Bayes’ Rule and the Two Armed Bandit

Complete all **Exercises**, and submit answers to **Questions** on the Coursera platform.

Background

Some people refer to slot machines as “One-armed Bandits” due to the older style of machine requiring the player to pull a mechanical handle to play. Statisticians and mathematicians often develop theory / models based on games of chance which turn out to be more generally useful. One general class of probability / optimization problems is known as the multi-armed bandit problems which is based on the following analogy: A player walks into a casino and sees a wall of slot machines. All of the machines pay out at different rates - some pay out more often than others, some pay out less often. Since the player does not know which machines are “good” and which are “bad”, how should he / she play so as to make as much money (or at least lose as little) as possible?

Simulated Slots

Today we will examine a simplified case where there are only two machines (one “Good” and one “Bad”). We will also assume that we know the probability of winning on the “Good” machine and the probability of winning on the “Bad” machine - what we don’t know is which machine is which.

The Shiny App below will let you simulate playing slot machines when the probability of winning on the “good” machine is 1/2 and the probability of winning on the “bad” machine is 1/3. Each time you play the App will “flip a coin” and randomly select either Machine 1 or Machine 2 to be the “good” machine, and the other to be the “bad” machine.

Machine 1:

Play!

Machine 2:

Play!

Events:

Log Data

play	machine	outcome

Decision:

Machine 1 is good

Machine 2 is good

Reset

Results:

Wins	Loses	Plays
0	0	0
0	0	0

Exercise: Use the Shiny app above to play the slot machines a total of *10* times. You can divide your plays up however you like between the two machines. Once you have played 10 times, use the results of your plays to decide which machine you think has the better payout (i.e. the good machine) - click the button on the right that corresponds to your guess, the App will tell you if you are right. If you were right or wrong, press the reset button and play again and guess which machine you think is the good machine. As you are playing think about what it is about your results that enabled you make the correct guess.

Exercise: Press the Reset button again, now play *30* times and use those results to guess which machine is the good one. Do you think it was easier or harder to make a decision with the additional plays? Why do you think that is?

Hopefully what you have observed is that as you played the slot machine initially it was difficult to determine which machine was which but as you played more it became more and more clear. In particular each time you play you naturally reassesses which machine you think is good, with the initial handful of plays your beliefs stay close to 50-50, potentially with a small bias towards the machine you've won more on. By the time you get to 30 plays you should have very strong beliefs about which machine is the "good" one.

This is the way in which we usually interact with the world - we try something and based on the outcome we modify our mental model of the world. This idea of updating beliefs based on observed data is one of the core tenets of Bayesian statistics - in the following sections we will work through the probability calculations and see how they correspond with our intuitive understanding.

Posterior Probabilities

We will start by examining the result of playing just once. Imagine that you play Machine 1 and you win, what do we now know about the probability of the two machines being "good" or "bad"? Since each machine is equally likely to be the "good" machine we can express this as

$$P(M_1 \text{ is Good}) = P(M_2 \text{ is Bad}) = 1/2$$

$$P(M_1 \text{ is Bad}) = P(M_2 \text{ is Good}) = 1/2.$$

We have also been told the probability of winning for each type of machine

$$P(\text{Win on } M_1 \mid M_1 \text{ is Good}) = 1/2 \qquad P(\text{Win on } M_1 \mid M_1 \text{ is Bad}) = 1/3.$$

We can use these probabilities to calculate the probability of losing for each type of machine as well

$$P(\text{Lose on } M_1 \mid M_1 \text{ is Good}) = 1/2 \qquad P(\text{Lose on } M_1 \mid M_1 \text{ is Bad}) = 2/3.$$

Note that while these probabilities are all for Machine 1, but they are exactly the same as the probabilities for Machine 2. We have seen how we can use Bayes' rule to calculate $P(M_1 \text{ is Good} \mid \text{Win on } M_1)$

$$\begin{aligned}
P(M_1 \text{ is Good} \mid \text{Win on } M_1) &= \frac{P(\text{Win on } M_1 \mid M_1 \text{ is Good})\, P(M_1 \text{ is Good})}{P(\text{Win on } M_1)} \\
&= \frac{P(\text{Win on } M_1 \mid M_1 \text{ is Good})\, P(M_1 \text{ is Good})}{P(\text{Win on } M_1 \mid M_1 \text{ is Good})\, P(M_1 \text{ is Good}) + P(\text{Win on } M_1 \mid M_1 \text{ is Bad})\, P(M_1 \text{ is Bad})} \\
&= \frac{1/2 \times 1/2}{1/2 \times 1/2 + 1/3 \times 1/2} = 0.6
\end{aligned}$$

Based on the preceding result, what is the probability that Machine 1 is "Bad" given you won playing on Machine 1?

- 0.3
- 0.4
- 0.5
- 0.6
- 0.7

Based on the preceding result, what is the probability that Machine 2 is "Good" given you won playing on Machine 1?

- 0.3
- 0.4
- 0.5
- 0.6
- 0.7

Under the Bayesian paradigm, which of the following correctly matches the probabilities with their names?

- Posterior - $P(M_1 \text{ is Good} \mid \text{Win on } M_1)$
  Prior - $P(M_1 \text{ is Good})$
  Likelihood - $P(\text{Win on } M_1 \mid M_1 \text{ is Good})$
- Posterior: $P(M_1 \text{ is Good} \mid \text{Win on } M_1)$
  Prior: $P(\text{Win on } M_1 \mid M_1 \text{ is Good})$
  Likelihood: $P(M_1 \text{ is Good})$
- Posterior: $P(\text{Win on } M_1 \mid M_1 \text{ is Good})$
  Prior: $P(M_1 \text{ is Good} \mid \text{Win on } M_1)$
  Likelihood: $P(M_1 \text{ is Good})$
- Posterior: $P(\text{Win on } M_1 \mid M_1 \text{ is Good})$
  Prior: $P(M_1 \text{ is Good})$
  Likelihood: $P(M_1 \text{ is Good} \mid \text{Win on } M_1)$

Bayesian Updating

We have implemented a function for calculating the posterior probability of Machine 1 and Machine 2 being the "good" machine after one or more plays of either machine. The function `bandit_posterior` expects a data frame representing your play history that contains two columns, `machine` which records which machine was played (e.g. either a 1 or 2) and `outcome` which records whether you won (`"W"`) or lost (`"L"`). An optional parameter to `bandit_posterior` is `prior` , a vector of length two that specifies the prior probability of each machine being "good". If left unspecified, equal prior probabilities (0.5, 0.5) are assumed. We can repeat the calculation from the previous section using the following code:

```
bandit_posterior(data = data.frame(machine=1L, outcome="W"))
```

```
## m1_good m2_good
##     0.6     0.4
```

We can also use this function to calculate the posterior probabilities for additional plays, for example playing Machine 1 twice, first winning and then losing.

```
bandit_posterior(data = data.frame(machine=c(1L,1L), outcome=c("W","L")))
```

```
##    m1_good   m2_good
## 0.5294118 0.4705882
```

We have discussed how the Bayesian approach allows for updating procedures where for each new data observation we are able to use the previous posterior probabilities as our new prior probabilities and thereby simplify the calculation (e.g. multiple simple updates can be used instead of one single large calculation). We can explore this process by chaining multiple calls to `bandit_posterior` together by using the returned posterior values as the prior in the next call.

```
data1 = data.frame(machine=c(1L), outcome=c("W"))
data2 = data.frame(machine=c(1L), outcome=c("L"))
bandit_posterior(data1) %>% bandit_posterior(data2, prior=.)
```

```
##    m1_good   m2_good
## 0.5294118 0.4705882
```

Note that this exactly matches the probabilities we calculated when we provided the data all at once.

Using the `bandit_posterior` function calculate the posterior probabilities of Machine 1 and 2 being "good" after playing Machine 1 twice and winning both times and then playing Machine 2 three times and winning twice and then losing.

$P(M_1 \text{ is good} \mid \text{data}) = 0.250, P(M_2 \text{ is good} \mid \text{data}) = 0.750$

$P(M_1 \text{ is good} \mid \text{data}) = 0.429, P(M_2 \text{ is good} \mid \text{data}) = 0.571$

$P(M_1 \text{ is good} \mid \text{data}) = 0.571, P(M_2 \text{ is good} \mid \text{data}) = 0.429$

$P(M_1 \text{ is good} \mid \text{data}) = 0.750, P(M_2 \text{ is good} \mid \text{data}) = 0.250$

What would the posterior probabilities be if we had instead played Machine 2 first, playing three times, winning twice and losing once and then playing Machine 1 twice and winning both times?

$P(M_1 \text{ is good} \mid \text{data}) = 0.250, P(M_2 \text{ is good} \mid \text{data}) = 0.750$

$P(M_1 \text{ is good} \mid \text{data}) = 0.429, P(M_2 \text{ is good} \mid \text{data}) = 0.571$

$P(M_1 \text{ is good} \mid \text{data}) = 0.571, P(M_2 \text{ is good} \mid \text{data}) = 0.429$

$P(M_1 \text{ is good} \mid \text{data}) = 0.750, P(M_2 \text{ is good} \mid \text{data}) = 0.250$

Exercise: Confirm the updating property we just discussed by connecting together two calls of `bandit_posterior`, the first of which calculates the posterior probability for the first two plays on Machine 1. The second call should use these values as its prior and then calculate a new posterior using the data from the subsequent three plays on Machine 2.

Back to the Bandits

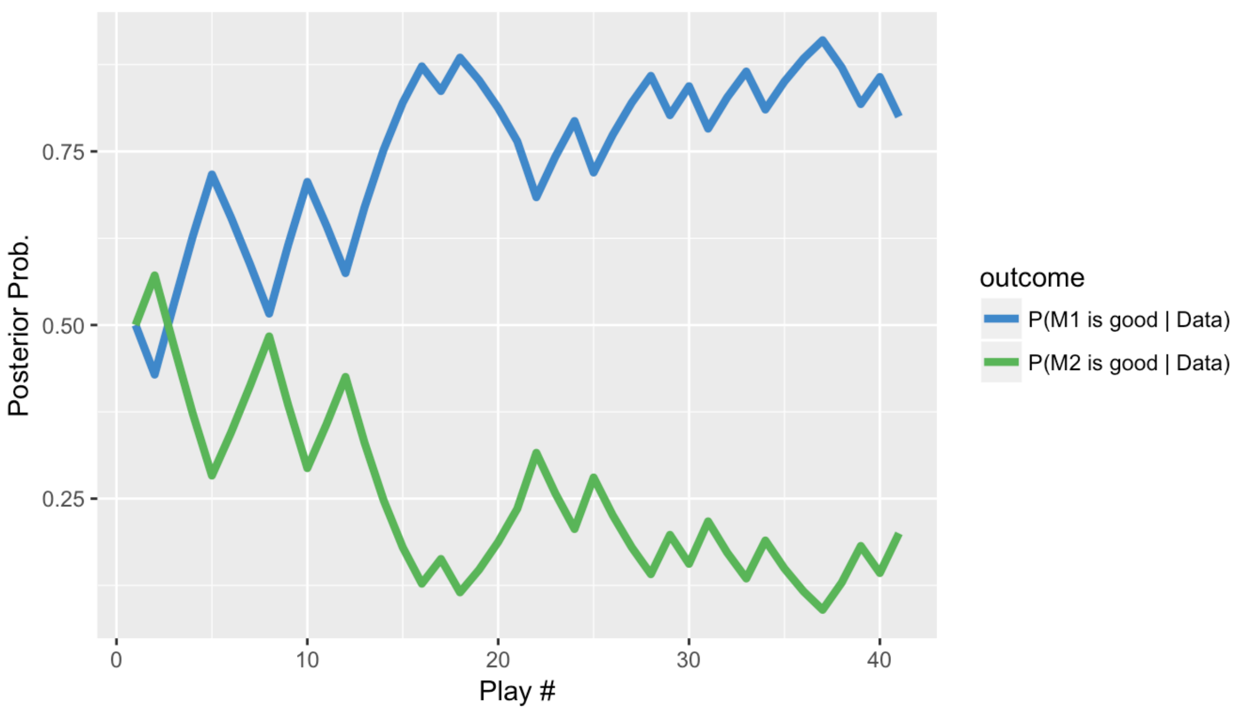
You may have notice that if you click on the Data tab in the middle of the App above you are given code for a data frame that represents the results of your plays within the machine.

Use this data frame with the `bandit_posterior` function to calculate the exact posterior probability of each machine being "good". Do these probabilities match with your intuition about which machine was good?

Reset the simulation and then play at least 50 times. Use subsetting (e.g. `data[1:10,]` for the first 10 plays) to calculate the posterior probability at every 10th play and observe how it changes as more plays are made.

We can visualize how these posterior probabilities update using the `plot_posterior` function which calculates and plots the posterior probability after each play.

```
data = data.frame(machine = c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
                              1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
                              2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
                              2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L),
                  outcome = c("L", "W", "W", "W", "L", "L", "L", "W", "W", "L",
                              "L", "W", "W", "W", "W", "L", "W", "L", "L", "L",
                              "W", "L", "L", "W", "L", "L", "L", "W", "L", "W",
                              "L", "L", "W", "L", "L", "L", "W", "W", "L", "W"))
plot_bandit_posterior(data)
```



Plot the result of your last 50 plays, describe the pattern you see for the two posterior probabilities.

Why do the posterior probabilities for Machine 1 and Machine 2 mirror each other?

$P(M_1 \mid \text{data})$ and $P(M_2 \mid \text{data})$ are complementary

Machine 1 and Machine 2 being “good” are mutually exclusive events

All of the above

This is a derivative of an OpenIntro (https://www.openintro.org/stat/labs.php) lab, and is released under a Attribution-NonCommercial-ShareAlike 3.0 United States (https://creativecommons.org/licenses/by-nc-sa/3.0/us/) license.