Probability

Table of Contents

[Uncertainty 3](#_Toc135925029)

[Random Variables 3](#_Toc135925030)

[Probability Distributions 4](#_Toc135925031)

[Joint Distribution 5](#_Toc135925032)

[Probabilistic Models 6](#_Toc135925033)

[Events 6](#_Toc135925034)

[Marginal Distributions 7](#_Toc135925035)

[Conditional Probabilities 7](#_Toc135925036)

[Conditional Distribution 8](#_Toc135925037)

[Normalization Trick 9](#_Toc135925038)

[Probabilistic Inference 9](#_Toc135925039)

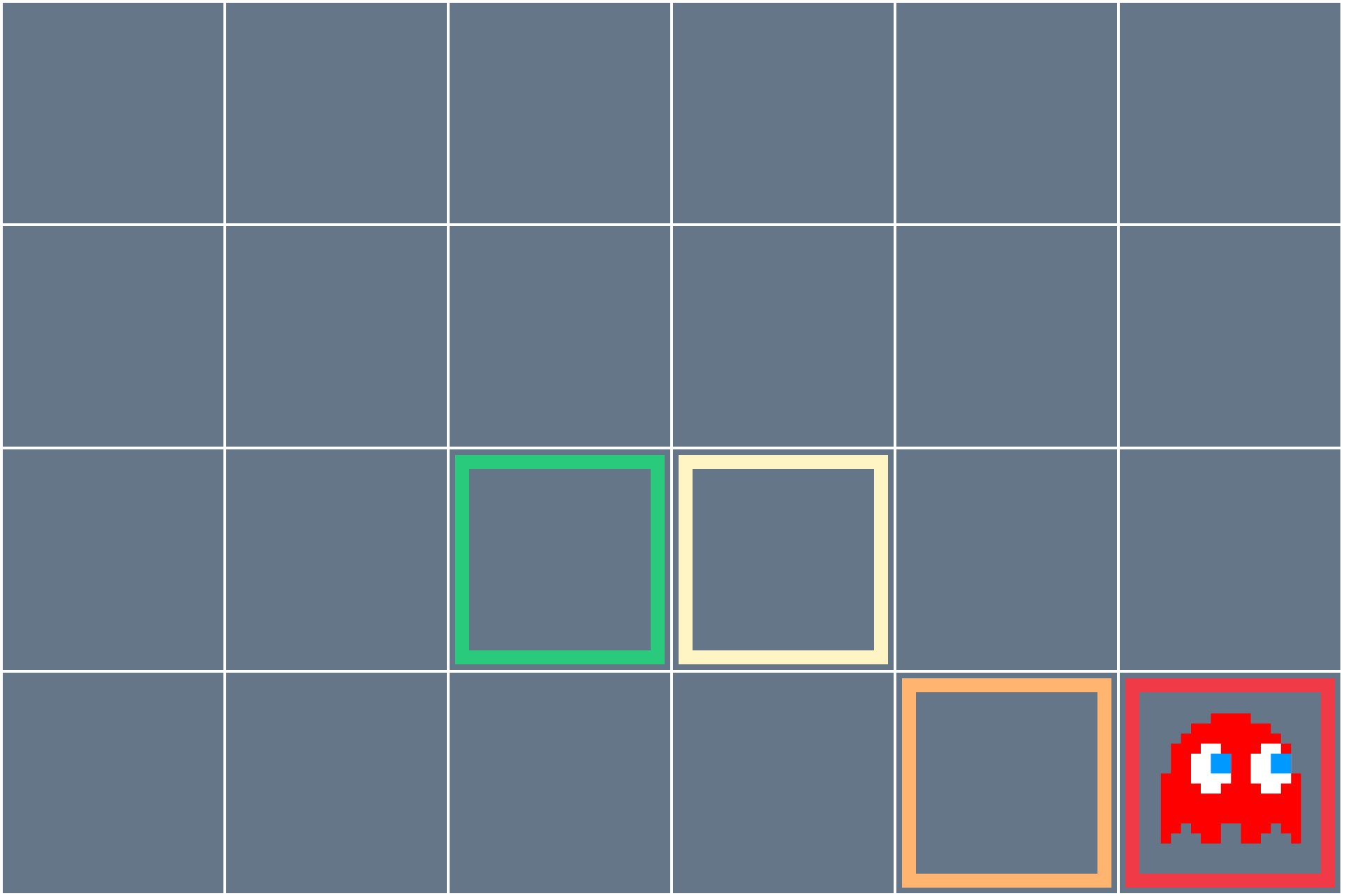
[Inference by Enumeration 10](#_Toc135925040)

[Product Rule 11](#_Toc135925041)

[Chain Rule 11](#_Toc135925042)

[Bayes Rule 12](#_Toc135925043)

[Ghostbusters 13](#_Toc135925044)



Suppose we have a grid that has a ghost hidden in one of the cells. Our agent needs to catch the ghost. To do this, it can use one of two actions. The first action is the ‘bust’ action. The agent can only use this action once, on the cell which it thinks the ghost is hiding in. If the agent gets it right, it gets a large reward. If it gets it wrong, it gets a large penalty. The second action is to scan a cell using a sensor. Each cell scan has a small cost to deter scanning every cell. The scan can return one of the following results:

* Red – The ghost is in this cell.
* Orange – The ghost is 1 or 2 cells away.
* Yellow – The ghost is 3 or 4 cells away.
* Green – The ghost is 5+ cells away.

The problem is, the sensor readings are noisy, meaning the results are not necessarily 100% accurate. We do, however, know the probability distribution, given as .

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

## Uncertainty

To deal with the noisy sensor readings, we need to deal with **uncertainty**. We will have some **observed variables**, whose values we know (such as the sensor readings), some **unobserved variables**, whose values we don’t know (such as where the ghost is) and finally, a **model**, which describes how the observed variables relate to the unobserved ones.

Since we’re dealing with uncertainty, we should go over the basics of probabilistic reasoning.

### Random Variables

A **random variable** is some aspect of the world about which we might have some uncertainty. For example, we could have a random variable which denotes whether or not it will rain today. These random variables are denoted using **capital letters**, such as .

The domains of the random variables depend on what they represent. Some cases are:

* , which can also be written as

### Probability Distributions

A **probability distribution** is just a table which associates a probability with each value of a random variable. For example, we could have a probability distribution for a random variable for the weather.

|  |  |
| --- | --- |
|  | |
|  |  |
| Sun | 0.6 |
| Rain | 0.1 |
| Fog | 0.3 |
| Meteor | 0.0 |

Generally, we try to avoid probabilities since they cause issues with calculations. Instead, we might use some miniscule value, such as .

Only **unobserved random variables** can have a probability distribution. Once a random variable is observed, it has a specific value with no uncertainty involved.

From the probability distribution, we can derive **probability values**, such as . This can also be written in shorthand as , given that the shorthand does not cause any ambiguity.

There are two key rules that probability distributions must follow:

### Joint Distribution

A **joint distribution** is like a probability distribution, except that it lists the probabilities of different combinations of values of multiple random variables occurring together.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
| Hot | Sun | 0.4 |
| Hot | Rain | 0.1 |
| Cold | Sun | 0.2 |
| Cold | Rain | 0.3 |

The two rules of probability distributions also apply to join distributions.

A joint distribution table is essential for solving problems related to uncertainty, and we will be using them in heavily. However, they have a huge issue. For random variables, each with a domain size of , the total table size is . Because of this, joint distribution tables quickly become too big to even have in memory. We will later look into ways in which we can avoid having to create a join distribution table at all.

### Probabilistic Models

The joint distribution over all the random variables involved in some scenario is called the **probabilistic model** of that scenario. It consists of the random variables and their domains, the different combinations of values, called **outcomes** and the probability values of each combination. The probability values are **normalized** so that they sum to 1.0. Ideally, only a few of the variables will directly interact.

### Events

An **event** is a specific outcome from a probabilistic model. Using a joint distribution, we can calculate the probability of any outcome. For example, from the joint distribution table below, we can tell that the probability of it being hot is .

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
| Hot | Sun | 0.4 |
| Hot | Rain | 0.1 |
| Cold | Sun | 0.2 |
| Cold | Rain | 0.3 |

### Marginal Distributions

A **marginal distribution** removes columns from a join distribution by summing taking the sum of the different combinations of values from those columns. For example, we can remove the weather column from the joint distribution above.

|  |  |
| --- | --- |
|  | |
|  |  |
| Hot | 0.5 |
| Cold | 0.5 |

Formally,

It is also possible to go in the reverse direction, deriving the joint distribution given the marginal distribution.

### Conditional Probabilities

The **conditional probability** of an event is the probability of the event occurring given that we already know that some variable in that event has a specific value.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
| Hot | Sun | 0.4 |
| Hot | Rain | 0.1 |
| Cold | Sun | 0.2 |
| Cold | Rain | 0.3 |

For the table above, if we know that it is cold, the probability of it being sunny is given by:

### Conditional Distribution

A **conditional distribution** is a probability distribution where we know the values of one or more variables.

|  |  |
| --- | --- |
|  | |
|  |  |
| Sun | 0.4 |
| Rain | 0.6 |

## Normalization Trick

The **normalization trick** is just a different way of looking at the conditional probabilities. Instead of looking at the formula, we can consider it as taking one portion of the total probability under some condition.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
| Hot | Sun | 0.4 |
| Hot | Rain | 0.1 |
| Cold | Sun | 0.2 |
| Cold | Rain | 0.3 |

Thus, .

## Probabilistic Inference

**Probabilistic Inference** is the process by which the probability of an unknown variable changes as other variables become known. For example, initially, the probability of sunny weather was . However, once we got to know that the temperature would be cold, this dropped to . If we had more variables, the probability would keep updating as we got to know more values.

## Inference by Enumeration

We want to find the value of some **query variable**, , given the values of some **evidence variables**, , i.e., we want to find .

As we gather more evidence, we can start removing rows. By normalizing the values, we can maintain the sum being .

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Summer | Hot | Sun | 0.30 |
| Summer | Hot | Rain | 0.05 |
| Summer | Cold | Sun | 0.10 |
| Summer | Cold | Rain | 0.05 |
| Winter | Hot | Sun | 0.10 |
| Winter | Hot | Rain | 0.05 |
| Winter | Cold | Sun | 0.15 |
| Winter | Cold | Rain | 0.20 |

The problem with using this mechanism is that it has a time complexity and a space complexity of . Additionally, we do not always know the probability values.

## Product Rule

If we know the conditional distribution but need to joint distribution, we can obtain it using the **product rule**.

## Chain Rule

The **Chain Rule** is a generalization of the product rule that allows us to work with multiple conditions.

## Bayes Rule

If we know one conditional probability, it is possible to obtain the other.

This is useful because often, it is difficult to calculate one conditional probability but easy to calculate the other.

|  |  |
| --- | --- |
|  | |
|  |  |
| Sun | 0.8 |
| Rain | 0.2 |

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |
| Wet | Sun | 0.1 |
| Dry | Rain | 0.9 |
| Wet | Sun | 0.7 |
| Dry | Rain | 0.3 |

From the table above, we can calculate as:

## Ghostbusters

Coming back to the original problem, we can calculate the probability of a successful hit at a particular location based on the results of location readings in the surrounding cells. Suppose we have a 3x3 grid and initially, ever cell has the ghost in it with probability . When we get a reading, we know that it is a specific colour with some probability. Suppose if the ghost is in the cell, the colour is yellow with probability , i.e., . From this, we can use Bayes rule to calculate the probability of the ghost actually being there. With enough readings, we can find the ghost with 100% probability.