MITx: 6.008.1x Computational Probability and Inference

Bookmarks

▸ Introduction

▾ **1. Probability and Inference**

Introduction to Probability
Exercises due Sep 22, 2016 at 02:30 IST

Probability Spaces and Events
Exercises due Sep 22, 2016 at 02:30 IST

Random Variables
Exercises due Sep 22, 2016 at 02:30 IST

1. Probability and Inference > Random Variables > Two Ways to Specify a Random Variable in Code

Bookmark

Follow along in an IPython prompt.

From the video, we see that two ways we can fully represent a random variable on a computer are as follows.

Approach 1. Go with the mathematical definition of a random variable. First, specify what the underlying probability space is:

```
> prob_space = {'sunny': 1/2, 'rainy': 1/6, 'snowy': 1/3}
```

Then provide a way to map from the sample space to the alphabet:

```
> W_mapping = {'sunny': 'sunny', 'rainy': 'rainy', 'snowy': 'snowy'}
> I_mapping = {'sunny': 1, 'rainy': 0, 'snowy': 0}
```

Then we can generate a random sample/draw for random variables W and I:

```
> random_outcome = comp_prob_inference.sample_from_finite_probability_space(prob_space)
> W = W_mapping[random_outcome]
> I = I_mapping[random_outcome]
```

Approach 2. Remember how we wrote out probability tables for random variables W and I? Let's directly store these probability tables:

```
> W_table = {'sunny': 1/2, 'rainy': 1/6, 'snowy': 1/3}
> I_table = {0: 1/2, 1: 1/2}
```

Treating the tables as probability spaces, draw samples for W and I:

```
> W = comp_prob_inference.sample_from_finite_probability_space(W_table)
> I = comp_prob_inference.sample_from_finite_probability_space(I_table)
```

Exercise: Two Ways to Specify a Random Variable in Code

(4/4 points)

- In approach 1, is it possible that the variable `W` stored at the end is equal to `'sunny'`, yet the variable `I` stored at the end is equal to `0`? (Feel free to re-run the code for approach 1 a few times.)

  ○ Yes

  ● No ✓

- In approach 2, is it possible that the variable W stored at the end is equal to sunny, yet the variable I stored at the end is equal to 0? (Feel free to re-run the code for approach 2 a few times.)

  ● Yes ✓

  ○ No

- In approach 1, from running through the code once, the number of times a random sample is drawn is

  1

  ✓ **Answer:** 1 .

- In approach 2, from running through the code once, the number of times a random sample is drawn is

  2

  ✓ **Answer:** 2 .

Solution:

- In approach 1, is it possible that the variable `W` stored at the end is equal to `'sunny'`, yet the variable `I` stored at the end is equal to `0`? (Feel free to re-run the code for approach 1 a few times.)

  No because `W` and `I` are both assigned based on the same random outcome `random_outcome`.

- In approach 2, is it possible that the variable W stored at the end is equal to sunny, yet the variable I stored at the end is equal to 0? (Feel free to re-run the code for approach 2 a few times.)

  Yes (just running the code a few times makes it clear that this is the case).

- In approach 1, from running through the code once, the number of times a random sample is drawn is **1**.

- In approach 2, from running through the code once, the number of times a random sample is drawn is **2**.

  Note that each time `comp_prob_inference.sample_from_finite_probability_space` is called, another sample is drawn.

You have used 3 of 5 submissions

What happened?

In approach 2, even though W by itself has the right probability table, and I by itself has the right probability table, somehow we have lost how they should behave together! Can you see how approach 1 avoids this problem?

Later on in the course, we'll see many more examples of random variables that are related!