

# Computational Probability and Inference

L<sup>A</sup>T<sub>E</sub>Xdocument by Colin Leach, from an HTML original by MIT course staff

September 2016

## **Abstract**

This is a set of notes for the online course “6.008.1x Computational Probability and Inference” given by the staff of MIT during Fall 2016. Most of the sections are copied from notes on the edX platform (with grateful thanks to those at MIT who provided them!), but I added some further sections as necessary. You should assume that everything here is MIT copyright.

All  $\text{\LaTeX}$ source is currently available at <https://github.com/colinleach/probinf-notes> but there is no guarantee that it will remain publicly accessible. I will take guidance from MIT about what they want done with the repository.

# Contents

<b>1</b>	<b>Probability and Inference</b>	<b>3</b>
1.1	Introduction to Probability	3
1.1.1	Introduction	3
1.1.2	A First Look at Probability	3
1.1.3	Probability and the Art of Modeling Uncertainty	6
1.2	Probability Spaces and Events	6
1.2.1	Two Ingredients to Modeling Uncertainty	6
1.2.2	Probability Spaces	7
1.2.3	Table Representation	8
1.2.4	More on Sample Spaces	8
1.2.5	Probabilities with Events	8
1.2.6	Events as Sets	9
1.2.7	Code for Dealing with Sets in Python	9
1.2.8	Probabilities with Events and Code	9
1.3	Random Variables	9
1.3.1	A First Look at Random Variables	9
1.3.2	Random Variables	10
1.3.3	Random Variables Notation and Terminology	12
1.4	Jointly Distributed Random Variables	12
1.4.1	Relating Two Random Variables	12
1.4.2	Representing a Joint Probability Table in Code	14
1.4.3	Marginalization	16
1.4.4	Marginalization for Many Random Variables	17
1.4.5	Conditioning for Random Variables	19
1.4.6	Moving Toward a More General Story for Conditioning	20
1.5	Conditioning on Events	20
1.5.1	Conditioning on Events Intro	20
1.5.2	The Product Rule for Events	20
1.5.3	Bayes' Theorem for Events	20
1.5.4	Practice Problem: Bayes' Theorem and Total Probability	20
1.5.5	Take-Away Lessons:	21
1.6	Inference with Bayes' Theorem for Random Variables	22
1.6.1	The Product Rule for Random Variables (Also Called the Chain Rule)	22
1.6.2	Bayes' Rule for Random Variables (Also Called Bayes' Theorem for Random Variables)	23
1.6.3	Bayes' Theorem for Random Variables: A Computational View	24
1.6.4	Maximum A Posteriori (MAP) Estimation	25
1.7	Independence Structure	25
1.7.1	Independent Events	25
1.7.2	Independent Random Variables	26
1.7.3	Mutual vs Pairwise Independence	27
1.7.4	Conditional Independence	28

1.7.5	Explaining Away . . . . .	29
1.7.6	Practice Problem: Conditional Independence . . . . .	31
1.8	Decisions and Expectations . . . . .	32
1.8.1	Introduction to Decision Making and Expectations . . . . .	32
1.8.2	The Expected Value of a Random Variable . . . . .	33
1.8.3	Variance and Standard Deviation . . . . .	35
1.8.4	Practice Problem: The Law of Total Expectation . . . . .	35
1.9	Measuring Randomness . . . . .	36
1.9.1	Introduction to Information-Theoretic Measures of Randomness . . . . .	36
1.9.2	Shannon Information Content . . . . .	37
1.9.3	Shannon Entropy . . . . .	37
1.9.4	Information Divergence . . . . .	38
1.9.5	Proof of Gibbs' Inequality . . . . .	40
1.9.6	Mutual Information . . . . .	42
1.9.7	Exercise: Mutual Information . . . . .	42
1.9.8	Information-Theoretic Measures of Randomness: Where We'll See Them Next . . . . .	43
1.10	Towards Infinity in Modeling Uncertainty . . . . .	43
1.10.1	Infinite Outcomes . . . . .	43
1.10.2	The Geometric Distribution . . . . .	44
1.10.3	Practice Problem: The Geometric Distribution . . . . .	44
1.10.4	Infinite Outcomes . . . . .	44
1.10.5	Infinite Outcomes . . . . .	44
<b>2</b>	<b>Graphical Models</b>	<b>45</b>
<b>3</b>	<b>Learning a Probabilistic Model from Data</b>	<b>46</b>
<b>A</b>	<b>Notation Summary</b>	<b>47</b>

# Chapter 1

## Probability and Inference

### 1.1 Introduction to Probability

#### 1.1.1 Introduction

Probabilities appear in everyday life and feed into how we make decisions. For example:

The weather forecast might say that “tomorrow there is a 70% chance of rain”. This 70% chance of rain is a probability, and if it is sufficiently high, then we may want to bring an umbrella when we go outdoors.

We could predict that the probability of car traffic is higher during rush hour than otherwise, so if we don’t want to be stuck in traffic while driving, we should avoid driving during rush hour.

We aim to build computer programs that can reason with probabilities.

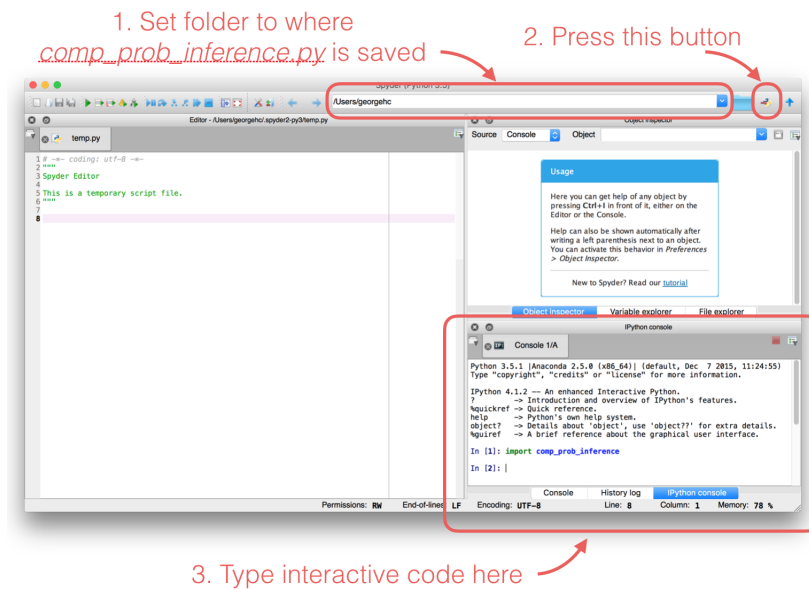
#### 1.1.2 A First Look at Probability

Perhaps the simplest example of probability is flipping a fair coin for which we say that the probability of heads is  $1/2$  and, similarly, the probability of tails is also  $1/2$ . (Don’t worry, we’ll see much more exciting problems soon!) What do we mean when we say that the probability of heads is  $1/2$ ?

The basic idea is that if we repeat this experiment of flipping a coin a huge number of times, say  $n$ , then the number of heads we should see should be close to  $n/2$  as we increase  $n$ . While you could certainly try this out in real life by flipping a coin, say, 100,000 times, doing this would be disastrously tedious. Let’s simulate these flips in Python instead.

**Simulating Coin Flips** Follow along in an IPython prompt within Spyder.

We have provided a package `comp_prob_inference.py`, which you should save to your computer. Within Spyder, do the following:



Let's start by importing the package `comp_prob_inference`:

```
> import comp_prob_inference
```

To simulate flipping a fair coin, enter:

```
> comp_prob_inference.flip_fair_coin()
```

You should get either 'heads' or 'tails'. Try re-running the above line a few times. You should see that the coin flip results are random.

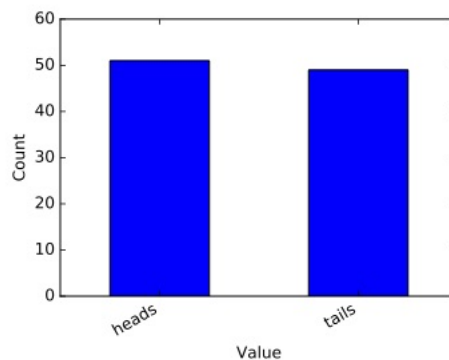
To flip the fair coin 100 times, enter:

```
> flips = comp_prob_inference.flip_fair_coins(100)
```

Let's plot how many times we see the two possible outcomes in the same bar graph, called a histogram:

```
> comp_prob_inference.plot_discrete_histogram(flips)
```

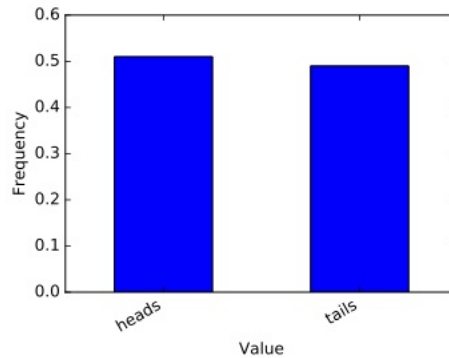
For example, we get the following plot:



Often what we will care about in this course is the fraction (also called the frequency of times an outcome happens). To plot the fraction of times heads or tails occurred, we again use the `plot_discrete_histogram` function but now add the keyword argument `frequency=True`:

```
> comp_prob_inference.plot_discrete_histogram(flips, frequency=True)
```

Doing so, we get the following plot:



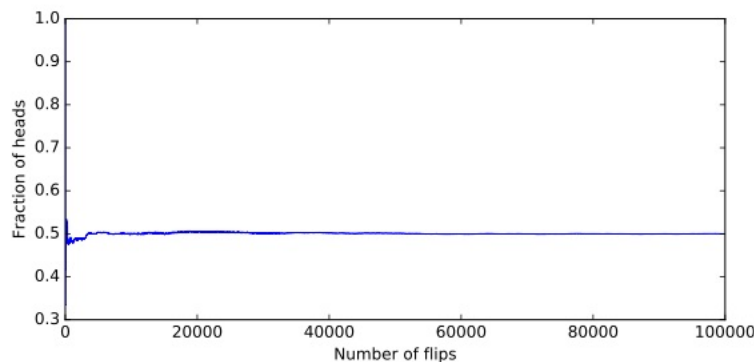
Next, let's plot the fraction of heads as a function of the number of flips (going up to 100,000 flips).

```
n = 100000
heads_so_far = 0
fraction_of_heads = []
for i in range(n):
    if comp_prob_inference.flip_fair_coin() == 'heads':
        heads_so_far += 1
    fraction_of_heads.append(heads_so_far / (i+1))
```

Note that `fraction_of_heads[i]` tells us what the fraction of heads is after the first  $i$  tosses. Then to actually plot the fraction of heads vs the number of tosses, enter the following:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 4))
plt.plot(range(1, n+1), fraction_of_heads)
plt.xlabel('Number of flips')
plt.ylabel('Fraction of heads')
```

For example, when we run this we get the following plot:



The fraction of heads initially can be far from  $1/2$  but as the number of flips increases, the fraction stabilizes and gets closer to  $1/2$ , the probability of heads.

**Computer note:** Many times in this course, it will be helpful to run simulations to test code and plot histograms for different outcomes to get a sense of how likely the outcomes are. Simulations and visualizations can be powerful not only in making sure your code is working correctly but also to present results to people!

### 1.1.3 Probability and the Art of Modeling Uncertainty

Since probability effectively corresponds to a fraction, it is a value between 0 and 1. Of course, we can have impossible events that have probability 0, or events that deterministically happen and thus have probability 1. Each time we model uncertainty in the world, there will be some underlying experiment (such as flipping a coin in our running example). An event happens with probability  $q \in [0, 1]$  if in a massive number of repeats of the experiment, the event happens roughly a fraction  $q$  of the time; more repeats of the experiment make it so that the fraction gets closer to  $q$ .

Some times, an underlying experiment cannot possibly be repeated. Take for instance weather forecasting. Whereas we could actually physically flip a coin many times to repeat the same experiment, we cannot physically repeat a real-life experiment for what different realizations of tomorrow's weather will be. We could wait until tomorrow to see the weather, but then we would need a time machine to go back in time by one day to repeat and see what the weather is like tomorrow (and this assumes that there's some inherent randomness in tomorrow's weather)! In such a case, our only hope is to somehow model or simulate tomorrow's weather given measurements up to present time.

Different people could model the same real world problem differently! Throughout the course, a recurring challenge in building computer programs that reason probabilistically is figuring out how to model real-world problems. A good model — even if not actually accurate in describing, for instance, the science behind weather — enables us to make good predictions.

Once a weather forecaster has anchored some way of modeling or simulating weather, then if it claims that there's a 30% chance of rain tomorrow, we could interpret this as saying that using their way of simulating tomorrow's weather, in roughly 30% of simulated results for tomorrow's weather, there is rain.

## 1.2 Probability Spaces and Events

### 1.2.1 Two Ingredients to Modeling Uncertainty

When we think of an uncertain world, we will always think of there being some underlying experiment of interest. To model this uncertain world, it suffices to keep track of two things:

The set of all possible outcomes for the experiment: this set is called the sample space and is usually denoted by the Greek letter Omega  $\Omega$ . (For the fair coin flip, there are exactly two possible outcomes: heads, tails. Thus,  $\Omega = \{\text{heads}, \text{tails}\}$ .)

The probability of each outcome: for each possible outcome, assign a probability that is at least 0 and at most 1. (For the fair coin flip,  $\mathbb{P}(\text{heads}) = \frac{1}{2}$  and  $\mathbb{P}(\text{tails}) = \frac{1}{2}$ .)

Notation: Throughout this course, for any statement  $\mathcal{S}$ , “ $\mathbb{P}(\mathcal{S})$ ” denotes the probability of  $\mathcal{S}$  happening.

In Python:

```
> model = {'heads': 1/2, 'tails': 1/2}
```

In particular, we see that we can model uncertainty in code using a Python dictionary. The sample space is precisely the keys in the dictionary:

```
> sample_space = set(model.keys())
{'tails', 'heads'}
```

Of course, the dictionary gives us the assignment of probabilities, meaning that for each outcome in the sample space (i.e., for each key in the dictionary), we have an assigned probability:

```
> model['heads']
0.5
```



```
> model['tails']
0.5
```

A few important remarks:

- The sample space is always specified to be *collectively exhaustive*, meaning that every possible outcome is in it, and *mutually exclusive*, meaning that once the experiment is run (e.g., flipping the fair coin), exactly one possible outcome in the sample space happens. It's impossible for multiple outcomes in the sample space to simultaneously happen! It's also impossible for none of the outcomes to happen!
- Probabilities can be thought of as fractions of times outcomes occur; thus, probabilities are nonnegative and at least 0 and at most 1.
- If we add up the probabilities of all the possible outcomes in the sample space, we get 1. (For the fair coin flip,  $\mathbb{P}(\text{heads}) + \mathbb{P}(\text{tails}) = \frac{1}{2} + \frac{1}{2} = 1$ .)

Some intuition for this: Consider the coin flipping experiment. What does the fraction of times heads occur and the fraction of times tails occur add up to? Since these are the only two possible outcomes (and again, recall that these outcomes are exclusive in that they can't simultaneously occur, and exhaustive since they are the only possible outcomes), these two fractions will always sum to 1. For a massive number of repeats of the experiment, these two fractions correspond to  $\mathbb{P}(\text{heads})$  and  $\mathbb{P}(\text{tails})$ ; the fractions sum to 1 and so these probabilities also sum to 1.

## 1.2.2 Probability Spaces

At this point, we've actually already seen the most basic data structure used throughout this course for modeling uncertainty, called a *finite probability space* (in this course, we'll often also just call this either a *probability space* or a *probability model*):

A *finite probability space* consists of two ingredients:

- a sample space  $\Omega$  consisting of a *finite* (i.e., not infinite) number of collectively exhaustive and mutually exclusive possible outcomes
- an assignment of probabilities: for each possible outcome  $\omega \in \Omega$ , we assign a probability  $\mathbb{P}(\text{outcome } \omega)$  at least 0 and at most 1, where we require that the probabilities across all the possible outcomes in the sample space add up to 1:

$$\sum_{\omega \in \Omega} \mathbb{P}(\text{outcome } \omega) = 1$$

**Notation:** As shorthand we occasionally use the tuple “ $(\Omega, \mathbb{P})$ ” to refer to a finite probability space to remind ourselves of the two ingredients needed, sample space  $\Omega$  and an assignment of probabilities  $\mathbb{P}$ . As we already saw, in code these two pieces can be represented together in a single Python dictionary. However, when we want to reason about probability spaces in terms of the mathematics, it's helpful to have names for the two pieces.

**Why finite?** Of the two pieces making up a finite probability space  $(\Omega, \mathbb{P})$ , the sample space  $\Omega$  being finite is a fairly natural constraint, corresponding to how we typically work with Python dictionaries where there is only a finite number of keys. As we'll see, finite probability spaces are already extremely useful in practice. Pedagogically, finite probability spaces also provide a great intro to probability theory as they already carry a wealth of intuition, much of which carries over to a more complete story of general probability spaces!

### 1.2.3 Table Representation

A probability space is a data structure in that we can always visualize as a table of nonnegative entries that sum to 1. Let's see a concrete example of this, first writing the table out on paper and then coding it up.

Example: Suppose we have a model of tomorrow's weather given as follows: sunny with probability  $1/2$ , rainy with probability  $1/6$ , and snowy with probability  $1/3$ . Here's the probability space, shown as a table:

	Probability	
Outcome	sunny	$1/2$
	rainy	$1/6$
	snowy	$1/3$

Note: This a table of 3 nonnegative entries that sum to 1. The rows correspond to the sample space  $\Omega = \{\text{sunny}, \text{rainy}, \text{snowy}\}$ .

We will often use this table representation of a probability space to tell you how we're modeling uncertainty for a particular problem. It provides the simplest of visualizations of a probability space.

Of course, in Python code, the above probability space is given by:

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

A different way to code up the same probability space is to separately specify the outcomes (i.e., the sample space) and the probabilities:

```
outcomes = ['sunny', 'rainy', 'snowy']
probabilities = np.array([1/2, 1/6, 1/3])
```

The  $i$ -th entry of `outcomes` has probability given by the  $i$ -th entry of `probabilities`. Note that `probabilities` is a vector of numbers that we represent as a Numpy array. Numpy has various built-in methods that enable us to easily work with vectors (and more generally arrays) of numbers.

### 1.2.4 More on Sample Spaces

In the video, we saw that a sample space encoding the outcomes of 2 coin flips encodes all the information for 1 coin flip as well. Thus, we could use the same sample space to model a single coin flip. However, if we really only cared about a single coin flip, then a sample space encoding 2 coin flips is richer than we actually need it to be!

When we model some uncertain situation, how we specify a sample space is not unique. We saw an example of this already in an earlier exercise where for rolling a single six-sided die, we can choose to name the outcomes differently, saying for instance "roll 1" instead of "1". We could even add a bunch of extraneous outcomes that all have probability 0. We could add extraneous information that doesn't matter such as "Alice rolls 1", "Bob rolls 1", etc where we enumerate out all the people who could roll the die in which the outcome is a 1. Sure, depending on the problem we are trying to solve, maybe knowing who rolled the die is important, but if we don't care about who rolled the die, then the information isn't helpful but it's still possible to include this information in the sample space.

Generally speaking it's best to choose a sample space that is as simple as possible for modeling what we care about solving. For example, if we were rolling a six-sided die, and we actually only care about whether the face shows up at least 4 or not, then it's sufficient to just keep track of two outcomes, "at least 4" and "less than 4".

### 1.2.5 Probabilities with Events

TODO – add notes from video

## 1.2.6 Events as Sets

TODO – add notes from video

## 1.2.7 Code for Dealing with Sets in Python

In the video, the set operations can actually be implemented in Python as follows:

```
sample_space = {'HH', 'HT', 'TH', 'TT'}
A = {'HT', 'TT'}
B = {'HH', 'HT', 'TH'}
C = {'HH'}
A_intersect_B = A.intersection(B) # equivalent to "B.intersection(A)" or "A & B"
A_union_C = A.union(C) # equivalent to "C.union(A)" and also "A | C"
B_complement = sample_space.difference(B) # equivalent also to "sample_space - B"
```

## 1.2.8 Probabilities with Events and Code

From the videos, we see that an event is a subset of the sample space  $\Omega$ . If you remember our table representation for a probability space, then an event could be thought of as a subset of the rows, and the probability of the event is just the sum of the probability values in those rows!

The probability of an event  $\mathcal{A} \subseteq \Omega$  is the sum of the probabilities of the possible outcomes in  $\mathcal{A}$ :

$$\mathbb{P}(\mathcal{A}) \triangleq \sum_{\omega \in \mathcal{A}} \mathbb{P}(\text{outcome } \omega),$$

where “ $\triangleq$ ” means “defined as”.

We can translate the above equation into Python code. In particular, we can compute the probability of an event encoded as a Python set event, where the probability space is encoded as a Python dictionary `prob_space`:

```
def prob_of_event(event, prob_space):
    total = 0
    for outcome in event:
        total += prob_space[outcome]
    return total
```

Here’s an example of how to use the above function:

```
prob_space = {'sunny': 1/2, 'rainy': 1/6, 'snowy': 1/3}
rainy_or_snowy_event = {'rainy', 'snowy'}
print(prob_of_event(rainy_or_snowy_event, prob_space))
```

## 1.3 Random Variables

### 1.3.1 A First Look at Random Variables

Follow along in an IPython prompt.

We continue with our weather example.

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

We can simulate tomorrow’s weather using the above model of the world. Let’s simulate two different values, one

(which we'll call  $W$  for “weather”) for whether tomorrow will be sunny, rainy, or snowy, and another (which we'll call  $I$  for “indicator”) that is 1 if it is sunny and 0 otherwise:

```
> random_outcome = comp_prob_inference.sample_from_finite_probability_space(prob_space)
> W = random_outcome
> if random_outcome == 'sunny':
>     I = 1
> else:
>     I = 0
```

Print out the variables  $W$  or  $I$  to see that they take on specific values. Then re-run the above block of code a few times.

You should see that  $W$  and  $I$  change and are random (following the probabilities given by the probability space).

This code shows something that's of key importance that we'll see throughout the course. Variables  $W$  and  $I$  store the values of what are called random variables.

### 1.3.2 Random Variables

To mathematically reason about a random variable, we need to somehow keep track of the full range of possibilities for what the random variable's value could be, and how probable different instantiations of the random variable are. The resulting formalism may at first seem a bit odd but as we progress through the course, it will become more apparent how this formalism helps us study real-world problems and address these problems with powerful solutions.

To build up to the formalism, first note, computationally, what happened in the code in the previous part.

1. First, there is an underlying probability space  $(\Omega, \mathbb{P})$ , where  $\Omega = \{\text{sunny, rainy, snowy}\}$ , and

$$\begin{aligned}\mathbb{P}(\text{sunny}) &= 1/2, \\ \mathbb{P}(\text{rainy}) &= 1/6, \\ \mathbb{P}(\text{snowy}) &= 1/3.\end{aligned}$$

2. A random outcome  $\omega \in \Omega$  is sampled using the probabilities given by the probability space  $(\Omega, \mathbb{P})$ . This step corresponds to an underlying experiment happening.
3. Two random variables are generated:

- $W$  is set to be equal to  $\omega$ . As an equation:

$$W(\omega) = \omega \quad \text{for } \omega \in \{\text{sunny, rainy, snowy}\}.$$

This step perhaps seems entirely unnecessary, as you might wonder “Why not just call the random outcome  $W$  instead of  $\omega$ ?” Indeed, this step isn't actually necessary for this particular example, but the formalism for random variables has this step to deal with what happens when we encounter a random variable like  $I$ .

- $I$  is set to 1 if  $\omega = \text{sunny}$ , and 0 otherwise. As an equation:

$$I(\omega) = \begin{cases} 1 & \text{if } \omega = \text{sunny}, \\ 0 & \text{if } \omega \in \{\text{rainy, snowy}\}. \end{cases}$$

Importantly, multiple possible outcomes (rainy or snowy) get mapped to the same value 0 that  $I$  can take on.

We see that random variable  $W$  maps the sample space  $\Omega = \{\text{sunny, rainy, snowy}\}$  to the same set  $\{\text{sunny, rainy, snowy}\}$ . Meanwhile, random variable  $I$  maps the sample space  $\Omega = \{\text{sunny, rainy, snowy}\}$  to the set  $\{0, 1\}$ .

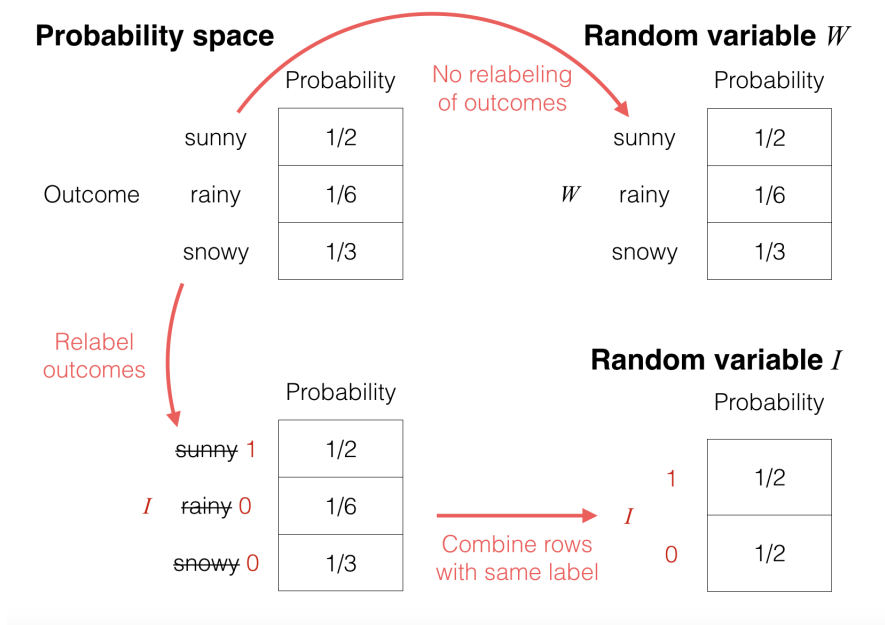
In general:

**Definition of a “finite random variable” (in this course, we will just call this a “random variable”):** Given a finite probability space  $(\Omega, \mathbb{P})$ , a finite random variable  $X$  is a mapping from the sample space  $\Omega$  to a set of values  $X$  that random variable  $X$  can take on. (We will often call  $X$  the “alphabet” of random variable  $X$ .)

For example, random variable  $W$  takes on values in the alphabet sunny,rainy,snowy, and random variable  $I$  takes on values in the alphabet  $\{0, 1\}$ .

**Quick summary:** There’s an underlying experiment corresponding to probability space  $(\Omega, \mathbb{P})$ . Once the experiment is run, let  $\omega \in \Omega$  denote the outcome of the experiment. Then the random variable takes on the specific value of  $X(\omega) \in \mathcal{X}$ .

**Explanation using a picture:** Continuing with the weather example, we can pictorially see what’s going on by looking at the probability tables for: the original probability space, the random variable  $W$ , and the random variable  $I$ :



These tables make it clear that a “random variable” really is just reassigning/relabeling what the values are for the possible outcomes in the underlying probability space (given by the top left table):

- In the top right table, random variable  $W$  does not do any sort of relabeling so its probability table looks the same as that of the underlying probability space.
- In the bottom left table, the random variable  $I$  relabels/reassigns “sunny” to 1, and both “rainy” and “snowy” to 0. Intuitively, since two of the rows now have the same label 0, it makes sense to just combine these two rows, adding their probabilities  $\frac{1}{6} + \frac{1}{3} = \frac{1}{2}$ . This results in the bottom right table.

**Technical note:** Even though the formal definition of a finite random variable doesn’t actually make use of the probability assignment  $\mathbb{P}$ , the probability assignment will become essential as soon as we talk about how probability works with random variables.

### 1.3.3 Random Variables Notation and Terminology

In this course, we denote random variables with capital/uppercase letters, such as  $X$ ,  $W$ ,  $I$ , etc. We use the phrases “probability table”, “probability mass function” (abbreviated as PMF), and “probability distribution” (often simply called a distribution) to mean the same thing, and in particular we denote the probability table for  $X$  to be  $p_X$  or  $p_X(\cdot)$ .

We write  $p_X(x)$  to denote the entry of the probability table that has label  $x \in \mathcal{X}$  where  $\mathcal{X}$  is the set of values that random variable  $\mathcal{X}$  takes on. Note that we use lowercase letters like  $x$  to denote variables storing nonrandom values. We can also look up values in a probability table using specific outcomes, e.g., from earlier, we have  $p_W(\text{rainy}) = 1/6$  and  $p_I(1) = 1/2$ .

Note that we use the same notation as in math where a function  $f$  might also be written as  $f(\cdot)$  to explicitly indicate that it is the function of one variable. Both  $f$  and  $f(\cdot)$  refer to a function whereas  $f(x)$  refers to the value of the function  $f$  evaluated at the point  $x$ .

As an example of how to use all this notation, recall that a probability table consists of nonnegative entries that add up to 1. In fact, each of the entries is at most 1 (otherwise the numbers would add to more than 1). For a random variable  $X$  taking on values in  $\mathcal{X}$ , we can write out these constraints as:

$$0 \leq p_X(x) \leq 1 \quad \text{for all } x \in \mathcal{X}, \quad \sum_{x \in \mathcal{X}} p_X(x) = 1.$$

Often in the course, if we are making statements about all possible outcomes of  $X$ , we will omit writing out the alphabet  $\mathcal{X}$  explicitly. For example, instead of the above, we might write the following equivalent statement:

$$0 \leq p_X(x) \leq 1 \quad \text{for all } x, \quad \sum_x p_X(x) = 1.$$

## 1.4 Jointly Distributed Random Variables

### 1.4.1 Relating Two Random Variables

At the most basic level, inference refers to using an observation to reason about some unknown quantity. In this course, the observation and the unknown quantity are represented by random variables. The main modeling question is: How do these random variables relate?

Let’s build on our earlier weather example, where now another outcome of interest appears, the temperature, which we quantize into two possible values “hot” and “cold”. Let’s suppose that we have the following probability space:

		Probability
Outcome	sunny, hot	3/10
	sunny, cold	1/5
	rainy, hot	1/30
	rainy, cold	2/15
	snowy, hot	0
	snowy, cold	1/3

You can check that the nonnegative entries do add to 1. If we let random variable  $W$  be the weather (sunny, rainy, snowy) and random variable  $T$  be the temperature (hot, cold), then notice that we could rearrange the table in the following fashion:

		Probability
Outcome	$W = \text{sunny}, T = \text{hot}$	$3/10$
	$W = \text{sunny}, T = \text{cold}$	$1/5$
	$W = \text{rainy}, T = \text{hot}$	$1/30$
	$W = \text{rainy}, T = \text{cold}$	$2/15$
	$W = \text{snowy}, T = \text{hot}$	$0$
	$W = \text{snowy}, T = \text{cold}$	$1/3$

Rearrange  
table entries

↓

		$T$	
		hot	cold
$W$	sunny	$3/10$	$1/5$
	rainy	$1/30$	$2/15$
	snowy	$0$	$1/3$

When we talk about two separate random variables, we could view them either as a single “super” random variable that happens to consist of a pair of values (the first table; notice the label for each outcome corresponds to a pair of values), or we can view the two separate variables along their own different axes (the second table).

The first table tells us what the underlying probability space is, which includes what the sample space is (just read off the outcome names) and what the probability is for each of the possible outcomes for the underlying experiment at hand.

The second table is called a joint probability table  $p_{W,T}$  for random variables  $W$  and  $T$ , and we say that random variables  $W$  and  $T$  are jointly distributed with the above distribution. Since this table is a rearrangement of the earlier table, it also consists of nonnegative entries that add to 1.

The joint probability table gives probabilities in which  $W$  and  $T$  co-occur with specific values. For example, in the above, the event that “ $W = \text{sunny}$ ” and the event that “ $T = \text{hot}$ ” co-occur with probability  $3/10$ . Notationally, we write

$$p_{W,T}(\text{sunny}, \text{hot}) = \mathbb{P}(W = \text{sunny}, T = \text{hot}) = \frac{3}{10}.$$

**Conceptual note:** Given the joint probability table, we can easily go backwards and write out the first table above, which is the underlying probability space.

**Preview of inference:** Inference is all about answering questions like “if we observe that the weather is rainy, what is the probability that the temperature is cold?” Let’s take a look at how one might answer this question.

First, if we observe that it is rainy, then we know that “sunny” and “snowy” didn’t happen so those rows aren’t relevant anymore. So the space of possible realizations of the world has shrunk to two options now: ( $W = \text{rainy}, T = \text{hot}$ ) or ( $W = \text{rainy}, T = \text{cold}$ ). But what about the probabilities of these two realizations? It’s not just  $1/30$  and  $2/15$  since these don’t sum to 1 — by observing things, adjustments can be made to the probabilities of different realizations but they should still form a valid probability space.

Why not just scale both  $1/30$  and  $2/15$  by the same constant so that they sum to 1? This can be done by dividing  $1/30$  and  $2/15$  by their sum:

$$\text{hot: } \frac{\frac{1}{30}}{\frac{1}{30} + \frac{2}{15}} = \frac{1}{5}, \quad \text{cold: } \frac{\frac{2}{15}}{\frac{1}{30} + \frac{2}{15}} = \frac{4}{5}.$$

Now they sum to 1. It turns out that, given that we’ve observed the weather to be rainy, these are the correct probabilities for the two options “hot” and “cold”. Let’s formalize the steps. We work backwards, first explaining

what the the denominator “ $\frac{1}{30} + \frac{2}{15} = \frac{1}{6}$ ” above comes from.

## 1.4.2 Representing a Joint Probability Table in Code

There are various ways to represent a joint probability table in code. Here are a few.

Note that we have updated `comp_prob_inference.py`! Please re-download it!

**Approach 0:** Don’t actually represent the joint probability table. This doesn’t store the 2D table at all but is a first attempt at coding up something that has all the information in the joint probability table. Specifically, we can just code up the entries like how we coded up a probability space:

```
> prob_table = {('sunny', 'hot'): 3/10,
>               ('sunny', 'cold'): 1/5,
>               ('rainy', 'hot'): 1/30,
>               ('rainy', 'cold'): 2/15,
>               ('snowy', 'hot'): 0,
>               ('snowy', 'cold'): 1/3}
```

Thus, if we want the probability of  $W = \text{rainy}$  and  $T = \text{cold}$ , we write:

```
> prob_table[('rainy', 'cold')]
0.13333333333333333
```

Some times, this representation is sufficient. Given a specific weather and temperature stored as strings in `w` and `t` respectively, `prob_table[(w, t)]` gives you the joint probability table evaluated at  $W = w$  and  $T = t$ .

**Approach 1:** Use dictionaries within a dictionary. This works as follows:

```
> prob_W_T_dict = {}
> for w in {'sunny', 'rainy', 'snowy'}:
>     prob_W_T_dict[w] = {}
>
> prob_W_T_dict['sunny']['hot'] = 3/10
> prob_W_T_dict['sunny']['cold'] = 1/5
> prob_W_T_dict['rainy']['hot'] = 1/30
> prob_W_T_dict['rainy']['cold'] = 2/15
> prob_W_T_dict['snowy']['hot'] = 0
> prob_W_T_dict['snowy']['cold'] = 1/3
>
> comp_prob_inference.print_joint_prob_table_dict(prob_W_T_dict)
      cold      hot
rainy 0.133333 0.033333
snowy 0.333333 0.000000
sunny 0.200000 0.300000
```

Note that because dictionary keys aren’t ordered, the row ordering and column ordering need not match the tables we have been showing in the course notes earlier. This is not a problem.

If we want the probability of  $W = \text{rainy}$  and  $T = \text{cold}$ , we write:

```
> prob_W_T_dict['rainy']['cold']
0.13333333333333333
```

The probability for  $W = w$  and  $T = t$  is stored in `prob_W_T_dict[w][t]`.



**Approach 2:** Use a 2D array. Another approach is to directly store the joint probability table as a 2D array, separately keeping track of what the rows and columns are. We use a NumPy array (but really you could use Python lists within a Python list, much like how the previous approach used dictionaries within a dictionary; the indexing syntax changes only slightly):

```
> import numpy as np
> prob_W_T_rows = ['sunny', 'rainy', 'snowy']
> prob_W_T_cols = ['hot', 'cold']
> prob_W_T_array = np.array([[3/10, 1/5], [1/30, 2/15], [0, 1/3]])
> comp_prob_inference.print_joint_prob_table_array(prob_W_T_array, prob_W_T_rows, prob_W_T_cols)
      hot      cold
sunny 0.300000 0.200000
rainy 0.033333 0.133333
snowy 0.000000 0.333333
```

Note that the ordering of rows is specified, as is the ordering of the columns, unlike in the dictionaries within a dictionary representation.

Retrieving a specific table entry requires a little bit more code since we need to figure out what the row and column numbers are corresponding to a specific pair of row and column labels. For example, if we want the probability of  $W = \text{rainy}$  and  $T = \text{cold}$ , we write:

```
> prob_W_T_array[prob_W_T_rows.index('rainy'), prob_W_T_cols.index('cold')]
0.13333333333333333
```

Note that `prob_W_T_rows.index('rainy')` finds the row number (starting from 0) corresponding to “rainy”.

Using `.index` does a search through the whole list of row/column labels, which for large lists can be slow. Let’s fix this!

A cleaner and faster way is to create separate dictionaries mapping the row and column labels to row and column indices in the 2D array. In other words, instead of writing `prob_W_T_rows.index('rainy')` to find the row number for ‘rainy’, we want to just be able to write something like `prob_W_T_row_mapping['rainy']`, which returns the row number. We can define Python variable `prob_W_T_row_mapping` as follows:

```
> prob_W_T_row_mapping = {}
> for index, label in enumerate(prob_W_T_rows):
>     prob_W_T_row_mapping[label] = index
```

Note that `enumerate(prob_W_T_rows)` produces an iterator that consists of pairs (0, ‘sunny’), (1, ‘rainy’), (2, ‘snowy’). You can see this by entering:

```
> print(list(enumerate(prob_W_T_rows)))
[(0, 'sunny'), (1, 'rainy'), (2, 'snowy')]
```

Note that each pair consists of the row number followed by the label.

In fact, the three lines we used to define `prob_W_T_row_mapping` can be written in one line with a Python dictionary comprehension:

```
> prob_W_T_row_mapping = {label: index for index, label in enumerate(prob_W_T_rows)}
```

We can do the same thing to define a mapping of column labels to column numbers:

```
> prob_W_T_col_mapping = {label: index for index, label in enumerate(prob_W_T_cols)}
```

In summary, we can represent the joint probability table as follows:

```
> prob_W_T_rows = ['sunny', 'rainy', 'snowy']
```

```
> prob_W_T_cols = ['hot', 'cold']
> prob_W_T_row_mapping = {label: index for index, label in enumerate(prob_W_T_rows)}
> prob_W_T_col_mapping = {label: index for index, label in enumerate(prob_W_T_cols)}
> prob_W_T_array = np.array([[3/10, 1/5], [1/30, 2/15], [0, 1/3]])
```

Now the probability for  $W = w$  and  $T = t$  is given by:

```
> prob_W_T_array[prob_W_T_row_mapping[w], prob_W_T_col_mapping[t]]
```

**Some remarks:** The 2D array representation, as we'll see soon, is very easy to work with when it comes to basic operations like summing rows, and retrieving a specific row or column. The main disadvantage of this representation is that you need to store the whole array, and if the alphabet sizes of the random variables are very large, then storing the array will take a lot of space!

The dictionaries within a dictionary representation allows for easily retrieving rows but not columns (try it: write a Python function that picks out a specific row and another function that picks out a specific column; you should see that retrieving a row is easier because it corresponds to looking at the value stored for a single key of the outer dictionary). This also means that summing a column's probabilities is more cumbersome than summing a row's probabilities. However, a huge advantage of this way of representing a joint probability table is that in many problems, we have a massive joint probability table that is mostly filled with 0's. Thus, the 2D array representation would require storing a very, very large table with many 0's, whereas the dictionaries within a dictionary representation is able to only store the nonzero table entries. We'll see more about this issue when we look at robot localization in the second section of the course on inference in graphical models.

### 1.4.3 Marginalization

Given a joint probability table, often we'll want to know what the probability table is for just one of the random variables. We can do this by just summing or "marginalizing" out the other random variables. For example, to get the probability table for random variable  $W$ , we do the following:

		$T$							
		hot	cold					Prob.	
$W$	sunny	3/10	1/5	1/2	→	sunny	1/2		
	rainy	1/30	2/15	1/6		rainy	1/6		
	snowy	0	1/3	1/3		snowy	1/3		
		Add up each row				Numbers in right margin form table $p_W$			

We take the joint probability table (left-hand side) and compute out the row sums (which we've written in the margin).

The right-hand side table is the probability table  $p_W$  for random variable  $W$ ; we call this resulting probability distribution the marginal distribution of  $W$  (put another way, it is the distribution obtained by marginalizing out the random variables that aren't  $W$ ).

In terms of notation, the above marginalization procedure whereby we used the joint distribution of  $W$  and  $T$  to produce the marginal distribution of  $W$  is written:

$$p_W(w) = \sum_{t \in \mathcal{T}} p_{W,T}(w, t),$$

where  $\mathcal{T}$  is the set of values that random variable  $T$  can take on. In fact, throughout this course, we will often omit explicitly writing out the alphabet of values that a random variable takes on, e.g., writing instead



$$p_W(w) = \sum_t p_{W,T}(w, t).$$

It's clear from context that we're summing over all possible values for  $t$ , which is going to be the values that random variable  $T$  can possibly take on.

As a specific example,

$$p_W(\text{rainy}) = \sum_t p_{W,T}(\text{rainy}, t) = \underbrace{p_{W,T}(\text{rainy}, \text{hot})}_{1/30} + \underbrace{p_{W,T}(\text{rainy}, \text{cold})}_{2/15} = \frac{1}{6}.$$

We could similarly marginalize out random variable  $W$  to get the marginal distribution  $p_T$  for random variable  $T$ :

		$T$		
		hot	cold	
$W$	sunny	3/10	1/5	 Add up each column
	rainy	1/30	2/15	
	snowy	0	1/3	
		1/3	2/3	
		 Numbers in bottom margin form table $p_T$		
		$T$		
		hot	cold	
Prob.		1/3	2/3	

(Note that whether we write a probability table for a single variable horizontally or vertically doesn't actually matter.)

As a formula, we would write:

$$p_T(t) = \sum_w p_{W,T}(w, t).$$

For example,

$$p_T(\text{hot}) = \sum_w p_{W,T}(w, \text{hot}) = \underbrace{p_{W,T}(\text{sunny}, \text{hot})}_{3/10} + \underbrace{p_{W,T}(\text{rainy}, \text{hot})}_{1/30} + \underbrace{p_{W,T}(\text{snowy}, \text{hot})}_0 = \frac{1}{3}.$$

In general:

**Marginalization:** Consider two random variables  $X$  and  $Y$  (that take on values in the sets  $\mathcal{X}$  and  $\mathcal{Y}$  respectively) with joint probability table  $p_{X,Y}$ . For any  $x \in \mathcal{X}$ , the marginal probability that  $X = x$  is given by

$$p_X(x) = \sum_y p_{X,Y}(x, y).$$

#### 1.4.4 Marginalization for Many Random Variables

What happens when we have more than two random variables? Let's build on our earlier example and suppose that in addition to weather  $W$  and temperature  $T$ , we also had a random variable  $H$  for humidity that takes on values in the alphabet dry, humid. Then having a third random variable, we can draw out a 3D joint probability table for random variables  $W$ ,  $T$ , and  $H$ . As an example, we could have the following:

		<i>T</i>		
		hot cold		
<i>W</i>	sunny	1/5	1/10	dry humid <i>H</i>
	rainy	1/10	1/10	
	snowy	1/30	2/15	
		0	2/9	1/9

Here, each of the cubes/boxes stores a probability. Not visible are two of the cubes in the back left column, which for this particular example both have probability values of 0.

Then to marginalize out the humidity *H*, we would add values as follows:

		<i>T</i>		
		hot cold		
<i>W</i>	sunny	1/5	1/10	dry humid <i>H</i>
	rainy	1/10	1/10	
	snowy	1/30	2/15	
		0	2/9	1/9

Add along this direction to  
marginalize out humidity

		<i>T</i>		
		hot cold		
<i>W</i>	sunny	3/10	1/5	
	rainy	1/30	2/15	
	snowy	0	1/3	

The resulting numbers  
form the table  $p_{W,T}$

The result is the joint probability table for weather *W* and temperature *T*, shown still in 3D cubes with each cube storing a single probability.

As an equation:

$$p_{W,T}(w,t) = \sum_h p_{W,T,H}(w,t,h).$$

In general, for three random variables *X*, *Y*, and *Z* with joint probability table  $p_{X,Y,Z}$ , we have

$$\begin{aligned} p_{X,Y}(x,y) &= \sum_z p_{X,Y,Z}(x,y,z), \\ p_{X,Z}(x,z) &= \sum_y p_{X,Y,Z}(x,y,z), \\ p_{Y,Z}(y,z) &= \sum_x p_{X,Y,Z}(x,y,z). \end{aligned}$$

Note that we can marginalize out different random variables in succession. For example, given joint probability table  $p_{X,Y,Z}$ , if we wanted the probability table  $p_X$ , we can get it by marginalizing out the two random variables *Y* and *Z*:

$$p_X(x) = \sum_y p_{X,Y}(x,y) = \sum_y \left( \sum_z p_{X,Y,Z}(x,y,z) \right).$$

Even with more than three random variables, the idea is the same. For example, with four random variables *W*, *X*,

$Y$ , and  $Z$  with joint probability table  $p_{W,X,Y,Z}$ , if we want the joint probability table for  $X$  and  $Y$ , we would do the following:

$$p_{X,Y}(x,y) = \sum_w \left( \sum_z p_{W,X,Y,Z}(w,x,y,z) \right).$$

### 1.4.5 Conditioning for Random Variables

When we observe that a random variable takes on a specific value (such as  $W = \text{rainy}$  from earlier for which we say that we condition on random variable  $W$  taking on the value “rainy”), this observation can affect what we think are likely or unlikely values for another random variable.

When we condition on  $W = \text{rainy}$ , we do a two-step procedure; first, we only keep the row for  $W$  corresponding to the observed value:

	$T$			$T$	
	hot	cold		hot	cold
W sunny	3/10	1/5	Keep row for $W = \text{rainy}$ →		
W rainy	1/30	2/15		W rainy	1/30    2/15
W snowy	0	1/3			
					Not valid prob. distribution (since sum $\neq 1$ )

Second, we “normalize” the table so that its entries add up to 1, which corresponds to dividing it by the sum of the entries, which is equal to  $p_W(\text{rainy})$  in this case:

	$T$			$T$	
	hot	cold		hot	cold
W rainy	1/30	2/15	Rescale entries so they add to 1 →	W rainy	1/5    4/5

Notation: The resulting probability table  $p_{T|W}(\cdot | \text{rainy})$  is associated with the random variable denoted  $(T | W = \text{rainy})$ ; we use “|” to denote that we’re conditioning on things to the right of “|” happening (these are things that we have observed or that we are given as having happened). We read “ $T | W = \text{rainy}$ ” as either “ $T$  given  $W$  is rainy” or “ $T$  conditioned on  $W$  being rainy”. To refer to specific entries of the table, we write, for instance,

$$p_{T|W}(\text{cold} | \text{rainy}) = \mathbb{P}(T = \text{cold} | W = \text{rainy}) = \frac{4}{5}.$$

In general:

**Conditioning:** Consider two random variables  $X$  and  $Y$  (that take on values in the sets  $\mathcal{X}$  and  $\mathcal{Y}$  respectively) with joint probability table  $p_{X,Y}$  (from which by marginalization we can readily compute the marginal probability table  $p_Y$ ). For any  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$  such that  $p_Y(y) > 0$ , the conditional probability of event  $X = x$  given event  $Y = y$  has happened is

$$p_{X|Y}(x | y) \triangleq \frac{p_{X,Y}(x,y)}{p_Y(y)}.$$

For example,

$$p_{T|W}(\text{cold} | \text{rainy}) = \frac{p_{W,T}(\text{rainy}, \text{cold})}{p_W(\text{rainy})} = \frac{\frac{2}{15}}{\frac{1}{6}} = \frac{4}{5}.$$

**Computational interpretation:** To compute  $p_{X|Y}(x | y)$ , take the entry  $p_{X,Y}(x,y)$  in the joint probability table corresponding to  $X = x$  and  $Y = y$ , and then divide the entry by  $p_Y(y)$ , which is an entry in the marginal probability table  $p_Y$  for random variable  $Y$ .

## 1.4.6 Moving Toward a More General Story for Conditioning

Jointly distributed random variables play a central role in this course. Remember that we will model observations as random variables and the quantities we want to infer also as random variables. When these random variables are jointly distributed so that we have a probabilistic way to describe how they relate (through their joint probability table), then we can systematically and quantitatively produce inferences.

We just saw how to condition on a random variable taking on a specific value. What about if we wanted to condition on a random variable taking on any one of many values rather than just one specific value? To answer this question, we look at a more general story of conditioning which is in terms of events.

## 1.5 Conditioning on Events

### 1.5.1 Conditioning on Events Intro

TODO – add notes from video

### 1.5.2 The Product Rule for Events

TODO – add notes from video

### 1.5.3 Bayes' Theorem for Events

**Important note about dividing by probabilities:** We will often divide by probabilities. In videos, we might not always say this, but this is required: we cannot divide by 0. To ensure this, we will not condition on events that have probability 0.

Given two events  $\mathcal{A}$  and  $\mathcal{B}$  (both of which have positive probability), Bayes' theorem, also called Bayes' rule or Bayes' law, gives a way to compute  $\mathbb{P}(\mathcal{A}|\mathcal{B})$  in terms of  $\mathbb{P}(\mathcal{B}|\mathcal{A})$ . This result turns out to be extremely useful for inference because often times we want to compute one of these, and the other is known or otherwise straightforward to compute.

Bayes' theorem is given by

$$\mathbb{P}(\mathcal{A}|\mathcal{B}) = \frac{\mathbb{P}(\mathcal{B}|\mathcal{A})\mathbb{P}(\mathcal{A})}{\mathbb{P}(\mathcal{B})}.$$

The proof of why this is the case is a one liner:

$$\mathbb{P}(\mathcal{A}|\mathcal{B}) \stackrel{(a)}{=} \frac{\mathbb{P}(\mathcal{A} \cap \mathcal{B})}{\mathbb{P}(\mathcal{B})} \stackrel{(b)}{=} \frac{\mathbb{P}(\mathcal{B}|\mathcal{A})\mathbb{P}(\mathcal{A})}{\mathbb{P}(\mathcal{B})},$$

where step (a) is by the definition of conditional probability for events, and step (b) is due to the product rule for events (which follows from rearranging the definition of conditional probability for  $\mathbb{P}(\mathcal{B}|\mathcal{A})$ ).

### 1.5.4 Practice Problem: Bayes' Theorem and Total Probability

Your problem set is due in 15 minutes! It's in one of your drawers, but they are messy, and you're not sure which one it's in.

The probability that the problem set is in drawer  $k$  is  $d_k$ . If drawer  $k$  has the problem set and you search there, you have probability  $p_k$  of finding it. There are a total of  $m$  drawers.

Suppose you search drawer  $i$  and do not find the problem set.

- (a) Find the probability that the paper is in drawer  $j$ , where  $j \neq i$ .

- (b) Find the probability that the paper is in drawer  $i$ .

**Solution:** Let  $A_k$  be the event that the problem set is in drawer  $k$ , and  $B_k$  be the event that you find the problem set in drawer  $k$ .

(a) We'll express the desired probability as  $\mathbb{P}(A_j|B_i^c)$ . Since this quantity is difficult to reason about directly, we'll use Bayes' rule:

$$\mathbb{P}(A_j|B_i^c) = \frac{\mathbb{P}(B_i^c|A_j)\mathbb{P}(A_j)}{\mathbb{P}(B_i^c)}$$

The first probability,  $\mathbb{P}(B_i^c|A_j)$ , expresses the probability of not finding the problem set in drawer  $i$  given that it's in a different drawer  $j$ . Since it's impossible to find the paper in a drawer it isn't in, this is just 1.

The second quantity,  $\mathbb{P}(A_j)$ , is given to us in the problem statement as  $d_j$ .

The third probability,  $\mathbb{P}(B_i^c) = 1 - \mathbb{P}(B_i)$ , is difficult to reason about directly. But, if we knew whether or not the paper was in the drawer, it would become easier. So, we'll use total probability:

$$\begin{aligned}\mathbb{P}(B_i) &= \mathbb{P}(B_i|A_i)\mathbb{P}(A_i) + \mathbb{P}(B_i|A_i^c)\mathbb{P}(A_i^c) \\ &= p_i d_i + 0(1 - d_i)\end{aligned}$$

Putting these terms together, we find that

$$\mathbb{P}(A_j|B_i^c) = \frac{d_j}{1 - p_i d_i}$$

**Alternate method to compute the denominator  $\mathbb{P}(B_i^c)$ :** We could use the law of total probability to decompose  $\mathbb{P}(B_i^c)$  depending on which drawer the homework is actually in. We have

$$\begin{aligned}\mathbb{P}(B_i^c) &= \sum_{k=1}^m \underbrace{\mathbb{P}(A_k)}_{d_k} \underbrace{\mathbb{P}(B_i^c|A_k)}_{\substack{1 \text{ if } k \neq i, \\ (1-p_i) \text{ if } k=i}} \\ &= \sum_{\substack{k=1, \\ k \neq i}}^m d_k + (1 - p_i)d_i \\ &= \sum_{k=1}^m d_k - p_i d_i \\ &= 1 - p_i d_i.\end{aligned}$$

(b) Similarly, we'll use Bayes' rule:

$$\mathbb{P}(A_i|B_i^c) = \frac{\mathbb{P}(B_i^c|A_i)\mathbb{P}(A_i)}{\mathbb{P}(B_i^c)} = \frac{(1-p_i)d_i}{1-p_i d_i}$$

### 1.5.5 Take-Away Lessons:

- Defining the sample-space is not always going to help solve the problem. (It's difficult to precisely define the sample space for this particular problem)
- When in doubt of being able to precisely define the sample space, try to define events intelligently, i.e., in a way that you use what you're given in the problem.
- The probability law of a probability model is a function on events, or subsets of the sample space, i.e., one can work with the probability law without knowing precisely what the sample-space (as a set) is.

## 1.6 Inference with Bayes' Theorem for Random Variables

### 1.6.1 The Product Rule for Random Variables (Also Called the Chain Rule)

In many real world problems, we aren't given what the joint distribution of two random variables is although we might be given other information from which we can compute the joint distribution. Often times, we can compute out the joint distribution using what's called the product rule (often also called the chain rule). This is precisely the random variable version of the product rule for events.

As we saw from before, we were able to derive Bayes' theorem for events using the product rule for events:  $\mathbb{P}(\mathcal{A} \cap \mathcal{B}) = \mathbb{P}(\mathcal{A})\mathbb{P}(\mathcal{B} | \mathcal{A})$ . The random variable version of the product rule is derived just like the event version of the product rule, by rearranging the equation for the definition of conditional probability. For two random variables  $X$  and  $Y$  (that take on values in sets  $\mathcal{X}$  and  $\mathcal{Y}$  respectively), the product rule for random variables says that

$$p_{X,Y}(x,y) = p_Y(y)p_{X|Y}(x|y) \quad \text{for all } x \in \mathcal{X}, y \in \mathcal{Y} \text{ such that } p_Y(y) > 0.$$

**Interpretation:** If we have the probability table for  $Y$ , and separately the probability table for  $X$  conditioned on  $Y$ , then we can come up with the joint probability table (i.e., the joint distribution) of  $X$  and  $Y$ .

What happens when  $p_Y(y) = 0$ ? Even though  $p_{X|Y}(x|y)$  isn't defined in this case, one can readily show that  $p_{X,Y}(x,y) = 0$  when  $p_Y(y) = 0$ .

To see this, think about what is happening computationally: Remember how  $p_Y(y)$  is computed from joint probability table  $p_{X,Y}$ ? In particular, we have  $p_Y(y) = \sum_x p_{X,Y}(x,y)$ , so  $p_Y(y)$  is the sum of either a row or a column in the joint probability table (whether it's a row or column just depends on how you write out the table and which random variable is along which axis along rows or columns). So if  $p_Y(y) = 0$ , it must mean that the individual elements being summed are 0 (since the numbers we're summing up are nonnegative).

We can formalize this intuition with a proof:

**Claim:** Suppose that random variables  $X$  and  $Y$  have joint probability table  $p_{X,Y}$  and take on values in sets  $\mathcal{X}$  and  $\mathcal{Y}$  respectively. Suppose that for a specific choice of  $y \in \mathcal{Y}$ , we have  $p_Y(y) = 0$ . Then

$$p_{X,Y}(x,y) = 0 \quad \text{for all } x \in \mathcal{X}.$$

**Proof:** Let  $y \in \mathcal{Y}$  satisfy  $p_Y(y) = 0$ . Recall that we relate marginal distribution  $p_Y$  to joint distribution  $p_{X,Y}$  via marginalization:

$$0 = p_Y(y) = \sum_{x \in \mathcal{X}} p_{X,Y}(x,y).$$

Next, we use a crucial mathematical observation: If a sum of nonnegative numbers (such as probabilities) equals 0, then each of the numbers being summed up must also be 0 (otherwise, the sum would be positive!). Hence, it must be that each number being added up in the right-hand side sum is 0, i.e.,

$$p_{X,Y}(x,y) = 0 \quad \text{for all } x \in \mathcal{X}.$$

This completes the proof.  $\square$

Thus, in general:

$$p_{X,Y}(x,y) = \begin{cases} p_Y(y)p_{X|Y}(x|y) & \text{if } p_Y(y) > 0, \\ 0 & \text{if } p_Y(y) = 0. \end{cases}$$

**Important convention for this course:** For notational convenience, throughout this course, we will often just write  $p_{X,Y}(x,y) = p_Y(y)p_{X|Y}(x|y)$  with the understanding that if  $p_Y(y) = 0$ , even though  $p_{X|Y}(x|y)$  is not actually



defined,  $p_{X,Y}(x,y)$  just evaluates to 0 anyways.

**The product rule is symmetric:** We can use the definition of conditional probability with  $X$  and  $Y$  swapped, and rearranging factors, we get:

$$p_{X,Y}(x,y) = p_X(x)p_{Y|X}(y|x) \quad \text{for all } x \in \mathcal{X}, y \in \mathcal{Y} \text{ such that } p_X(x) > 0,$$

and so similarly we could show that

$$p_{X,Y}(x,y) = \begin{cases} p_X(x)p_{Y|X}(y|x) & \text{if } p_X(x) > 0, \\ 0 & \text{if } p_X(x) = 0. \end{cases}$$

Again for notational convenience, we'll typically just write  $p_{X,Y}(x,y) = p_X(x)p_{Y|X}(y|x)$  with the understanding that the expression is 0 when  $p_Y(y) = 0$ .

**Interpretation:** If we're given the probability table for  $X$  and, separately, the probability table for  $Y$  conditioned on  $X$ , then we can come up with the joint probability table for  $X$  and  $Y$ .

Importantly, for any two jointly distributed random variables  $X$  and  $Y$ , the product rule is always true, without making any further assumptions! Also, as a recurring theme that we'll see later on as well, we are decomposing the joint distribution into the product of factors (in this case, the product of two factors).

**Many random variables:** If we have many random variables, say,  $X_1, X_2$ , up to  $X_N$  where  $N$  is not a random variable but is a fixed constant, then we have

$$\begin{aligned} & p_{X_1, X_2, \dots, X_N}(x_1, x_2, \dots, x_N) \\ &= p_{X_1}(x_1)p_{X_2|X_1}(x_2|x_1)p_{X_3|X_1, X_2}(x_3|x_1, x_2) \\ & \quad \cdots p_{X_N|X_1, X_2, \dots, X_{N-1}}(x_N|x_1, x_2, \dots, x_{N-1}). \end{aligned}$$

Again, we write this to mean that this holds for every possible choice of  $x_1, x_2, \dots, x_N$  for which we never condition on a zero probability event. Note that the above factorization always holds without additional assumptions on the distribution of  $X_1, X_2, \dots, X_N$ .

Note that the product rule could be applied in arbitrary orderings. In the above factorization, you could think of it as introducing random variable  $X_1$  first, and then  $X_2$ , and then  $X_3$ , etc. Each time we introduce another random variable, we have to condition on all the random variables that have already been introduced.

Since there are  $N$  random variables, there are  $N!$  different orderings in which we can write out the product rule. For example, we can think of introducing the last random variable  $X_N$  first and then going backwards until we introduce  $X_1$  at the end. This yields the, also correct, factorization

$$\begin{aligned} & p_{X_1, X_2, \dots, X_N}(x_1, x_2, \dots, x_N) \\ &= p_{X_N}(x_N)p_{X_{N-1}|X_N}(x_{N-1}|x_N)p_{X_{N-2}|X_{N-1}, X_N}(x_{N-2}|x_{N-1}, x_N) \\ & \quad \cdots p_{X_1|X_2, X_3, \dots, X_N}(x_1|x_2, \dots, x_N). \end{aligned}$$

## 1.6.2 Bayes' Rule for Random Variables (Also Called Bayes' Theorem for Random Variables)

In inference, what we want to reason about is some unknown random variable  $X$ , where we get to observe some other random variable  $Y$ , and we have some model for how  $X$  and  $Y$  relate. Specifically, suppose that we have some

“prior” distribution  $p_X$  for  $X$ ; this prior distribution encodes what we believe to be likely or unlikely values that  $X$  takes on, before we actually have any observations. We also suppose we have a “likelihood” distribution  $p_{Y|X}$ .

After observing that  $Y$  takes on a specific value  $y$ , our “belief” of what  $X$  given  $Y = y$  is now given by what’s called the “posterior” distribution  $p_{X|Y}(\cdot | y)$ . Put another way, we keep track of a probability distribution that tells us how plausible we think different values  $X$  can take on are. When we observe data  $Y$  that can help us reason about  $X$ , we proceed to either upweight or downweight how plausible we think different values  $X$  can take on are, making sure that we end up with a probability distribution giving us our updated belief of what  $X$  can be.

Thus, once we have observed  $Y = y$ , our belief of what  $X$  is changes from the prior  $p_X$  to the posterior  $p_{X|Y}(\cdot | y)$ .

Bayes’ theorem (also called Bayes’ rule or Bayes’ law) for random variables explicitly tells us how to compute the posterior distribution  $p_{X|Y}(\cdot | y)$ , i.e., how to weight each possible value that random variable  $X$  can take on, once we’ve observed  $Y = y$ . Bayes’ theorem is the main workhorse of numerous inference algorithms and will show up many times throughout the course.

**Bayes’ theorem:** Suppose that  $y$  is a value that random variable  $Y$  can take on, and  $p_Y(y) > 0$ . Then

$$p_{X|Y}(x | y) = \frac{p_X(x)p_{Y|X}(y|x)}{\sum_{x'} p_X(x')p_{Y|X}(y|x')}$$

for all values  $x$  that random variable  $X$  can take on.

**Important:** Remember that  $p_{X|Y}(\cdot | y)$  could be undefined but this isn’t an issue since this happens precisely when  $p_X(x) = 0$ , and we know that  $p_{X,Y}(x, y) = 0$  (for every  $y$ ) whenever  $p_X(x) = 0$ .

**Proof:** We have

$$p_{X|Y}(x | y) \stackrel{(a)}{=} \frac{p_{X,Y}(x, y)}{p_Y(y)} \stackrel{(b)}{=} \frac{p_X(x)p_{Y|X}(y|x)}{p_Y(y)} \stackrel{(c)}{=} \frac{p_X(x)p_{Y|X}(y|x)}{\sum_{x'} p_X(x')p_{Y|X}(y|x')} \stackrel{(d)}{=} \frac{p_X(x)p_{Y|X}(y|x)}{\sum_{x'} p_X(x')p_{Y|X}(y|x')},$$

where step (a) uses the definition of conditional probability (this step requires  $p_Y(y) > 0$ ), step (b) uses the product rule (recall that for notational convenience we’re not separately writing out the case when  $p_X(x) = 0$ ), step (c) uses the formula for marginalization, and step (d) uses the product rule (again, for notational convenience, we’re not separately writing out the case when  $p_X(x') = 0$ ).  $\square$

### 1.6.3 Bayes’ Theorem for Random Variables: A Computational View

Computationally, Bayes’ theorem can be thought of as a two-step procedure. Once we have observed  $Y = y$ :

For each value  $x$  that random variable  $X$  can take on, initially we believed that  $X = x$  with a score of  $p_X(x)$ , which could be thought of as how plausible we thought ahead of time that  $X = x$ . However now that we have observed  $Y = y$ , we weight the score  $p_X(x)$  by a factor  $p_{Y|X}(y | x)$ , so

$$\text{new belief for how plausible } X = x \text{ is: } \alpha(x | y) \triangleq p_X(x)p_{Y|X}(y | x),$$

where we have defined a new table  $\alpha(\cdot | y)$  which is not a probability table, since when we put in the weights, the new beliefs are no longer guaranteed to sum to 1 (i.e.,  $\sum_x \alpha(x | y)$  might not equal 1)!  $\alpha(\cdot | y)$  is an unnormalized posterior distribution!

Also, if  $p_X(x)$  is already 0, then as we already mentioned a few times,  $p_{Y|X}(y | x)$  is undefined, but this case isn’t a problem: no weighting is needed since an impossible outcome stays impossible.

To make things concrete, here is an example from the medical diagnosis problem where we observe  $Y = \text{positive}$ :

$p_X$		$p_{Y X}$		$X$	
healthy infected		healthy infected		healthy infected	
0.999	0.001	positive	0.01	0.99	
		negative	0.99	0.01	

entry-wise multiply to get  
unnormalized posterior

healthy infected	
0.00999	0.00099

We fix the fact that the unnormalized posterior table  $\alpha(\cdot | y)$  isn't guaranteed to sum to 1 by renormalizing:

$$p_{X|Y}(x | y) = \frac{\alpha(x|y)}{\sum_{x'} \alpha(x'|y)} = \frac{p_X(x)p_{Y|X}(y|x)}{\sum_{x'} p_X(x')p_{Y|X}(y|x')}.$$

An important note: Some times we won't actually care about doing this second renormalization step because we will only be interested in what value that  $X$  takes on is more plausible relative to others; while we could always do the renormalization, if we just want to see which value of  $x$  yields the highest entry in the unnormalized table  $\alpha(\cdot | y)$ , we could find this value of  $x$  without renormalizing!

## 1.6.4 Maximum A Posteriori (MAP) Estimation

For a hidden random variable  $X$  that we are inferring, and given observation  $Y = y$ , we have been talking about computing the posterior distribution  $p_{X|Y}(\cdot | y)$  using Bayes' rule. The posterior is a distribution for what we are inferring. Often times, we want to report which particular value of  $X$  actually achieves the highest posterior probability, i.e., the most probable value  $x$  that  $X$  can take on given that we have observed  $Y = y$ .

The value that  $X$  can take on that maximizes the posterior distribution is called the *maximum a posteriori* (MAP) estimate of  $X$  given  $Y = y$ . We denote the MAP estimate by  $\hat{x}_{\text{MAP}}(y)$ , where we make it clear that it depends on what the observed  $y$  is. Mathematically, we write

$$\hat{x}_{\text{MAP}}(y) = \arg \max_x p_{X|Y}(x|y).$$

Note that if we didn't include the "arg" before the "max", then we would just be finding the highest posterior probability rather than which value – or "argument" –  $x$  actually achieves the highest posterior probability.

In general, there could be ties, i.e., multiple values that  $X$  can take on are able to achieve the best possible posterior probability.

## 1.7 Independence Structure

### 1.7.1 Independent Events

When you flip a coin or roll dice, the outcome of a coin flip or a die roll isn't going to tell you anything about the outcome of a new coin toss or die roll unless you have some very peculiar coins or dice.

we're going to formalize this by saying that two events  $A$  and  $B$  are independent, which we'll denote by this thing that looks like an upside down T:  $\perp\!\!\!\perp$

$$A \perp\!\!\!\perp B \quad \text{if} \quad \mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$$

If  $\mathbb{P}(A) > 0$  we can use the product rule for events to rewrite the left side:

$$\mathbb{P}(B | A)\mathbb{P}(A) = \mathbb{P}(A)\mathbb{P}(B)$$

Cancelling  $\mathbb{P}(A)$  on both sides:

$$A \perp\!\!\!\perp B \quad \text{if} \quad \mathbb{P}(B | A) = \mathbb{P}(B)$$

Similarly, if  $\mathbb{P}(B) > 0$

$$A \perp\!\!\!\perp B \quad \text{if} \quad \mathbb{P}(A | B) = \mathbb{P}(A)$$

So knowing  $B$  doesn't tell us anything new about  $A$ , and *vice versa*.

## 1.7.2 Independent Random Variables

$X$  and  $Y$  are independent ( $X \perp\!\!\!\perp Y$ ) if  $p_{X,Y}(x,y) = p_X(x)p_Y(y)$ .

Knowing one gives no information about the other, so  $p_{X|Y}(x|y) = p_X(x)$ .

**Exercise: Independent Random Variables** In this exercise, we look at how to check if two random variables are independent in Python. Please make sure that you can follow the math for what's going on and be able to do this by hand as well.

Consider random variables  $W$ ,  $I$ ,  $X$ , and  $Y$ , where we have shown the joint probability tables  $p_{W,I}$  and  $p_{X,Y}$ .

		$I$				$Y$	
		1	0			1	0
$W$	sunny	1/2	0	$X$	sunny	1/4	1/4
	rainy	0	1/6		rainy	1/12	1/12
	snowy	0	1/3		snowy	1/6	1/6

In Python:

```
prob_W_I = np.array([[1/2, 0], [0, 1/6], [0, 1/3]])
```

Note that here, we are not explicitly storing the labels, but we'll keep track of them in our heads. The labels for the rows (in order of row index): sunny, rainy, snowy. The labels for the columns (in order of column index): 1, 0.

We can get the marginal distributions  $p_W$  and  $p_I$ :

```
prob_W = prob_W_I.sum(axis=1)
prob_I = prob_W_I.sum(axis=0)
```

Then if  $W$  and  $I$  were actually independent, then just from their marginal distributions  $p_W$  and  $p_I$ , we would be able to compute the joint distribution with the formula:

$$\text{If } W \text{ and } I \text{ are independent:} \quad p_{W,I}(w,i) = p_W(w)p_I(i) \quad \text{for all } w,i.$$

Note that variables `prob_W` and `prob_I` at this point store the probability tables  $p_W$  and  $p_I$  as 1D NumPy arrays, for which NumPy does *not* store whether each of these should be represented as a row or as a column.

We could however ask NumPy to treat them as column vectors, and in particular, taking the outer product of `prob_W` and `prob_I` yields what the joint distribution would be if  $W$  and  $I$  were independent:

$$\begin{bmatrix} p_W(\text{sunny}) \\ p_W(\text{rainy}) \\ p_W(\text{snowy}) \end{bmatrix} \begin{bmatrix} p_I(1) & p_I(0) \end{bmatrix} = \begin{bmatrix} p_W(\text{sunny})p_I(1) & p_W(\text{sunny})p_I(0) \\ p_W(\text{rainy})p_I(1) & p_W(\text{rainy})p_I(0) \\ p_W(\text{snowy})p_I(1) & p_W(\text{snowy})p_I(0) \end{bmatrix}.$$

The left-hand side is an outer product, and the right-hand side is precisely the joint probability table that would result if  $W$  and  $I$  were independent.

To compute and print the right-hand side, we do:

```
print(np.outer(prob_W, prob_I))
```

### 1.7.3 Mutual vs Pairwise Independence

To extend the independence story to more than two variables, the strongest way is by using something called “mutual independence”. We’ll say that three random variables  $X$ ,  $Y$ , and  $Z$  are mutually independent if we can write the joint distribution as simply the product of the three individual distributions:

$$p_{X,Y,Z}(x,y,z) = p_X(x)p_Y(y)p_Z(z)$$

Knowing  $X$  and  $Y$  won’t tell you anything about  $Z$ . Knowing  $X$  won’t tell you anything about  $Y$ . They’re completely independent.

There are also weaker forms of independence that we’re interested in, for example, “pairwise independence”. This means that for any two variables, you can write:

$$p_{X,Y}(x,y) = p_X(x)p_Y(y)$$

and similarly for  $Y,Z$  and  $X,Z$ . This is saying if I know any one, it doesn’t tell me think anything about any of the others. But this is not the same as mutual independence, it’s not as strong.

As an example of why, suppose that we have two random variables  $X$  and  $Y$ . They both represent independent fair coin flips. We’ll write the outcomes as 0 and 1, and each one has a 50-50 chance of being heads or tails, 0 or 1.

We’ll define  $Z = X \oplus Y$ , where “XOR” (written  $\oplus$ ) is defined to be a function that takes in two things and returns 1 when exactly one of them is 1:

$x$	$y$	$z$
0	0	0
0	1	1
1	0	1
1	1	0

What’s the probability  $p_{X,Y}$  for each of these configurations? They’re independent fair coin flips, so each is equally likely. The probability of any particular one is 0.5 for  $X$  times 0.5 for  $Y$ , so 0.25:

$p_{X,Y}$	$x$	$y$	$z$
0.25	0	0	0
0.25	0	1	1
0.25	1	0	1
0.25	1	1	0

What’s the distribution for  $Z$ ? There are two ways to get  $z = 0$  and they each have probability 0.25. If we add them up then we have a 0.5 chance of  $z$  being 0 and similarly a 0.5 chance of  $z$  being 1.

$$p_Z(z) = \begin{cases} 0.5 & \text{if } z = 0 \\ 0.5 & \text{if } z = 1 \end{cases}$$

What’s  $Z$  given  $X$ ? Well, it’s actually the same. If  $x = 0$ , then we can restrict ourselves to just looking at the top two rows of the table, so  $Z$  can either be 0 or 1. And, again, they’re equally likely so it’s 50-50. If  $x = 1$ , we can just look at the bottom two rows, and again they’re equally likely, 50-50.

$$p_{Z|X}(z|x) = \begin{cases} 0.5 & \text{if } z = 0 \\ 0.5 & \text{if } z = 1 \end{cases}$$

So  $p_{Z|X}(z|x) = p_Z(z)$ , it doesn't depend on  $X$  at all. So this means that  $Z \perp\!\!\!\perp X$ .

By symmetry, we can make the same argument for  $Z \perp\!\!\!\perp Y$ , and we said at the start that  $X \perp\!\!\!\perp Y$ . So here we have three random variables that are all pairwise independent — if you look at any pair, knowing one doesn't tell you about the other.

But if we look all together, they're not mutually independent. For example, if I know any two of them, then I know exactly what the third one is going to be. The distribution over the third one changes from being fair 50-50 to being deterministic. For example, if  $X$  is 0 and  $Y$  is 0, then  $Z$  is also going to be 0. And if I didn't know that, then the probability would be 50-50. Once I do know that, the probability becomes 1 that it's 0 and 0 that it's anything else. So here, they're not mutually independent. In this example it may seem a little contrived, but in general, it's often tempting to assume that knowing things are pairwise independent tells you that they're mutually independent. But you have to be aware that they're not.

### 1.7.4 Conditional Independence

We say that two random variables  $X$  and  $Y$  are conditionally independent given the third random variable  $Z$  if we can write the conditional distribution for both of them as the product of the individual conditionals:

$$p_{X,Y|Z}(x,y|z) = p_{X|Z}(x|z)p_{Y|Z}(y|z)$$

Intuitively, this means that once we know  $Z$ , then knowing something about  $Y$  doesn't tell you anything about  $X$ , and *vice versa*.

Notice that marginal independence and conditional independence are not the same thing. So if you have marginal independence, then that does not necessarily imply conditional independence. And the reverse is also not true. So two random variables  $X$  and  $Y$  could be independent but not conditionally independent given something else. Or two random variables could be conditionally independent but not marginally independent given something else.

As an example that illustrates this, suppose we have three random variables  $R$ ,  $S$ , and  $T$ . We'll ask two questions: (a) is  $S \perp\!\!\!\perp T$ ? (b) is  $S \perp\!\!\!\perp T | R$ ? There are two different ways of writing the joint distribution:

$$\begin{aligned} (a) p_{R,S,T}(r,s,t) &= p_R(r)p_{S|R}(s|r)p_{T|R}(t|r) \\ (b) p_{R,S,T}(r,s,t) &= p_S(s)p_T(t)p_{R|S,T}(r|s,t) \end{aligned}$$

**Part (a):** To determine whether two things are independent we just write out the distribution and see if it factors — that is, see if we can write it as something that only depends on  $S$  times something that only depends on  $T$ . If we can, then the former is  $p_S(s)$ , the latter is  $p_T(t)$ .

To compute the distribution of  $p_{S,T}(s,t)$  from  $p_{R,S,T}(r,s,t)$ , we just have to sum over every possible value of  $r$ :

$$\begin{aligned} p_{S,T}(s,t) &= \sum_r p_{R,S,T}(r,s,t) \\ &= \sum_r p_R(r)p_{S|R}(s|r)p_{T|R}(t|r) \\ &\neq p_S(s)p_T(t) \end{aligned}$$

In general, we will not be able to factor the result because it will be a sum of different things that depend on  $S$  and  $T$  in different ways. So therefore,  $S$  and  $T$  are not independent.

Are they conditionally independent given  $R$ ? To compute that, we have to see whether  $p_{S,T}(s,t) | R = p_{S|R}(s | r)p_{T|R}(t | r)$ . To compute a conditional distribution, remember that we just have to divide by the distribution of whatever we're conditioning on:

$$\begin{aligned} p_{S,T|R}(s,t | r) &= \frac{p_{R,S,T}(r,s,t)}{p_R(r)} \\ &= \frac{p_R(r)p_{S|R}(s | r)p_{T|R}(t | r)}{p_R(r)} \\ &= p_{S|R}(s | r)p_{T|R}(t | r) \end{aligned}$$

Canceling  $p_R(r)$  leaves us with exactly the two terms we want. So the joint conditional distribution becomes the product of the two individual conditionals. So  $S$  and  $T$  are, in fact, conditionally independent given  $R$ .

**Part (b):** We have that

$$p_{R,S,T}(r,s,t) = p_S(s)p_T(t)p_{R|S,T}(r | s,t)$$

As in part (a), if we want to compute whether  $S$  and  $T$  are independent, we have to look at the joint distribution and see whether we can write it as a product of the two individual distributions.

Again, to compute the joint distribution we have sum over every possible  $r$ :

$$\begin{aligned} p_{S,T}(s,t) &= \sum_r p_S(s)p_T(t)p_{R|S,T}(r | s,t) \\ &= p_S(s)p_T(t) \sum_r p_{R|S,T}(r | s,t) \\ &= p_S(s)p_T(t) \end{aligned}$$

As  $p_S(s)$  and  $p_T(t)$  don't depend on  $r$  they can be factored outside the summation. But no matter what  $S$  and  $T$  are, if we sum over every possible value of  $r$ , then  $\sum_r p_{R|S,T}(r | s,t)$  is just going to give us 1 because it is just a probability distribution over  $R$ , and all probability distributions have to sum to 1. Hence  $p_{S,T}(s,t) = p_S(s)p_T(t)$ , and that's exactly the condition for independence. So in this case,  $S \perp\!\!\!\perp T$ .

What about conditional independence? As in part (a), we have to look at the conditional distribution  $p_{S,T|R}(s,t | r)$  and see whether it factors.

$$p_{S,T|R}(s,t | r) = \frac{p_S(s)p_T(t)p_{R|S,T}(r | s,t)}{p_R(r)}$$

If we try to factor this, we'll see that we run into trouble when we get to the term  $p_{R|S,T}(r | s,t)p_R(r)$  because in general, this is not going to factor. We don't know how  $S$  and  $T$  interact here, and there are no simplifications we can do to make this term go away. So here,  $S \not\perp\!\!\!\perp T | R$ .

We've seen from the last two examples that sometimes we can have marginal independence without conditional independence, and that sometimes we can have conditional independence without marginal independence. So it's important to keep in mind that these two are not always the same thing, and knowing one doesn't necessarily tell you the other.

### 1.7.5 Explaining Away

Let's look at an example with conditional probability. Suppose we have three events,  $R$ ,  $A$ , and  $T$ , where

- $R$  is the event that a Red Sox game
- $A$  is the event there's an accident downtown
- $T$  is the event there is unusually bad traffic

The chance of having a game and the chance of having an accident are each independently 50:50 :

$$p_R(r) = \begin{cases} 0.5 & r = 1 \\ 0.5 & r = 0. \end{cases}$$

$$p_A(a) = \begin{cases} 0.5 & a = 1 \\ 0.5 & a = 0. \end{cases}$$

Whether or not there is traffic depends on whether there's a game and whether there's an accident. This table shows the probability that  $T$  is 1 for each configuration of  $r$  and  $a$ :

$r$	$a$	$p_{T R,A}(1   r, a)$
0	0	0.3
0	1	0.9
1	0	0.9
1	1	0.3

So with no game and no accident, the probability of having bad traffic is 0.3. With either or both, then the probability goes up to 0.9. The table shows us the distribution for  $T = 1$ . The distribution for  $T = 0$  would just have 1 minus this in every entry.

We want to calculate three probabilities:

- (a)  $\mathbb{P}(R = 1)$
- (b)  $\mathbb{P}(R = 1 | T = 1)$
- (b)  $\mathbb{P}(R = 1 | T = 1, A = 1)$

Part (a) is easy, as  $\mathbb{P}(R = 1)$  is given to us as 0.5. But in (b) and (c), we're asked to find information about  $R$  given  $T$ , whereas the problem setup gives us information about  $T$  in terms of  $R$ . So we'll use Bayes' rule to turn the conditioning around. In both cases, we're only interested in the case where  $T = 1$ , so we'll not worry about the full distribution for now.

Bayes' rule tells us that

$$p_{R,A|T}(r, a | 1) = \frac{p_{T|R,A}(1|r,a)p_R(r)p_A(a)}{p_T(1)}$$

If we look carefully, we'll see that, in this case, because  $R$  and  $A$  are independent 50-50,  $p_R(r)$  and  $p_A(a)$  are going to be the same for every combination of  $r$  and  $a$ . Similarly, the denominator is also going to be the same for every  $r$  and  $a$ . So we can take a shortcut and say that these terms together are equal to some constant  $C$ :

$$p_{R,A|T}(r, a | 1) = C \cdot p_{T|R,A}(1 | r, a)$$

The table for  $p_{T|R,A}(1 | r, a)$  is shown above, and we know that  $p_{R,A|T}(r, a | 1)$  is some constant times this table. The probabilities are 0.3, 0.9, 0.9, 0.9. Except we know that they have to sum to 1. So if we take everything and divide by their sum, we get the normalized distribution of  $r$  and  $a$  given  $t = 1$ .

		$a$	
		0	1
$r$	0	0.1	0.3
	1	0.3	0.3



In part (b), we're asking for  $\mathbb{P}(R = 1 \mid T = 1)$ . So in this case, we don't care about  $a$ , we only care about  $r$  being 1 and we're marginalizing over  $A$ . So from the bottom row of the normalized distribution,  $\mathbb{P}(R = 1 \mid T = 1) = 0.3 + 0.3 = 0.6$ .

In part (b), we're asking for  $\mathbb{P}(R = 1 \mid T = 1, A = 1)$ . If we condition on  $a = 1$ , then we're restricting ourselves to the right column:

		<b>a</b>
		1
<b>r</b>	0	0.3
	1	0.3

The probability that  $r = 1$  is 0.3 divided by the marginal probability of the column, that is 0.3 divided by 0.6. It occurs half the time because it's equally likely to be 0 or 1. So here our probability goes back down to 0.5. Intuitively, why is this? Why does our probability change? When we had more information the first time, in (b),  $T = 1$ , our probability went up from 0.5 to 0.6. But when we added even more information, then it went back down.

If we go back and look at the scenario, what this is asking is only given that there is traffic, what's the probability that there was a game? Well, it's a little higher in (b) because we know games cause traffic. So if there was traffic, then it's reasonable to assume that there was a game. So the probability goes up. But we know that both games and accidents cause traffic. So if we know that there was traffic and we know there was an accident, in (c), then it's more likely that the traffic was caused by the accident. So this is called explaining away, where once we observe one explanation, that is the accident, our belief in a different explanation, the Red Sox game, goes back down.

### 1.7.6 Practice Problem: Conditional Independence

Suppose  $X_0, \dots, X_{100}$  are random variables whose joint distribution has the following factorization:

$$p_{X_0, \dots, X_{100}}(x_0, \dots, x_{100}) = p_{X_0}(x_0) \cdot \prod_{i=1}^{100} p_{X_i|X_{i-1}}(x_i|x_{i-1})$$

This factorization is what's called a Markov chain. We'll be seeing Markov chains a lot more later on in the course.

Show that  $X_{50} \perp X_{52} | X_{51}$ .

Solution: Notice that we can marginalize out  $x_{100}$  as such:

$$p_{X_0, \dots, X_{99}}(x_0, \dots, x_{99}) = p_{X_0}(x_0) \cdot \prod_{i=1}^{99} p_{X_i|X_{i-1}}(x_i|x_{i-1}) \cdot \underbrace{\sum_{x_{100}} p_{X_{100}|X_{99}}(x_{100}|x_{99})}_{=1} \quad (2.5)$$

Now we can repeat the same marginalization procedure to get:

$$p_{X_0, \dots, X_{50}}(x_0, \dots, x_{50}) = p_{X_0}(x_0) \cdot \prod_{i=1}^{50} p_{X_i|X_{i-1}}(x_i|x_{i-1})$$

In essence, we have shown that the given joint distribution factorization applies not just to the last random variable ( $X_{100}$ ), but also up to any point in the chain.

For brevity, we will now use  $p(x_i^j)$  as a shorthand for  $p_{X_i, \dots, X_j}(x_i, \dots, x_j)$ . We want to exploit what we have shown to rewrite  $p(x_{50}^{52})$

$$\begin{aligned}
p(x_{50}^{52}) &= \sum_{x_0 \dots x_{49}} \sum_{x_{53} \dots x_{100}} p(x_0^{100}) \\
&= \sum_{x_0 \dots x_{49}} \sum_{x_{53} \dots x_{100}} \left[ p(x_0) \prod_{i=0}^{50} p(x_i | x_{i-1}) \right] \cdot p(x_{51} | x_{50}) \cdot p(x_{52} | x_{51}) \cdot \prod_{i=53}^{100} p(x_i | x_{i-1}) \\
&= \sum_{x_0 \dots x_{49}} \sum_{x_{53} \dots x_{100}} p(x_0^{50}) \cdot p(x_{51} | x_{50}) \cdot p(x_{52} | x_{51}) \cdot \prod_{i=53}^{100} p(x_i | x_{i-1}) \\
&= p(x_{51} | x_{50}) \cdot p(x_{52} | x_{51}) \cdot \sum_{x_0 \dots x_{49}} p(x_0^{50}) \underbrace{\sum_{x_{53} \dots x_{100}} \prod_{i=53}^{100} p(x_i | x_{i-1})}_{=1} \\
&= p(x_{51} | x_{50}) \cdot p(x_{52} | x_{51}) \cdot \sum_{x_0 \dots x_{49}} p(x_0^{50}) \\
&= p(x_{50}) \cdot p(x_{51} | x_{50}) \cdot p(x_{52} | x_{51})
\end{aligned}$$

where we used (2.5) for the 3rd equality and the same marginalization trick for the 5th equality. We have just shown the Markov chain property, so the conditional independence property must be satisfied.

## 1.8 Decisions and Expectations

### 1.8.1 Introduction to Decision Making and Expectations

We now know the basics of working with probabilities. But how do we incorporate probabilities into making decisions?

Let's make this concrete. Suppose we are given the option of playing one of the following lotteries. Which one should we play?

- Lottery 1: Pay \$1 and have a one in one million chance of winning \$1000.
- Lottery 2: Pay \$1 and have a one in one million chance of winning \$1000000.
- Lottery 3: Pay \$1 and have a one in ten chance of winning \$10.

Of course, there isn't a right or wrong answer here, but what we would like to do is come up with some quantitative way to make a decision. Especially if we want to make decisions automatically based on huge amount of observations, a principled quantitative approach is crucial!

One way to go about making a decision is to compute a score for each of the possible choices and choose the decision with the highest score. In this case of choosing between the three lotteries, for each of the lotteries, we could compute some kind of "average" amount of winnings, accounting for the cost to play.

For example, if we play the first lottery, we definitely lose \$1, and then there is a  $\frac{1}{1000000}$  chance that we win \$1,000. So one way we could write out an "average" amount of winnings is something like

$$-1 + 1000 \cdot \frac{1}{1000000} = -1 + 0.001 = -0.999.$$

For now, multiplying the \$1,000 by  $\frac{1}{1000000}$  can be thought of as a heuristic that signifies that we aren't guaranteed to win \$1,000, and how much we multiply by we're just picking to be the probability of winning right now.

Using the above way we have just come up with for computing "average" winnings, the second lottery has average winnings

$$-1 + \frac{1}{1000000}(1000000) = -1 + 1 = 0.$$

That's not so bad right? The chance of winning is still really low but the average winnings according to this calculation is 0, so it seems like we stand nothing to lose right?

Well, it depends on how much uncertainty we're willing to tolerate. A one in a million chance seems really low so almost always we would just lose a dollar if we play lottery #2.

In the third lottery, the average winnings is

$$-1 + \frac{1}{10}(10) = 0,$$

the same as for lottery #2. Sure, we aren't going to win \$1000000 in this game but the chance of winning is way higher: 1/10 instead of 1/1000000. Somehow that should matter right?

On the basis of the average winnings calculation we have done though, lotteries #2 and #3 would be equally good as they give the same highest average winnings of \$0.

We now discuss how to quantitatively and rigorously reason about scenarios like the ones we've just sketched. The main tool we now introduce is what's called the expected value of a random variable. In making decisions that account for randomness, it often makes sense to account for an "average" scenario that we should expect. Expectation is about taking an average, accounting for how likely different outcomes are.

In 6.008.1x, after we cover our story here on expected values, we won't be seeing them again until the third part of the course on learning probabilistic models. The idea there is that what probabilistic model makes sense for your data can be thought of as decision making! We are deciding which model to use instead of, in our example here, deciding which lottery to play!

## 1.8.2 The Expected Value of a Random Variable

Consider, for example, the mean of three values: 3, 5, and 10. It can be computed as follows:

$$\frac{3+5+10}{3} = 3 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 10 \cdot \frac{1}{3} = 6.$$

Notice, on the right-hand side, that we are adding 3, 5, and 10 each weighted by  $\frac{1}{3}$ . Concretely, consider a random variable  $X$  given by the probability table below:

	Probability
3	1/3
5	1/3
10	1/3

Then the "expected value" of  $X$  is given by

$$3 \cdot p_X(3) + 5 \cdot p_X(5) + 10 \cdot p_X(10) = 3 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 10 \cdot \frac{1}{3} = \frac{18}{3} = 6 \dots$$

But what if, for instance, we think that 3 is actually much more plausible than 5 or 10? Then what we could do is have the weight on 3 be higher than  $\frac{1}{3}$  while decreasing the weights for 5 and 10. Consider if instead we had:

	Probability
3	2/3
5	1/6
10	1/6

Then the expected value of  $X$  is given by

$$3 \cdot p_X(3) + 5 \cdot p_X(5) + 10 \cdot p_X(10) = 3 \cdot \frac{2}{3} + 5 \cdot \frac{1}{6} + 10 \cdot \frac{1}{6} = \frac{18}{3} = 6 \dots$$

Using probability, we now formalize the concept of expected value of a random variable. As you can see, all we are doing is taking the sum of the labels in the probability table, where we weight each label by the probability of the

label. *Importantly, the labels are numbers so that it's clear what adding them means!*

Now, for the formal definition:

**Definition of expected value:** Consider a real-valued random variable  $X$  that takes on values in a set  $\mathcal{X}$ . Then the expected value of  $X$ , denoted as  $\mathbb{E}[X]$ , is

$$\mathbb{E}[X] \triangleq \sum_{x \in \mathcal{X}} x \cdot p_X(x).$$

Having the random variable be real-valued makes it so that we can add up the labels with weights!

Also, note that whereas  $X$  can be represented as a probability table, its expectation  $\mathbb{E}[X]$  is just a single number. The expected value is the sum of the values in the set  $\mathcal{X}$ , weighted by the probabilities of each of the values. The mean is simply the expected value when all of the values in the set  $\mathcal{X}$  when there is a uniform probability of each of the values.

Notice that how we came up with the expectation of a random variable  $X$  just relied on the probability table for  $X$ .

In fact, if we took a different probability table, if the labels are numbers, then we can still compute the expectation! Two important examples are below.

### *Conditional Expectation*

As a first example, suppose we have two random variables  $X$  and  $Y$  where we know (or we have already computed)  $p_{X|Y}(\cdot | y)$  for some fixed value  $y$ , and  $X$  is real-valued. Then we can readily compute the expectation for this probability table by multiplying each value  $x$  in the alphabet of random variable  $X$  by  $p_{X|Y}(x | y)$  and summing these up to get a weighted average. This yields what is called the conditional expectation of  $X$  given  $Y = y$ , denoted as

$$\mathbb{E}[X | Y = y] = \sum_{x \in \mathcal{X}} x \cdot p_{X|Y}(x | y).$$

### *Expectation of the Function of a Random Variable*

As another example, suppose we have a (possibly not real-valued) random variable  $X$  with probability table  $p_X$ , and we have a function  $f$  such that  $f(x)$  is real-valued for all  $x$  in the alphabet  $\mathcal{X}$  of  $X$ . Then  $f(X)$  has a probability table where the labels are all numbers, and so we can compute  $\mathbb{E}[f(X)]$ .

Let's work out the math here. First, let's determine the probability table for  $f(X)$ . To make the notation here easier to parse, let random variable  $Z = f(X)$ . Note that  $Z$  has alphabet  $\mathcal{Z} = \{f(x) : x \in \mathcal{X}\}$ . Then the probability table for  $f(X)$  can be written as  $p_Z$ . In terms of the probability table, to compute  $p_Z(z)$ , we first look at every label in table  $p_X$  that gets mapped to  $z$ , i.e., the set  $\{x \in \mathcal{X} : f(x) = z\}$ . Then we sum up the probabilities of these labels to get the probability that  $Z = z$ , i.e.,  $p_Z(z) = \sum_{x \in \mathcal{X} \text{ such that } f(x)=z} p_X(x)$ .

We introduce a new piece of notation here called an indicator function  $\mathbf{1}\{\cdot\}$  that takes as input a statement  $\mathcal{S}$  and outputs:

$$\mathbf{1}\{\mathcal{S}\} = \begin{cases} 1 & \text{if } \mathcal{S} \text{ happens,} \\ 0 & \text{otherwise.} \end{cases}$$

Then the probability that  $Z = z$  can be written

$$\begin{aligned} p_Z(z) &= \sum_{x \in \mathcal{X} \text{ such that } f(x)=z} p_X(x) \\ &= \sum_{x \in \mathcal{X}} \mathbf{1}\{f(x) = z\} p_X(x). \end{aligned}$$

Next, we compute the expectation of  $Z = f(X)$ :

$$\begin{aligned}
\mathbb{E}[Z] &= \sum_{z \in \mathcal{Z}} z p_Z(z) \\
&= \sum_{z \in \mathcal{Z}} z \left[ \sum_{x \in \mathcal{X}} \mathbf{1}\{f(x) = z\} p_X(x) \right] \\
&= \sum_{x \in \mathcal{X}} \underbrace{\sum_{z \in \mathcal{Z}} z \mathbf{1}\{f(x) = z\} p_X(x)}_{\text{there is only 1 nonzero term here: when } z=f(x)} \\
&= \sum_{x \in \mathcal{X}} f(x) p_X(x).
\end{aligned}$$

Hence, since  $Z = f(X)$ , we can write

$$\mathbb{E}[f(X)] = \sum_{x \in \mathcal{X}} f(x) p_X(x).$$

### 1.8.3 Variance and Standard Deviation

The variance of a real-valued random variable  $X$  is defined as

$$\text{var}(X) \triangleq \mathbb{E}[(X - \mathbb{E}[X])^2].$$

Note that as we saw previously,  $\mathbb{E}[X]$  is just a single number. To keep the variance of  $X$ , what you could do is first compute the expectation of  $X$ .

For example, if  $X$  takes on each of the values 3, 5, and 10 with equal probability  $1/3$ , then first we compute  $\mathbb{E}[X]$  to get 6, and then we compute  $\mathbb{E}[(X - 6)^2]$ , where we remember to use the result that for a function  $f$ , if  $f(X)$  is a real-valued random variable, then  $\mathbb{E}[f(X)] = \sum_x f(x) p_X(x)$ . Here,  $f$  is given by  $f(x) = (x - 6)^2$ . So

$$\text{var}(X) = (3 - 6)^2 \cdot \frac{1}{3} + (5 - 6)^2 \cdot \frac{1}{3} + (10 - 6)^2 \cdot \frac{1}{3} = \frac{26}{3}.$$

### 1.8.4 Practice Problem: The Law of Total Expectation

Remember the law of total probability? For a set of events  $\mathcal{B}_1, \dots, \mathcal{B}_n$  that partition the sample space  $\Omega$  (so the  $\mathcal{B}_i$ 's don't overlap and together they fully cover the full space of possible outcomes),

$$\mathbb{P}(\mathcal{A}) = \sum_{i=1}^n \mathbb{P}(\mathcal{A} \cap \mathcal{B}_i) = \sum_{i=1}^n \mathbb{P}(\mathcal{A} \mid \mathcal{B}_i) \mathbb{P}(\mathcal{B}_i),$$

where the second equality uses the product rule.

A similar statement is true for the expected value of a random variable, called the law of total expectation: for a random variable  $X$  (with alphabet  $\mathcal{X}$ ) and a partition  $\mathcal{B}_1, \dots, \mathcal{B}_n$  of the sample space,

$$\mathbb{E}[X] = \sum_{i=1}^n \mathbb{E}[X \mid \mathcal{B}_i] \mathbb{P}(\mathcal{B}_i),$$

where

$$\mathbb{E}[X \mid \mathcal{B}_i] = \sum_{x \in \mathcal{X}} x p_{X \mid \mathcal{B}_i}(x) = \sum_{x \in \mathcal{X}} x \frac{\mathbb{P}(X=x, \mathcal{B}_i)}{\mathbb{P}(\mathcal{B}_i)}.$$

We will be using this result in the section “Towards Infinity in Modeling Uncertainty”.

Show that the law of total expectation is true.

**Solution:** There are different ways to prove the law of total expectation. We take a fairly direct approach here, first writing everything in terms of outcomes in the sample space.

The main technical hurdle is that the events  $\mathcal{B}_1, \dots, \mathcal{B}_n$  are specified directly in the sample space, whereas working with values that  $X$  takes on requires mapping from the sample space to the alphabet of  $X$ .

We will derive the law of total expectation starting from the right-hand side of the equation above, i.e.,  $\sum_{i=1}^n \mathbb{E}[X | \mathcal{B}_i] \mathbb{P}(\mathcal{B}_i)$ .

We first write  $\mathbb{E}[X | \mathcal{B}_i]$  in terms of a summation over outcomes in  $\Omega$ :

$$\begin{aligned}
 \mathbb{E}[X | \mathcal{B}_i] &= \sum_{x \in \mathcal{X}} x \frac{\mathbb{P}(X=x, \mathcal{B}_i)}{\mathbb{P}(\mathcal{B}_i)} \\
 &= \sum_{x \in \mathcal{X}} x \frac{\mathbb{P}(\{\omega \in \Omega : X(\omega)=x\} \cap \mathcal{B}_i)}{\mathbb{P}(\mathcal{B}_i)} \\
 &= \sum_{x \in \mathcal{X}} x \frac{\mathbb{P}(\{\omega \in \Omega : X(\omega)=x \text{ and } \omega \in \mathcal{B}_i\})}{\mathbb{P}(\mathcal{B}_i)} \\
 &= \sum_{x \in \mathcal{X}} x \frac{\mathbb{P}(\{\omega \in \mathcal{B}_i : X(\omega)=x\})}{\mathbb{P}(\mathcal{B}_i)} \\
 &= \sum_{x \in \mathcal{X}} x \cdot \frac{\sum_{\omega \in \mathcal{B}_i \text{ such that } X(\omega)=x} \mathbb{P}(\{\omega\})}{\mathbb{P}(\mathcal{B}_i)} \\
 &= \frac{1}{\mathbb{P}(\mathcal{B}_i)} \sum_{x \in \mathcal{X}} x \sum_{\omega \in \mathcal{B}_i \text{ such that } X(\omega)=x} \mathbb{P}(\{\omega\}) \\
 &= \frac{1}{\mathbb{P}(\mathcal{B}_i)} \sum_{\omega \in \mathcal{B}_i} X(\omega) \mathbb{P}(\{\omega\}).
 \end{aligned}$$

Thus,

$$\begin{aligned}
 \sum_{i=1}^n \mathbb{E}[X | \mathcal{B}_i] \mathbb{P}(\mathcal{B}_i) &= \sum_{i=1}^n \left( \frac{1}{\mathbb{P}(\mathcal{B}_i)} \sum_{\omega \in \mathcal{B}_i} X(\omega) \mathbb{P}(\{\omega\}) \right) \mathbb{P}(\mathcal{B}_i) \\
 &= \sum_{i=1}^n \sum_{\omega \in \mathcal{B}_i} X(\omega) \mathbb{P}(\{\omega\}) \\
 &= \sum_{\omega \in \Omega} X(\omega) \mathbb{P}(\{\omega\}) \\
 &= \sum_{x \in \mathcal{X}} x \mathbb{P}(\{\omega \in \Omega \text{ such that } X(\omega) = x\}) \\
 &= \sum_{x \in \mathcal{X}} x p_X(x) \\
 &= \mathbb{E}[X].
 \end{aligned}$$

## 1.9 Measuring Randomness

### 1.9.1 Introduction to Information-Theoretic Measures of Randomness

We just saw some basics for decision making under uncertainty and expected values of random variables. One way we saw for measuring uncertainty was variance. Now we look at a different way of measuring uncertainty or randomness using some ideas from information theory.

In this section, we answer the following questions in terms of bits (as in bits on a computer; everything stored on a computer is actually 0's and 1's each of which is 1 bit):

- How do we measure how random an event is?
- How do we measure how random a random variable or a distribution is?
- How do we measure how different two distributions are?
- How much information do two random variables share?

For now, this material may seem like a bizarre exercise relating to expectation of random variables, but as we will see in the third part of the course on learning probabilistic models, information theory provides perhaps the cleanest derivations for some of the learning algorithms we will derive!

More broadly but beyond the scope of 6.008.1x, information theory is often used to show what the best possible performance we should even hope an inference algorithm can achieve such as fundamental limits to how accurate we can make a prediction. And if you can show that your inference algorithm's performance meets the fundamental

limit, then that certifies that your inference algorithm is optimal! Inference and information theory are heavily intertwined!

## 1.9.2 Shannon Information Content

First, let's consider storing an integer that isn't random. Let's say we have an integer that is from  $0, 1, \dots, 63$ . Then the number of bits needed to store this integer is  $\log_2(64) = 6$  bits: you tell me 6 bits and I can tell you exactly what the integer is.

A different way to think about this result is that we don't *a priori* know which of the 64 outcomes is going to be stored, and so each outcome is equally likely with probability  $\frac{1}{64}$ . Then the number of bits needed to store an event  $\mathcal{A}$  is given by what's called the "Shannon information content" (also called self-information):

$$\log_2 \frac{1}{\mathbb{P}(\mathcal{A})}.$$

In particular, for an integer  $x \in \{0, 1, \dots, 63\}$ , the Shannon information content of observing  $x$  is

$$\log_2 \frac{1}{\mathbb{P}(\text{integer is } x)} = \log_2 \frac{1}{1/64} = \log_2 64 = 6 \text{ bits}.$$

If instead, the integer was deterministically 0 and never equal to any of the other values  $0, 1, \dots, 63$ , then the Shannon information content of observing integer 0 is

$$\log_2 \frac{1}{\mathbb{P}(\text{integer is } 0)} = \log_2 \frac{1}{1} = 0 \text{ bits}.$$

This is not surprising in that a outcome that we deterministically always observe tells us no new information. Meanwhile, for each integer  $x \in \{0, 1, \dots, 63\}$ ,

$$\log_2 \frac{1}{\mathbb{P}(\text{integer is } x)} = \log_2 \frac{1}{0} = \infty \text{ bits}.$$

How could observing one of the integers  $\{1, 2, \dots, 63\}$  tell us infinite bits of information?! Well, this isn't an issue since the event that we observe any of these integers has probability 0 and is thus impossible. An interpretation of Shannon information content is how surprised we would be to observe an event. In this sense, observing an impossible event would be infinitely surprising.

It is possible to have the Shannon information content of an event be some fractional number of bits (e.g., 0.7 bits). The interpretation is that from many repeats of the underlying experiment, the average number of bits needed to store the event is given by the Shannon information content, which can be fractional.

## 1.9.3 Shannon Entropy

To go from the number of bits contained in an event to the number of bits contained in a random variable, we simply take the expectation of the Shannon information content across the possible outcomes. The resulting quantity is called the entropy of a random variable:

$$H(X) = \sum_x p_X(x) \underbrace{\log_2 \frac{1}{p_X(x)}}_{\text{Shannon information content of event } X=x}.$$

The interpretation is that on average, the number of bits needed to encode each i.i.d. sample of a random variable  $X$  is  $H(X)$ . In fact, if we sample  $n$  times i.i.d. from  $p_X$ , then two fundamental results in information theory that are beyond the scope of this course state that: (a) there's an algorithm that is able to store these  $n$  samples in  $nH(X)$  bits, and (b) we can't possibly store the sequence in fewer than  $nH(X)$  bits!

Example: If  $X$  is a fair coin toss "heads" or "tails" each with probability  $1/2$ , then

$$\begin{aligned}
H(X) &= p_X(\text{heads}) \log_2 \frac{1}{p_X(\text{heads})} + p_X(\text{tails}) \log_2 \frac{1}{p_X(\text{tails})} \\
&= \frac{1}{2} \cdot \underbrace{\log_2 \frac{1}{2}}_1 + \frac{1}{2} \cdot \underbrace{\log_2 \frac{1}{2}}_1 \\
&= 1 \text{ bit.}
\end{aligned}$$

Example: If  $X$  is a biased coin toss where heads occurs with probability 1 then

$$\begin{aligned}
H(X) &= p_X(\text{heads}) \log_2 \frac{1}{p_X(\text{heads})} + p_X(\text{tails}) \log_2 \frac{1}{p_X(\text{tails})} \\
&= 1 \cdot \underbrace{\log_2 \frac{1}{1}}_0 + 0 \cdot \underbrace{\log_2 \frac{1}{0}}_1 \\
&= 0 \text{ bits,}
\end{aligned}$$

where  $0 \log_2 \frac{1}{0} = 0 \log_2 1 - 0 \log_2 0 = 0$  using the convention that  $0 \log_2 0 \triangleq 0$ . (Note: You can use l'Hopital's rule from calculus to show that  $\lim_{x \rightarrow 0} x \log x = 0$  and  $\lim_{x \rightarrow 0} x \log \frac{1}{x} = 0$ .)

**Notation:** Note that entropy  $H(X) = \sum_x p_X(x) \log_2 \frac{1}{p_X(x)}$  is in the form of an expectation! So in fact, we can write an expectation:

$$H(X) = \mathbb{E} \left[ \log_2 \frac{1}{p_X(X)} \right].$$

## 1.9.4 Information Divergence

Information divergence (also called “Kullback-Leibler divergence” or “KL divergence” for short, or also “relative entropy”) is a measure of how different two distributions  $p$  and  $q$  (over the same alphabet) are. To come up with information divergence, first, note that entropy of a random variable with distribution  $p$  could be thought of as the expected number of bits needed to encode a sample from  $p$  using the information content according to distribution  $p$ :

$$\underbrace{\sum_x p(x)}_{\text{expectation using } p} \underbrace{\log_2 \frac{1}{p(x)}}_{\text{information content according to } p} \triangleq \mathbb{E}_{X \sim p} \left[ \log_2 \frac{1}{p(X)} \right].$$

Here, we have introduced a new notation:  $\mathbb{E}_{X \sim p}$  means that we are taking the expectation with respect to random variable  $X$  drawn from the distribution  $p$ . If it's clear which random variable we are taking the expectation with respect to, we will often just abbreviate the notation and write  $\mathbb{E}_p$  instead of  $\mathbb{E}_{X \sim p}$ .

If instead we look at the information content according to a different distribution  $q$ , we get

$$\underbrace{\sum_x p(x)}_{\text{expectation using } p} \underbrace{\log_2 \frac{1}{q(x)}}_{\text{information content according to } q} \triangleq \mathbb{E}_{X \sim p} \left[ \log_2 \frac{1}{q(X)} \right].$$

It turns out that if we are actually sampling from  $p$  but encoding samples as if they were from a different distribution  $q$ , then we always need to use more bits! This isn't terribly surprising in light of the fundamental result we alluded to that entropy of a random variable with distribution  $p$  is the minimum number of bits needed to encode samples from  $p$ .



Information divergence is the price you pay in bits for trying to encode a sample from  $p$  using information content according to  $q$  instead of according to  $p$ :

$$D(p \parallel q) = \mathbb{E}_{X \sim p} \left[ \log_2 \frac{1}{q(X)} \right] - \mathbb{E}_{X \sim p} \left[ \log_2 \frac{1}{p(X)} \right].$$

Information divergence is always at least 0, and when it is equal to 0, then this means that  $p$  and  $q$  are the same distribution (i.e.,  $p(x) = q(x)$  for all  $x$ ). This property is called Gibbs' inequality.

Gibbs' inequality makes information divergence seem a bit like a distance. However, information divergence is not like a distance in that it is not symmetric: in general,  $D(p \parallel q) \neq D(q \parallel p)$ .

Often times, the equation for information divergence is written more concisely as

$$D(p \parallel q) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)},$$

which you can get as follows:

$$\begin{aligned} D(p \parallel q) &= \mathbb{E}_{X \sim p} \left[ \log_2 \frac{1}{q(X)} \right] - \mathbb{E}_{X \sim p} \left[ \log_2 \frac{1}{p(X)} \right] \\ &= \sum_x p(x) \log_2 \frac{1}{q(x)} - \sum_x p(x) \log_2 \frac{1}{p(x)} \\ &= \sum_x p(x) \left[ \log_2 \frac{1}{q(x)} - \log_2 \frac{1}{p(x)} \right] \\ &= \sum_x p(x) \log_2 \frac{p(x)}{q(x)}. \end{aligned}$$

Meanwhile, suppose  $q$  is a distribution for a biased coin that always comes up heads (perhaps it's double-headed):

$$q(x) = \begin{cases} 1 & \text{if } x = \text{heads}, \\ 0 & \text{if } x = \text{tails}. \end{cases}$$

Then

$$\begin{aligned} D(p \parallel q) &= p(\text{heads}) \log_2 \frac{p(\text{heads})}{q(\text{heads})} + p(\text{tails}) \log_2 \frac{p(\text{tails})}{q(\text{tails})} \\ &= \frac{1}{2} \log_2 \frac{\frac{1}{2}}{1} + \underbrace{\frac{1}{2} \log_2 \frac{\frac{1}{2}}{0}}_{\infty} \\ &= \infty \text{ bits.} \end{aligned}$$

This is not surprising: If we are sampling from  $p$  (for which we could get tails) but trying to encode the sample using  $q$  (which cannot possibly encode tails), then if we get tails, we are stuck: we can't store it! This incurs a penalty of infinity bits.

Meanwhile,

$$\begin{aligned} D(q \parallel p) &= q(\text{heads}) \log_2 \frac{q(\text{heads})}{p(\text{heads})} + q(\text{tails}) \log_2 \frac{q(\text{tails})}{p(\text{tails})} \\ &= 1 \log_2 \frac{1}{\frac{1}{2}} + \underbrace{0 \log_2 \frac{0}{\frac{1}{2}}}_0 \\ &= 1 \text{ bit.} \end{aligned}$$

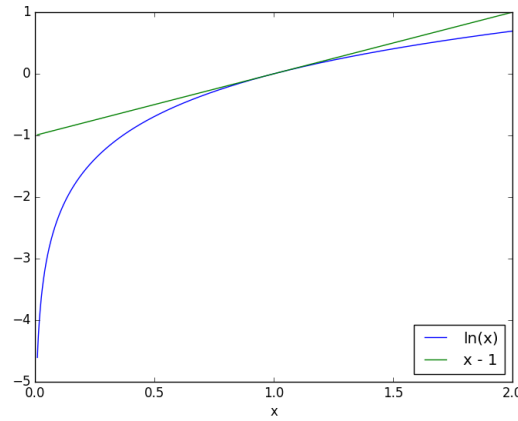
When we sample from  $q$ , we always get heads. In fact, as we saw previously, the entropy of the distribution for an always-heads coin flip is 0 bits since there's no randomness. But here we are sampling from  $q$  and storing the sample using distribution  $p$ . For a fair coin flip, encoding using distribution  $p$  would store each sample using on average 1 bit. Thus, even though a sample from  $q$  is deterministically heads, we store it using 1 bit. This is the penalty we pay for storing a sample from  $q$  using distribution  $p$ .

Notice that in this example,  $D(p \parallel q) \neq D(q \parallel p)$ . They aren't even close — one is infinity and the other is finite!

### 1.9.5 Proof of Gibbs' Inequality

We provide a proof for Gibbs' inequality here for those who are interested. For those of you up for the challenge, try to prove it yourself!

There are various ways to prove Gibbs' inequality. We'll be using a way that relies on the fact that  $\ln x \leq x - 1$  for all  $x > 0$ , with equality if and only if  $x = 1$ , which we provide a proof for at the end of this page, but for which you can also readily see from the following plot:



Code for producing this plot is also at the end of this section.

**Gibbs' inequality:** For any two distributions  $p$  and  $q$  defined over the same alphabet, we have  $D(p \parallel q) \geq 0$ , where equality holds if and only if  $p$  and  $q$  are the same distribution, i.e.,  $p(x) = q(x)$  for all  $x$ .

**Proof:** Recall that changing the base of a log just changes the log by a constant factor:

$$\log_2 x = \frac{\ln x}{\ln 2}.$$

Let  $\mathcal{X}$  be the alphabet of distribution  $p$  restricted to where the probability is positive, i.e.,  $\mathcal{X} = \{a \text{ such that } p(a) > 0\}$ . (There is no need to look at values  $a$  for which  $p(a) = 0$ .)

If  $q(a) = 0$  for any  $a \in \mathcal{X}$ , then  $D(p \parallel q) = \infty$ , so trivially  $D(p \parallel q) > 0$ .

What's left to consider is when  $q(a) > 0$  for every  $a \in \mathcal{X}$ . Then

$$\begin{aligned} D(p \parallel q) &= \sum_{a \in \mathcal{X}} p(a) \log_2 \frac{p(a)}{q(a)} \\ &= \frac{1}{\ln 2} \sum_{a \in \mathcal{X}} p(a) \ln \frac{p(a)}{q(a)} \\ &= -\frac{1}{\ln 2} \sum_{a \in \mathcal{X}} p(a) \ln \frac{q(a)}{p(a)}. \end{aligned}$$

Next, using the fact that  $\ln x \leq x - 1$  for all  $x > 0$ , and accounting for the minus sign outside the summation,

$$\begin{aligned}
D(p \parallel q) &= -\frac{1}{\ln 2} \sum_{a \in \mathcal{X}} p(a) \ln \frac{q(a)}{p(a)} \\
&= -\frac{1}{\ln 2} \sum_{a \in \mathcal{X}} p(a) \left( \frac{q(a)}{p(a)} - 1 \right) \\
&= -\frac{1}{\ln 2} \sum_{a \in \mathcal{X}} (q(a) - p(a)) \\
&= -\frac{1}{\ln 2} \left( \underbrace{\sum_{a \in \mathcal{X}} q(a)}_1 - \underbrace{\sum_{a \in \mathcal{X}} p(a)}_1 \right) \\
&= 0
\end{aligned}$$

Recall that inequality  $\ln x \leq x - 1$  becomes an equality if and only if  $x = 1$ . Thus, the inequality above becomes an equality if and only if, for all  $a \in \mathcal{X}$ , we have  $\ln \frac{q(a)}{p(a)} = \frac{q(a)}{p(a)} - 1$ , which holds if and only if  $\frac{q(a)}{p(a)} = 1$ . Thus  $D(p \parallel q) = 0$  if and only if  $p(a) = q(a)$  for all  $a \in \mathcal{X}$ . This finishes the proof.  $\square$

**Claim:**  $\ln x \leq x - 1$  for all  $x > 0$  where equality holds if and only if  $x = 1$ .

**Proof:** We show that the function  $f$  given by  $f(x) = x - 1 - \ln x$  is always at least 0 and achieves its minimum value at  $x = 1$ . First, note that  $f$  is differentiable for all  $x > 0$  (which implies that  $f$  is continuous on  $(0, \infty)$  and doesn't, for example, do some crazy jump midway through). In fact, the derivative of  $f$  is given by

$$\frac{d}{dx} f(x) = \frac{d}{dx} (x - 1 - \ln x) = 1 - \frac{1}{x}.$$

On the interval  $x > 0$ , the derivative is 0 (and so there's a local extremum) precisely when  $x = 1$ . The question is whether this is a local minimum or a local maximum. We look at the second derivative of  $f$  to do a second derivative test:

$$\frac{d^2}{dx^2} f(x) = \frac{1}{x^2},$$

which is strictly positive for all  $x > 0$ . In other words,  $x = 1$  is a local minimum. The only possible other extrema could happen at the boundaries, but it's easy to check that

$$\lim_{x \rightarrow 0} f(x) = \lim_{x \rightarrow 0} (x - 1 - \ln x) = -1 - \underbrace{\lim_{x \rightarrow 0} \ln x}_{-\infty} = \infty,$$

and

$$\lim_{x \rightarrow \infty} f(x) = \lim_{x \rightarrow \infty} (x - 1 - \ln x) = \infty,$$

since  $x$  grows faster than  $\ln x$ .

Hence,  $f$  attains its global minimum at  $x = 1$ , for which we have

$$f(1) = 1 - 1 - \ln 1 = 0.$$

Since this is the global minimum, we know that  $f(x) \geq 0$  for all  $x > 0$ . Furthermore, since there is only one unique global minimum  $x = 1$ , we further conclude that  $f(x) = 0$  if and only if  $x = 1$ .  $\square$

**Code to produce the plot at the top of this section:**

```
plt.figure()
plt.plot(x, np.log(x))
plt.plot(x, x - 1)
plt.xlabel('x')
plt.legend(['ln(x)', 'x - 1'], loc=4)
plt.show()
```

### 1.9.6 Mutual Information

Mutual information: For two discrete random variables  $X$  and  $Y$ , the mutual information between  $X$  and  $Y$ , denoted as  $I(X;Y)$ , measures how many much information they share. Specifically,

$$I(X;Y) \triangleq D(p_{X,Y} \parallel p_X p_Y),$$

where  $p_X p_Y$  is the distribution we get if  $X$  and  $Y$  were actually independent (i.e., if  $X$  and  $Y$  were actually independent, then we know that the joint probability table would satisfy  $\mathbb{P}(X = x, Y = y) = p_X(x)p_Y(y)$ ).

The mutual information could be thought of as how far  $X$  and  $Y$  are from being independent, since if indeed they were independent, then  $I(X;Y) = 0$ .

On the opposite extreme, consider when  $X = Y$ . Then we would expect  $X$  and  $Y$  to share the most possible amount of information. In this scenario, we can write  $p_{X,Y}(x,y) = p_X(x)\mathbf{1}\{x = y\}$ , and so

$$\begin{aligned} I(X;Y) &= D(p_{X,Y} \parallel p_X p_Y) \\ &= \sum_x \sum_y p_{X,Y}(x,y) \log_2 \frac{1}{p_X(x)p_Y(y)} \\ &\quad - \sum_x \sum_y p_{X,Y}(x,y) \log_2 \frac{1}{p_{X,Y}(x,y)} \\ &= \sum_x \sum_y p_X(x)\mathbf{1}\{x = y\} \log_2 \frac{1}{p_X(x)p_Y(y)} \\ &\quad - \sum_x \sum_y p_X(x)\mathbf{1}\{x = y\} \log_2 \frac{1}{p_X(x)\mathbf{1}\{x = y\}} \\ &= \sum_x p_X(x) \log_2 \left( \frac{1}{p_X(x)} \right)^2 - \sum_x p_X(x) \log_2 \frac{1}{p_X(x)} \\ &= 2 \sum_x p_X(x) \log_2 \frac{1}{p_X(x)} - \sum_x p_X(x) \log_2 \frac{1}{p_X(x)} \\ &= \sum_x p_X(x) \log_2 \frac{1}{p_X(x)} \\ &= H(X). \end{aligned}$$

This is not surprising: if  $X$  and  $Y$  are the same, then the number of bits they share is exactly the average number of bits needed to store  $X$  (or  $Y$ ), namely  $H(X)$  bits.

### 1.9.7 Exercise: Mutual Information

Consider the following joint probability table for random variables  $X$  and  $Y$ . We'll compute the mutual information  $I(X;Y)$  of random variables  $X$  and  $Y$  step-by-step.

		Y		
		0	1	2
X	0	0.10	0.09	0.11
	1	0.08	0.07	0.07
	2	0.18	0.13	0.17

Mutual information is about comparing the joint distribution of  $X$  and  $Y$  with what the joint distribution would be if  $X$  and  $Y$  were actually independent.

In Python (where we won't explicitly store the labels of the rows and columns):

```
import numpy as np
joint_prob_XY = np.array([[0.10, 0.09, 0.11], [0.08, 0.07, 0.07], [0.18, 0.13, 0.17]])
```

The marginal distributions  $p_X$  and  $p_Y$  are given by:

```
prob_X = joint_prob_XY.sum(axis=1)
prob_Y = joint_prob_XY.sum(axis=0)
```

Next, we produce what the joint probability table would be if  $X$  and  $Y$  were actually independent:

```
joint_prob_XY_indep = np.outer(prob_X, prob_Y)
```

At this point, we have the joint distribution of  $X$  and  $Y$  (denoted  $p_{X,Y}$ ) stored in code as `joint_prob_XY`, and also what the joint distribution would be if  $X$  and  $Y$  were independent (denoted  $p_X p_Y$ ) stored in code as `joint_prob_XY_indep`. The mutual information of  $X$  and  $Y$  is precisely given by the KL divergence between  $p_{X,Y}$  and  $p_X p_Y$ :

$$I(X;Y) = D(p_{X,Y} \parallel p_X p_Y) = \sum_x \sum_y p_{X,Y}(x,y) \log_2 \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}.$$

## 1.9.8 Information-Theoretic Measures of Randomness: Where We'll See Them Next

In the third part of 6.008.1x when we talk about learning probabilistic models, at a basic level, what we have are observations (i.e., data we collect from the world), and our goal is to decide which probabilistic model in some sense “best” fits the observations. How we can decide on which probabilistic model to use is to give each candidate probabilistic model a score and then we pick the one with the highest score.

The score we will use is what's called “maximum likelihood”, which is quite popular and has been extensively studied. By choosing to use maximum likelihood to decide on which candidate probabilistic model to use, very naturally entropy and information divergence will emerge! Importantly, information divergence will say how far a candidate model is from the observed data. Meanwhile, mutual information will come up to help us figure out which random variables we should directly model pairwise interactions with.

## 1.10 Towards Infinity in Modeling Uncertainty

### 1.10.1 Infinite Outcomes

What if we want an infinite number of outcomes? For example, consider an underlying experiment where we keep flipping a coin until we see the first heads. We might have to flip an arbitrarily large number of tosses! Here, the sample space would consist of getting heads for the first time after 1 toss, after 2 tosses, and so forth, ad infinitum.

On a computer, we can't actually store an arbitrary probability table with an infinite number of entries.

A few workarounds:

- Approximate the probability distribution with a finite probability space.
- In the above example, we could for example truncate and lump together all the possible outcomes in which heads appears after the 1000-th toss into a single possible outcome; we lose the ability to reason about the first heads appearing on, say, the 2000-th toss, but we could argue that heads appearing after the 1000th toss is extremely rare anyways.

- For very specific probability distributions, there can be a way for us to represent the distribution “in closed form” meaning that if we just keep track of a few numbers, these few numbers are enough to tell us what the probability is for any possible outcome in an infinitely large sample space.

Recall that the binomial distribution is an example of this: with just two numbers, we can query any entry in a probability table with far more than just two entries!

### 1.10.2 The Geometric Distribution

We now give an example of a distribution with an infinite alphabet size called the geometric distribution that has only 1 parameter. Thus, storing 1 number tells you what all the probability table entries are, even though there are an infinite number of entries!

TODO - add notes from video

### 1.10.3 Practice Problem: The Geometric Distribution

Let  $X \sim \text{Geo}(p)$  so that

$$p_X(x) = (1-p)^{x-1}p \quad \text{for } x = 1, 2, \dots$$

- Show that each of the table entries  $p_X(x)$  is nonnegative for  $x = 1, 2, \dots$

**Solution:** Note that  $(1-p) \geq 0$ ,  $x-1 \geq 0$ , and  $p \geq 0$ , and so  $(1-p)^{x-1}p \geq 0$  for all  $p \in (0, 1)$  and  $x = 1, 2, 3, \dots$

- Show that the sum of all the table entries is 1, i.e.,  $\sum_{x=1}^{\infty} p_X(x) = 1$ .

You may find the following result from calculus helpful: For  $r \in (-1, 1)$ ,

$$\sum_{i=0}^{\infty} r^i = \frac{1}{1-r}.$$

**Solution:** For  $p \in (0, 1)$ ,

$$\sum_{x=1}^{\infty} p_X(x) = \sum_{x=1}^{\infty} (1-p)^{x-1}p \stackrel{(a)}{=} p \sum_{i=0}^{\infty} (1-p)^i \stackrel{(b)}{=} p \cdot \frac{1}{1-(1-p)} = p \cdot \frac{1}{p} = 1,$$

where step (a) substitutes  $i = x - 1$ , and step (b) uses the result above from calculus.

### 1.10.4 Discrete Probability Spaces and Random Variables

In the case of the geometric distribution, the sample space  $\Omega$  is what’s called “countably infinite”. This means that it has an infinite (rather than finite) number of entries, and that there’s actually a way for us to arrange the elements so that there’s a 1st element, 2nd, 3rd, and so forth off into infinity. Note that the set of real numbers is not countable.

Before this section, every time we used the phrases “probability space” and “random variable”, we actually meant “finite probability space” and “finite random variable”.

More general than the finite probability space and finite random variable are the discrete probability space and discrete random variable:

**Definition of a “discrete probability space”:** A discrete probability space  $(\Omega, \mathbb{P})$  is the same thing as a finite probability space except that the sample space  $\Omega$  is allowed to be either finite or countably infinite. In particular, a discrete probability space consists of two ingredients:

a finite or countably infinite sample space  $\Omega$  that is the collectively exhaustive, mutually exclusive set of all possible outcomes

an assignment of probability  $\mathbb{P}$ , where for any outcome  $\omega \in \Omega$ , we have  $\mathbb{P}(\text{outcome } \omega)$  be a number at least 0 and at most 1, and

$$\sum_{\omega \in \Omega} \mathbb{P}(\text{outcome } \omega) = 1.$$

**Definition of a “discrete random variable”:** A discrete random variable  $X$  is the same thing as a finite random variable except that it’s associated with a discrete probability space rather than a finite probability space. In particular, given a discrete probability space  $(\Omega, \mathbb{P})$ , a discrete random variable  $X$  maps  $\Omega$  to a set of values  $\mathcal{X}$  that the random variable can take on. Again, we can think of such a random variable  $X$  as generated from a two step procedure: some possible outcome  $\omega$  is sampled from the discrete probability space  $(\Omega, \mathbb{P})$ , and then  $X$  takes on the value given by  $X(\omega)$ .

Formally defining random variables that are even more general than discrete random variables requires more sophisticated mathematical machinery that is beyond the scope of 6.008.1x.

## **Chapter 2**

# **Graphical Models**



## **Chapter 3**

# **Learning a Probabilistic Model from Data**

# Appendix A

## Notation Summary

Typically we use a capital letter like  $X$  to denote a random variable, a script (or calligraphic) letter  $\mathcal{X}$  to denote a set (or an event), and a lowercase letter like  $x$  to refer to a nonrandom variable. Occasionally we will also use capital letters to refer to a constant that is not varying throughout the problem (in contrast to using a lowercase letter like  $x$  that can be a “dummy” variable such as in a summation  $\sum_x p_X(x)$ , for which lowercase  $x$  refers to a specific constant value but we are varying what  $x$  is and it is effectively a temporary variable that we do not need after computing the summation).

$p_X$  or  $p_X(\cdot)$                       probability table/probability mass function (PMF)/probability distribution/marginal distribution of random variable  $X$

$p_X(x)$  or  $\mathbb{P}(X = x)$                       probability that random variable  $X$  takes on value  $x$

$p_{X,Y}$  or  $p_{X,Y}(\cdot, \cdot)$                       joint probability table/joint PMF/joint probability distribution of random variables  $X$  and  $Y$

$p_{X,Y}(x, y)$  or  $\mathbb{P}(X = x, Y = y)$                       probability that  $X$  takes on value  $x$  and  $Y$  takes on value  $y$

$p_{X|Y}(\cdot | y)$                       conditional probability table/conditional PMF/conditional probability distribution of  $X$  given  $Y$  takes on value  $y$

$p_{X|Y}(x | y)$  or  $\mathbb{P}(X = x | Y = y)$                       probability that  $X$  takes on value  $x$  given that  $Y$  takes on value  $y$

$X \sim p$  or  $X \sim p(\cdot)$                        $X$  is distributed according to distribution  $p$

$X \perp Y$                        $X$  and  $Y$  are independent

$X \perp Y | Z$                        $X$  and  $Y$  are independent given  $Z$

We will also of course be dealing with many events or many random variables. For example,  $\mathbb{P}(\mathcal{A}, \mathcal{B}, \mathcal{C} | \mathcal{D}, \mathcal{E})$  would be the probability that events  $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\mathcal{C}$  all occur, given that both events  $\mathcal{D}$  and  $\mathcal{E}$  occur, which by the definition of conditional probability would be

$$\mathbb{P}(\mathcal{A}, \mathcal{B}, \mathcal{C} | \mathcal{D}, \mathcal{E}) = \frac{\mathbb{P}(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E})}{\mathbb{P}(\mathcal{D}, \mathcal{E})}.$$

Similarly,  $p_{X,Y,Z|V,W}$  would refer to a joint conditional distribution of random variables  $X$ ,  $Y$ , and  $Z$  given both  $V$  and  $W$  taking on specific values together:

$$p_{X,Y,Z|V,W}(x, y, z | v, w) = \frac{p_{X,Y,Z,V,W}(x, y, z, v, w)}{p_{V,W}(v, w)}.$$

When we have a collection of random variables, e.g.,  $W, X, Y, Z$ , if we say that they are independent (without specifying what type of independence), then what we mean is mutual independence, which means that the joint distribution factorizes into the marginal distributions:

$$p_{W,X,Y,Z}(w,x,y,z) = p_W(w)p_X(x)p_Y(y)p_Z(z) \quad \text{for all } w,x,y,z.$$