion posted 2 months ago by **Bill_Connelly**

I don't know if the course organizers are reading this, but at no point have we been taught what [{P}(A = \text {admitted} | G = \text {female},...

This post is visible to everyone.

+ Expand discussion

one line solution for joint probability table Simpson's paradox

discussion posted 2 months ago by **tomvansteijn**

Could be constructed using nested list expressions (add to end of file):

`joint_prob_table_2 = np.array([[[prob_space[(g, d, l)]` `for...

This post is visible to everyone.

+ Expand discussion

Bug on line 46?

discussion posted 2 months ago by **Tikke**

I think there is a bug on line 46:

```
                department_mapping[gender],
```

This post is visible to everyone.

+ Expand discussion

The problem's premise that looking at dept "bias" is misleading-- conflated by distribution of women/men in each dept's applicant pool!

discussion posted 2 months ago by anonymous

I don't think the individual depts show a bias in the other direction -- the last set of questions are a bit misleading if you want to talk about...

This post is visible to everyone.

+ Expand discussion

simpsons_paradox_data.py bug

discussion posted 2 months ago by **RADUGROSU** **Community TA**

The indexing into `department_mapping` in the last line was incorrect. Also, for python 3+, the initialization of `joint_prob_table` was also...

This post is visible to everyone.

+ Expand discussion

Use of :

discussion posted 2 months ago by **DanielFelipeLopezVelasquez**

I didn't understand the following :

"Notice that we had to put a “:,". That is to indicate that we want to keep everything in the 0-th axis."

I...

This post is visible to everyone.

+ Expand discussion

use labels not mapping

discussion posted 2 months ago by **dnogues**

I used admission_mapping instead of admission_labels when making a dictionary of the conditioned probability by each department. The zip method...

This post is visible to everyone.

+ Expand discussion

Simpson's Paradox - Python

question posted 2 months ago by **surajitmishra**

I understood what Simpson's paradox is but I am unable to understand what the code is doing. Could anybody kindly explain. Its getting me frustrated....

This post is visible to everyone.

+ Expand discussion

dict(zip(...))

question posted 2 months ago by **AllisonABC**

I'm getting an error ('dict object is not callable') from this line of code. What am I doing wrong?

prob_admission_given_female_dict = dict(zip(admission_labels,...

This post is visible to everyone.

+ Expand discussion

Is the code for P(Admission | Female) correct?

question posted 2 months ago by **skumar34**

P(Admission | female) = P(Admission and Female) / P(female)

So according to above expression it should be female_only[admission_mapping['admitted']]/np.sum(female_only)

but...

This post is visible to everyone.

+ Expand discussion

Understanding the meaning

discussion posted 2 months ago by **osm3000**

Hi,

I got all the numbers right, but I don't think I get how to determine if there is "a gender bias" correctly.

Computing `P (admitted |...

This post is visible to everyone.

+ Expand discussion

Struggling to understand

discussion posted 2 months ago by **Graeme61**

So far, I got the prob_admission_given_female and prob_admission_given_male figures right.

and I got *one* of the other values correct, which...

This post is visible to everyone.

+ Expand discussion

lambda for normalize

discussion posted 2 months ago by **tomvansteijn**

What helped me is to create a small lambda function for normalizing arrays, such as:

```
normalize = lambda a: a / a.sum()
```

This post is visible to everyone.

+ Expand discussion

Axis sequence in arrays

discussion posted 2 months ago by **DJamesW**

It appears as if axis 0 (gender) is the "z-axis" in the array. Would the usual sequence not be x, y, z ...? It doesn't really matter of course,...

This post is visible to everyone.

+ Expand discussion

Corrected code

discussion posted 2 months ago by **JeffersonMendesSilva**

For the ones who couldn't corrected the code of the file, im posting part of the corrected code here, based on the answer of RADUGROSU

```
gender_labels…
```

This post is visible to everyone.

+ Expand discussion

1 2 Next >

© All Rights Reserved

© 2016 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY OPENedX

Download on the App Store

ANDROID APP ON Google play