Using Mathematics with Python

Student Number 201503311

Abstract

This report will look into the mathematics within Python which allow it to be used to perform various forms of data analysis on a given data set. This is how we're able to make sense of the data we're given and then, for example, use it to provide evidence to a prediction of how the future *"should"* unfold.

1 Introduction

What is Bayesian analysis in Python? In this article, you will learn about the mathematics for Bayesian statistics that goes on behind the scenes in Python modules. What methods and statistics are used to build up our foundation which we can then start to work with the mathematics to analyse our data to prove, and sometimes disprove, a given hypothesis.

2 Frequentist and Bayesian Statistics

There are two main ways for which we can view probability. One is the **Frequentist** where the only way a probability is relevant is as the limit of successes in a sequence of trials

$$\hat{p} = \lim_{n \to \infty} \frac{k}{n}$$

Here k is the number of successes and n is the number of trials. But, it wouldn't make sense to link a probability distribution with a parameter, because it doesn't make sense to ask ourselves *"what is the probability distribution on k conditional on the observed value $\hat{p}$?"*. This means that if we were to find the confidence interval we would view the ends as random variables and discuss *"the probability of the confidence interval including the actual parameter"*.

The other is the **Bayesian** where we see a probability distribution as a way to measure the uncertainty of the world. Thus we can discuss probability distributions of parameters, regardless that the parameter is **fixed**, because our knowledge of the actual value might be limited. For what we have above, we can flip the probability distribution $p(k \mid \hat{p})$ using Bayes' theorem, to get

$$\overbrace{p(\hat{p} \mid k)}^{\text{posterior}} = \frac{\overbrace{p(k \mid \hat{p})}^{\text{likelihood}}}{\underbrace{p(k)}_{\text{Evidance}}} \overbrace{p(\hat{p})}^{\text{prior}} \tag{1}$$

Now there is a slight technicality here, we have to bring in the *prior* to perform our analysis. This represents our *"assumption"* for the value of p before viewing the value k. The prior is therefore frowned upon by the frequentist because there is the arguable idea that it brings in subjectivity into the strict manner of probability. But, then again, a prior can be just another

assumption made while modelling the situation. Which would then make it just as objective as any of the other assumptions, like the likelihood... But, moving on.

In the Bayesian view, it is not seen as a confidence intervals any more but rather a **credible interval**. These are a more natural way to interpret the interval. An example would be, *"given a 90% credible interval, there is a probability of 90% that the parameter is within the interval"*.

3 The Prior, Likelihood and Posterior

- The **prior** is what reflects the information that we know about the value of some parameter before viewing the data k. If nothing is known about the parameter then we use a **flat prior** which doesn't expose much information.
- The **likelihood** is our way in introducing the data to the process. It represents how *plausible* the data is while considering the parameters.
- The **posterior** is a probability distribution of the parameters in our model. It is what we know about the problem and relates to the prior and the likelihood. If the prior and likelihood are both vague then we'll get a posterior that reflects these vague beliefs. It can be thought of as the updated version of the prior with the given data.

The **evidence** is the probability of getting the data averaged over all possible values. It can be thought of simply as a **normalising factor** and won't be an issue if we neglect because we're not concerned about the true values of the parameters. Thus, we can rewrite Bayes' theorem as a proportionality (A more general form than our example before)

$$p(H \mid D) \propto p(D \mid H)p(H) \tag{2}$$

Where H is the Hypothesis and D is the Data. To gather a true understanding for what each role actually *does*, it'll be beneficial to do a simple example.

4 Simple Example

We can consider a very common example that is also very simple; a coin toss. We toss a coin n times and record the number of heads h and tails t we get. Using just this simple data we can deduce answers to possible questions about the problem like *"Is the coin fair?"*. So, to do this, we'll need our data, which is assumed we already have, and a model.

4.1 The Model

Let's set up an understanding for how the bias can be for this problem. A coin a bias of 1 will always land on heads. A coin with a bias of 0 will always land on tails. A coin with a bias of 0.5 will land on heads and tails evenly. We'll use the parameter θ for the bias. Then we have the following formula (as seen in §3 equation (2))

$$p(\theta \mid h) \propto p(h \mid \theta)p(\theta) \tag{3}$$

Now, we need to choose which prior and likelihood we should use.

4.1.1 Choosing the Likelihood

Firstly, let's assume that each coin toss is independent of each other. And that only heads or tails in a possible outcome (no side shots). Then a good distribution for the likelihood is the **binomial distribution** because this is a discrete probability distribution for the number of heads in a sequence of independent flips. This is given by:

$$p(h \mid \theta) = \frac{n!}{h!(n-h)!}\theta^h(1-\theta)^{(n-h)} \tag{4}$$

This gives us the probability (or likelihood) of getting h heads out of n flips, given a fixed bias θ.

4.1.2 Choosing the Prior

For this problem it will be wise to use the **beta distribution**. This distribution is good for our parameter θ because it is restricted between 0 and 1 just like the bias. The first term normalises the distribution so that it will always integrate to 1. It has two parameters, α and β, that control the distribution. How it looks is as follows:

$$p(\theta) = \frac{\Gamma\big(\alpha+\beta\big)}{\Gamma\big(\alpha\big)\Gamma\big(\beta\big)}\theta^{\alpha-1}\big(1-\theta\big)^{\beta-1} \tag{5}$$

This distribution is something called a **conjugate prior** to the binomial distribution. This means that we will get a beta distribution as a posterior, since the likelihood is a binomial distribution and the prior is the beta distribution. This makes the posterior capable of being easily controlled mathematically. So that we don't end up with a posterior that we cannot solve. This is an older practice and now, because of computational improvements, this isn't as much of an issue as we could solve numerically with the help of Python. But it'll make things easier for now.

4.1.3 Getting the Posterior

This is where we start cooking! (mathematically, of course) Recall equation (3) that says the posterior is proportional to the likelihood multiplied by the prior. So, by using equations (4) and (5) we get the following:

$$p(\theta \mid h) \propto \left(\overbrace{\frac{n!}{h!(n-h)!}\theta^h(1-\theta)^{(n-h)}}^{\text{Likelihood}}\right)\left(\overbrace{\frac{\Gamma\big(\alpha+\beta\big)}{\Gamma\big(\alpha\big)\Gamma\big(\beta\big)}\theta^{\alpha-1}\big(1-\theta\big)^{\beta-1}}^{\text{Prior}}\right)$$

To simplify this expression we can drop the terms that do not depend on θ as we only care about the practical aspect. So we rewrite to find

$$\begin{aligned} p(\theta \mid h) &\propto \Big(\theta^h(1-\theta)^{(n-h)}\Big)\Big(\theta^{\alpha-1}\big(1-\theta\big)^{\beta-1}\Big) \\ &\propto \theta^{(\alpha-1)+h}\big(1-\theta\big)^{(\beta-1)+(n-h)} \end{aligned}$$

There is a relation between this form and the form of the beta distribution.

$$\alpha_{\text{posterior}} = \alpha_{\text{prior}} + h, \quad \beta_{\text{posterior}} = \beta_{\text{prior}} + (n-h)$$

This means that the posterior is the beta distribution.

$$p(\theta \mid h) = \text{Beta}\big(\alpha_{\text{prior}} + h,\, \beta_{\text{prior}} + (n-h)\big)$$

4.2 Computing the Posterior

Using the above analytical expression, we can compute the final result using Python. For example, let's set $\theta = 0.4$, the number of flips $n = 50$ and the number of times we get heads $h = 18$. In a real case, we do not know what the true value of θ is and for any old coin we would go in assuming that the coin is fair. So we will set our prior to follow this belief. So, we believe that over 50 experiments, we will get $h = 25$ and $t = 25$. Then our prior is $\text{Beta}(25, 25)$. Now, in our real findings, we have that over 50 experiments, we received $h = 18$ and $t = 32$. Then our posterior is $\text{Beta}(25 + 18, 25 + 32) = \text{Beta}(43, 57)$ (same as above). We can then calculate and plot the results with Python.

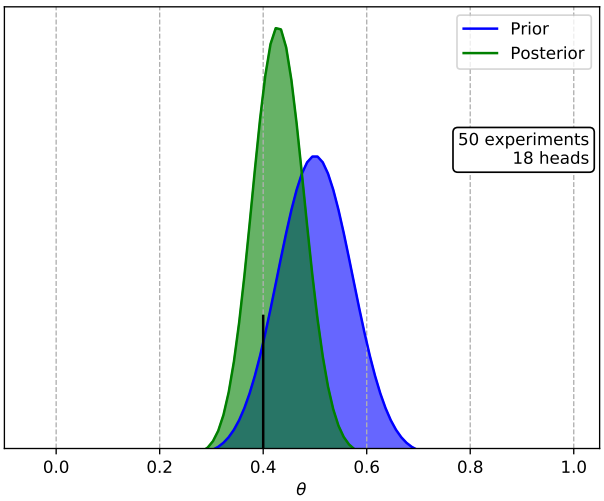


Figure 1: Plot of $p(\theta \mid h)$ for set successive experiments.

The posterior is moving towards the left which would signify that the coin isn't fair and that the bias $\theta < 0.5$. We could approximate the bias as being $\theta \approx 0.43$ so the coin has a sightly higher probability of landing on tails than heads with each flip. It can give us a much better understanding for why the prior is to be chosen with caution if we see the results we get for the posterior when we choose a poor prior.

Let's say we choose the prior to be the belief that over 50 experiments we will get 40 heads. Then our prior is $\text{Beta}(40, 10)$. We shall keep the same data as 50 flips and 18 heads. But this poor prior will result in our posterior being $\text{Beta}(58, 42)$. Obviously, with some understanding and experience of a coin flip, you would never choose a prior like this as it would suggest that you believe the coin has a bias of $\theta = 0.8$. But let us say that maybe a Martian has come whom isn't very familiar with what a coin is and chooses this poor prior. Now we can repeat the process in Python and see the results.

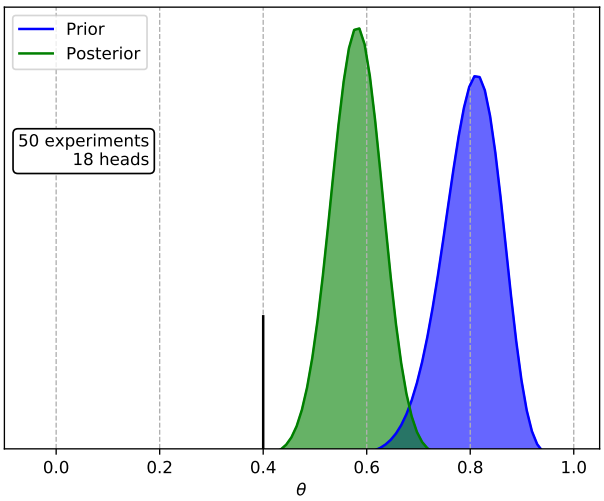


Figure 2: Plot of $p(\theta \mid h)$ with a poor prior.

We now see what we suggested before, that the bias would be $40/50 = 0.8$ just as the peak of the prior is very close to that value of θ. The error in the prior then results in error with out posterior. It is giving us the analysis that the coin has a bias of $\theta = 0.6$ which we secretly know isn't right. Although, one thing is that the separation or difference in the distribution is greater than before which could point out that the prior is far off from the true value of θ. So we could then adjust the prior to out *new* belief that the bias of the coin is less than 0.8.

The posterior is our updated prior given the data above. The black vertical line at $\theta = 0.4$ is there just to illustrate the true value that we have assigned to the bias of the coin. In true practice this would be unknown and we would use the posterior to make a decision on the most plausible value of the bias θ which is pretty close to at around $\theta \approx 0.38$. This figure gives us an understanding about Bayesian analysis, we see that:

- The result is a posterior distribution, not a number. It gives us a distribution on the probability with our data.
- The most plausible value is the peak of the distribution as highlighted before.
- The spread of the posterior distribution is proportional to the uncertainty of the value for θ. The less spread, the more certain the value.
- How quickly different posteriors converge to the same distribution depends on the data and the model. After 50 flips the blue curve is spread away from the posterior.
- The posterior we get from computing with 150 flips once will be the same as if we computed the posterior 150 times with each one adding one extra observation and using the previous posterior as the new prior.

5 Linear Example

Here we will use what was discussed in §4 for a more less common example while still being simple in approach. We'll also use more of Python to solve our problem and then discuss the mathematics being used in the background. Our problem is that we are given data and we want to find the original graph. We have the assumption that the original graph is a linear

regression and will be of the form: $y = mx + c$. Although there is something extra. This is an exact analytical equation. Our problem will also include a random term called noise. So our problem is to solve the linear equation:

$$y = mx + c + \varepsilon \quad \text{where } m \text{ is the gradient, } c \text{ is the intercept and } \varepsilon \text{ is some noise}$$

Now we need to start setting up our model.

5.1 The Model

We have some data from our mystery linear equation. Now, using this we want to find the equation that produced the data. It's always most useful to plot the data first so that we can make some educated assumptions.

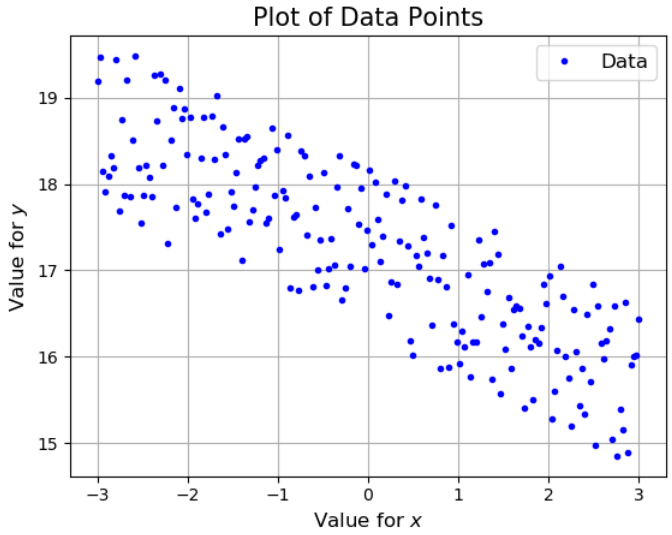


Figure 3: Plot of our given data coordinates

Now we can start to investigate further. Firstly, we can make the assumption that our linear equation will have a negative gradient. There is a general slope in the data from top left to bottom right. Next, we can see that our data passes through the y-axis between $16 < y < 19$. So we can assume the the intercept is positive. Combining these two observations will tell us that our equation is of the form: $y = -mx + c$. We will need to conduct Bayesian analysis on our data to conclude what value these two unknowns are, but making these observations are simple yet necessary to build up our prior.

5.1.1 Choosing the Prior

It will be helpful to look into what we will then go on to make as our prior. we can play around with what we have assumed above and plot some values as you will see below. If we try and approximate the centre value at each end of the data cluster then we can think of these as two points. The points will be where our lines cross. Then, if we were to plot a line through these two points, we will plot a linear line that will give us a visual idea of what the real linear equation should look like. Observe:

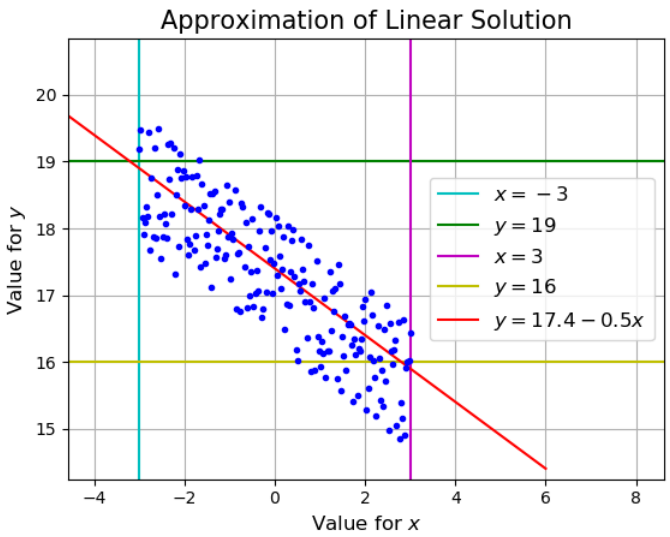


Figure 4: Plot of approximate values

So now, after looking at our plot, we see that the intercept should be around $c \approx 17.4$ and that the gradient is around $m \approx -0.5$. This is just some clarification mixed with some common sense. We will now go on to decide on our prior for our model but in the back of our minds we will, to some extent, expect to get values like this.

5.1.2 The Intercept

We need to think about the best distribution to choose for the parameters we want to find. The intercept, c, is a number that will lie between $16 < y < 19$. So the minimum is 16 and the maximum is 19 and we have our x values which is random but it is **contained** within these values. So I shale choose the prior for the intercept to be a uniform distribution.

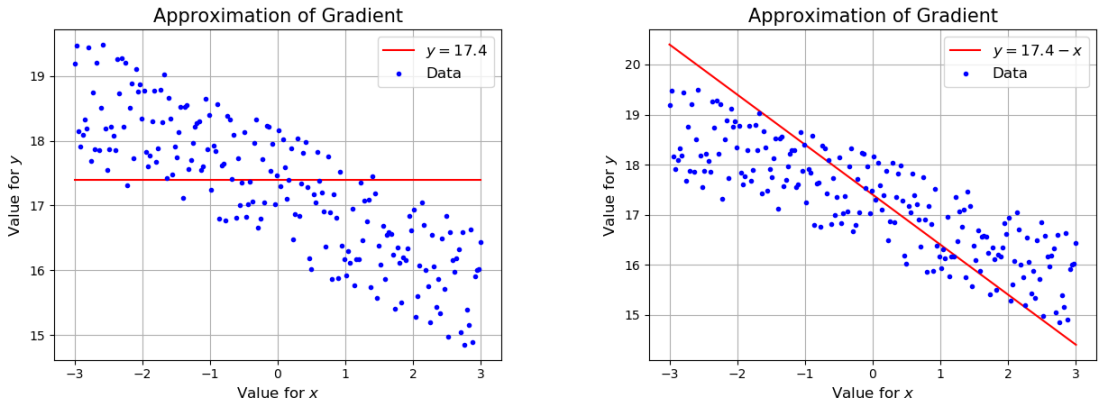
$$c \sim U(16, 19)$$

5.1.3 The Gradient

The gradient, m, is *again* a number that will lie between -1 and 0. So, like before, we have a minimum and maximum with our x values between so we will also have our gradient prior as a uniform distribution.

$$m \sim U(-1, 0)$$

If we observe what these different graphs look like when compared to the data, it will help to see what has why we chose these values for our uniform priors and how our uniform distribution will work. By choosing these two values, our gradient will *converge* to an estimation of the gradient using Markov chain Monte Carlo (MCMC). It is the same process for the intercept as well. Our two comparisons are seen below:



(a) Plot with $m = 0$ (b) Plot with $m = -1$

Figure 5: Comparison of data with different gradient lines

5.1.4 Getting the Posterior