CS4375 Practical Assignment: Bayes

September 12, 2022

Contents

1 Intro

In this practical assignment, we will get more acquainted with Bayes theorem through programming.

Deliverable
For this assignment you are required to upload a zip-file that contains three components:

- a .txt file with answers to the questions, with 1 answer per line.
- your code to generate the .txt file containing solutions to the coding exercises.
- a readme file that includes a single script to generate the .txt file with your code.

You can round your answers up to 3 decimals, where necessary, e.g.:

```
0.348
0.846
0.234
Expert
```

The .txt should have a name like "group_xx.txt", where "xx" should be replaced by your groupnumber.

Some Theory
Bayes's theorem states:

$$P(H|O) = \frac{P(H)P(O|H)}{P(O)}$$

We will refer to H as the *hypothesis*, and to O as the *observation*. So we have

1. $P(H)$, the probability of the hypothesis before any observations, which we call the prior probability, or *prior*.
2. $P(H|O)$, the probability of the hypothesis after an observation, which we call the posterior probability, or *posterior*.
3. $P(O|H)$, the probability of the observation under the assumption that our hypothesis is true, which we call the *likelihood*.
4. $P(O)$ is the probability of the observation under any hypothesis, which we call the *normalizing constant*. This constant is necessary to ensure the probabilities sum up to 1.

In this assignment we will create a class **Bayes** that is able to calculate posterior probabilities when given the priors, likelihood function, and possible observations. In the first problem you will write the code for this class and use it to answer some questions. In the second problem you should only have to initialize a new instance of the **Bayes** class in order to answer the questions posed there.

2 The Cookie Problem [1]

In this problem we have two bowls of cookies. In Bowl 1 there are 35 vanilla cookies and 15 chocolate cookies. In Bowl 2 there are 20 vanilla cookies and 30 chocolate cookies.
At random we pick one of the bowls and, without looking, we take a cookie from the bowl. We observe that the cookie is vanilla. What is the probability that it came from Bowl 1? To answer this question we will write some code. In Section 2.1 an example main function is provided to show calls to the functions.

[1] Adapted from the book "Think Bayes: Bayesian Statistics Made Simple" by Allen B. Downey.

We have two hypothesis, either it comes from Bowl 1 or it comes from Bowl 2. Since we picked a bowl at random we have the following priors: $P(\text{Bowl 1}) = P(H_1) = 0.5$, $P(\text{Bowl 2}) = P(H_2) = 0.5$.
Coding Exercise 1. Write a class **Bayes** that takes as initialization arguments the following: a list of hypothesis, a list with the priors of the hypothesis, a list of possible observations and a likelihood array.

Now that we have the priors we can start with working towards the posterior, given that we observed that we got a vanilla cookie. To get to the posterior we first need to get the likelihood, $P(O|H)$. In this problem this is fairly straightforward, e.g. we have $P(\text{vanilla}|\text{Bowl 1}) = 0.70$.
Coding Exercise 2. In **Bayes** write a function **likelihood** that takes as arguments an observation (e.g. "vanilla") and an hypothesis (e.g. "Bowl 1") and returns the likelihood.

With the likelihood we can almost calculate the posterior, first we need to calculate the normalizing constant, $P(O)$. We have $P(O) = \sum_i P(H_i)P(O|H_i)$.
Coding Exercise 3. In **Bayes** write a function **norm_constant** that takes as input an observation and returns the normalizing constant.

Now we can calculate the posterior, $P(H|O)$.
Coding Exercise 4. In **Bayes** write a function **single_posterior_update** that takes as input an observation and priors and returns the posterior probabilities.

We should now be able to answer our initial question:
Question 1. At random we pick one of the bowls and, without looking, we take a cookie from the bowl. We observe that the cookie is vanilla. What is the probability that it came from Bowl 1?

Finally, in order for our agent to be able to update its belief over time, we also want to be able to calculate the posterior given a sequence of observations. The agent should compute the posterior belief that is the result of having its priors (given at initialization) and then observing the sequence of observations specified.

There is one complication: with every (e.g., chocolate) cookie that is taken from the bowl, the probabilities of picking a next (chocolate) cookie would be different. To avoid the hassle of having to track how many cookies have already been taken from the bowl, we make the assumption of "sampling with replacement". That is, as soon as one cookie is observed it is being put back in the bowl.
Coding Exercise 5. In **Bayes** write a function **compute_posterior** that takes as input a list of observations and returns the posterior probabilities.

Question 2.[2] At random we pick one of the bowls and, without looking, we take two cookies from the bowl. We observe that one of the cookies is chocolate and the other vanilla. What is the probability that they came from Bowl 2?

2.1 Example main function

```
if __name__ == '__main__':
    hypos = ["Bowl1", "Bowl2"]
    priors = [0.5, 0.5]
    obs = ["chocolate", "vanilla"]
    # e.g. likelihood[0][1] corresponds to the likehood of Bowl1 and vanilla, or 35/50
    likelihood = [[15/50, 35/50], [30/50, 20/50]]

    b = Bayes(hypos, priors, obs, likelihood)
```

[2] Note that it is important that you start calculating this probability from the initial priors ($[0.5, 0.5]$)

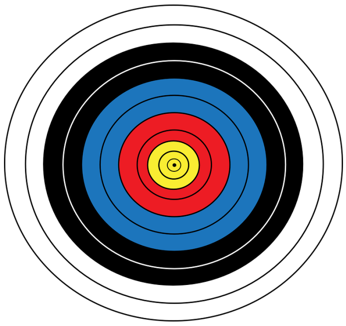

Figure 1: Archery

Table 1: Likelihood table for the archery problem

	Yellow	Red	Blue	Black	White
Beginner	0.05	0.1	0.4	0.25	0.2
Intermediate	0.1	0.2	0.4	0.2	0.1
Advanced	0.2	0.4	0.25	0.1	0.05
Expert	0.3	0.5	0.125	0.05	0.025

```
l = b.likelihood("chocolate", "Bowl1")
print("likelihood(chocolate, Bowl1) = %s " % l)

n_c = b.norm_constant("vanilla")
print("normalizing constant for vanilla: %s" % n_c)

p_1 = b.single_posterior_update("vanilla", [0.5, 0.5])
print("vanilla - posterior: %s" % p_1)

p_2 = b.compute_posterior(["chocolate", "vanilla"])
print("chocolate, vanilla - posterior: %s" % p_2)
```

3 The Archery Problem

This time we are watching an archer shoot at a target, we want to figure out if the archer is at a beginner, intermediate, advanced, or expert level. We use a uniform prior, i.e. $P(\text{beginner}) = P(\text{intermediate}) = P(\text{advanced}) = P(\text{expert}) = 0.25$. We know the probabilities of hitting either the yellow, red, blue, black, or white part of the target, the likelihood is shown in Table 1. We are watching an archer and observe the following: Yellow, white, blue, red, red, blue. With this information we should be able to initialize a new **Bayes** instance.
Coding Exercise 5. Initialize a new **Bayes** instance for the Archer Problem.

Given the observations, yellow, white, blue, red, red, blue, we can answer some questions:
Question 3 What is the probability that the archer is at an intermediate level?
Question 4 What is the most likely level of the archer?