Bayesian Statistics - Introduction

Introduction

In this section, you'll investigate the Bayesian statistical framework. Bayesian statistics are an alternative perspective to classical Frequentist approaches which you've seen thus far. Bayesian statistics applies reasoning to unknown probabilities in a manner in which the Frequentist approach does not allow.

Thomas Bayes

Bayesian statistics owes its name to the famous mathematician Thomas Bayes. Born (sometime) in the early 1700s, he bucked many academic traditions of his time due to his families religious beliefs. Cambridge and Oxford were known for the most prestigious mathematics of the time, but Bayes was a Presbytarien barring him from these universities which had ties to the Church of England.

Bayes' theorem

Bayes' theorem is a method for rewriting conditional probabilities. The formula is:

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

In the following lessons, you'll learn more about two traditional interpretations of this formula. The first provides an intuitive understanding, viewing the numerator as the probability of both A and B occuring:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

This should make perfect sense: the probability that A is true, given B is true, is the probability that A and B are both true, divided by the probability that B was true in the first place.

The second interpretation of Bayes theorem leads straight into the Bayesian statistical framework itself, bringing about discussions of priors, likelihoods, and posterior probabilities.

Summary

Get ready to jump in! This section provides an exciting introduction to Bayes theorem and Bayesian statistics, further rounding out your statistical toolbox!

Bayesians vs Frequentists

Introduction

Up until now, all of the statistical theory you have encountered has been through the lens of a Frequentist. This has included discussions of z-tests, t-tests, p-values, and ANOVA; all are from the Frequentist perspective. In this lesson, you'll start to explore an alternative perspective donned by Bayesians.

Objectives

You will be able to:

- Compare the Bayesian v. Frequentist statistical frameworks

Philosophical Interpretations

A natural place to start when outlining the differences between Bayesians and Frequentists is to talk of their interpretation of probability itself. For Frequentists, the probability of an event is the limit of the rate of occurrences of the event if the same scenario including context and assumptions were repeated ad infinitum. In contrast, Bayesians interpret probability as the level of confidence, or belief, in a particular event occurring. In many ways, this makes a more natural interpretation for rare events that cannot possibly reoccur in the same context and circumstances.

Practical Implications

The practical implications of Bayesians versus Frequentists rest upon making assumptions about unknown quantities. In the Bayesian framework, you make assumptions about unknown variables which you are attempting to estimate. For example, you might assume that the number of individuals who will buy a product can be represented by a binomial variable with parameter p. In contrast, the Frequentist perspective does not allow embedding of prior beliefs such as this into statistical experiments and analyses.

Summary

In this lesson, you started to learn about the differences in Bayesian versus Frequentist statistical perspectives. Keep in mind that there are not always rigid lines between these modes of thought. Nonetheless, the discussion is a worthwhile one in considering which approach you wish to take in your research and analysis.

Bayes' Theorem

Introduction

Bayes theorem is an indispensable law of probability, allowing you to deductively quantify unknown probabilities. The theory rests upon conditional probability. Let's take a look at it in practice.

Objectives

You will be able to:

- Define Bayes' theorem in relation to conditional probabilities
- Identify examples of applications of Bayes' theorem

Bayes' formula

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

Breaking the formula apart

Bayes' theorem is quite intuitive, decomposing the conditional probability of 'A given B' in terms of the probability that both events are true divided by the probability that B is true. Bayes theorem takes this natural idea a step further, expressing the probability that both events are true as a conditional probability multiplied by the condition itself.

To recap:

Bayes' Theorem takes the definition of the conditional likelihood:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

and rewrites the $P(A \cap B)$ as $P(B|A)P(A)$, which makes perfect sense; the probability of B given A is true, multiplied by the probability that A is true, gives us the probability that both are true.

Making this substitution, you have Bayes' Theorem:

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

A simple example

Let's take a simple theoretical example to demonstrate. Imagine there are two fish tanks at the local pet store. The small tank holds 10 Betta fish. The large tank has 200 goldfish and 35 Betta fish. Given that a fish is a Betta fish, what's the probability it comes from the small tank?

On the one hand, it seems that if you were to select a fish from the large tank, you'd probably end up with a goldfish. However, because these tanks are of such vastly different sizes, the probability that the fish came from the larger tank is actually more probable.

Using Bayes' theorem, you are looking to find the probability that the fish came from the small tank, given that it is a Betta fish:

$$P(\text{small\_tank} \mid \text{Betta\_fish}) = \frac{P(\text{Beta\_fish} \mid \text{small\_tank})P(\text{small\_tank})}{P(\text{Beta\_fish})}$$

Furthermore, you know:

$P(\text{Beta\_fish} \mid \text{small\_tank}) = 1$

$$P(\text{small\_tank}) \quad \frac{\text{number of fish in small tank}}{\text{number of all fish}} \quad \frac{10}{245}$$

$$P(\text{small\_tank}) = \quad \text{number of all fish} \quad = 245$$

$$P(\text{Beta\_fish}) = \frac{45}{245}$$

Giving you:

$$P(\text{small\_tank} \mid \text{Betta\_fish}) = \frac{1 \cdot \frac{10}{245}}{\frac{45}{245}}$$

$$P(\text{small\_tank} \mid \text{Betta\_fish}) = \frac{10}{45}$$

While concrete, this example fails to demonstrate the full power of Bayes' theorem since you had all of the underlying information, so you don't even need to use Bayes' theorem. You could have simply looked at the number of Betta fish in the small tank versus the number of Betta fish overall:

$$\frac{10}{45}$$

giving you exactly the same result.

An NLP example

With this simple example out of the way, let's examine a more practical example from the field of Natural Language Processing.

A common introductory example to Natural Language Processing or classification is detecting spam. While you may enjoy spam in a can, you probably don't enjoy getting spam in your inbox. Bayes' theorem can serve as a natural classification method in these scenarios. Assume that the word "offer" (as in Special Offer, We Have an Offer for You, or Don't Miss This Offer!) occurs in 73% of the spam messages you receive. In comparison, only 10% of your desired mail contains the word "offer". If 20% of the messages you receive are spam, and you receive another message with the word "offer", what is the probability that it is spam?

As you might have guessed, you can solve this using the Bayes' theorem!

First, set up the problem:

$$P(\text{Spam} \mid \text{Offer}) = \frac{P(\text{Offer} \mid \text{Spam})P(\text{Spam})}{P(\text{Offer})}$$

Then substituting some of the immediate knowledge we have from the scenario:

$$P(\text{Spam} \mid \text{Offer}) = \frac{.73 \cdot .20}{P(\text{Offer})}$$

Finally, the probability of receiving an email with the word "offer", $P(\text{Offer})$, can be evaluated by decomposing it into the two subsets spam and not spam:

$$P(\text{Offer}) = P(\text{Spam}) \cdot P(\text{Offer} \mid \text{Spam}) + P(\sim\text{Spam}) \cdot P(\text{Offer} \mid \sim\text{Spam})$$
$$P(\text{Offer}) = .20 \cdot .73 + .8 \cdot .10$$
$$P(\text{Offer}) = .146 + .08$$
$$P(\text{Offer}) = .226$$

Finally, substituting this into the original Bayes formula you have:

$$P(\text{Spam} \mid \text{Offer}) = \frac{.73 \cdot .20}{P(\text{Offer})}$$
$$P(\text{Spam} \mid \text{Offer}) = \frac{.73 \cdot .20}{.226}$$
$$P(\text{Spam} \mid \text{Offer}) = .6460$$

As you can see, while spam has a much higher occurrence of the word "offer", the presence of the word alone does not provide strong confidence that the message is spam. To provide more statistical power, you will eventually extend Bayes' theorem to multiple observations simultaneously using the relative probabilities of multiple words.

Summary

In this lesson, you were introduced to the Bayes' theorem, and saw how it can be used to quantify conditional probabilities. With that, let's turn to some more simple examples for you to practice and deepen your understanding.

that, let's turn to some more simple examples for you to practice and deepen your understanding.

Maximum Likelihood Estimation (MLE)

Introduction

"Parameter Inference" is one of the most important concepts of predictive machine learning. In this lesson, you will begin to build an intuition surrounding the ideas around this concept. You'll first look at the maximum likelihood estimation (MLE) for the posterior probability based on observed data. (A direct application of Bayes theorem.) From there, you'll conduct a random experiment involving a series of coin tosses to derive the general formula for MLE of a binomial distribution.

Objectives

You will be able to:

- Describe the process of parameter inference
- Define likelihood and compare it to probability
- List the two assumptions of Maximum Likelihood Estimation
- Describe how Maximum Likelihood Estimation is related to parameter estimation

Parameter Inference

Parameter Inference is the process of probabilistically inferring parameter(s) for a model of our choice, that is which parameter values best describe the underlying dataset, used in an analytical context. Let's try to understand this with a simple experiment with a 10 times coin flip and inspecting the outcome.

In [1]:

```python
import random
def coinToss():
    number = int(input("Number of times to flip coin: "))
    recordList = []
    heads = 0
    tails = 0
    for amount in range(number):
         flip = random.randint(0, 1)
         if (flip == 0):
              print("Toss", amount+1 ,':' , "Heads")
              recordList.append("Heads")
         else:
              print("Toss", amount+1 ,':' , "Tails")
              recordList.append("Tails")
    print(str(recordList))
    print(str(recordList.count("Heads")) + str(recordList.count("Tails")))
    return recordList
```

In [2]:

```python
lst = coinToss()
```

```
---------------------------------------------------------------------------
StdinNotImplementedError                  Traceback (most recent call last)
<ipython-input-2-49ab0443fb3a> in <module>
----> 1 lst = coinToss()

<ipython-input-1-e936f9989e40> in coinToss()
      1 import random
      2 def coinToss():
----> 3     number = int(input("Number of times to flip coin: "))
      4     recordList = []
      5     heads = 0

C:\Anaconda3\envs\learn-env\lib\site-packages\ipykernel\kernelbase.py in raw_input(self, prompt)
    853         if not self._allow_stdin:
    854             raise StdinNotImplementedError(
--> 855                 "raw_input was called, but this frontend does not support input requests."
    856             )
    857         return self._input_request(str(prompt),
StdinNotImplementedError: raw_input was called, but this frontend does not support input requests.
```

Remember its a random experiment so the output will change everytime you run it. Here is the output sequence we'll use in this lesson:

```
['Heads', 'Heads', 'Tails', 'Tails', 'Tails', 'Heads', 'Tails', 'Heads', 'Heads', 'Heads']
```

Considering its a random experiment, you can say that there has to be *some* underlying parameter for the outcome of a coin flip. Also, consider other random experiments with dice rolls. Can you identify a parameter that determines the outcome of such experiments?

Parameter Inference is all to do with identifying that parameter with its optimal value. The first key step in this process is Maximum Likelihood Estimation (MLE).

Maximum Likelihood Estimation

MLE primarily deals with **determining the parameters** that **maximize the probability of the data**. Such a determination can help you predict the outcome of future experiments, e.g., if we toss the coin 1 more time, what is the probability of seeing a head?

- It's a fair coin so the probability is 0.5

This is a safe assumption as it assumes independence between coin flips and hence past events have no impact on future ones.

In [3]:

```
p_head = lst.count('Heads')/10
p_head
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-3-66dec4b3d86a> in <module>
----> 1 p_head = lst.count('Heads')/10
      2 p_head

NameError: name 'lst' is not defined
```

With both these approaches in hand, let's see which answer is more suitable by creating a general case from this example. You want to know the probability of 11th flip $p_{11}(f_{11})$, being a head so you can write:

> $p_{11}(f_{11} = \text{Heads})$

You can also write above for calculating the probability of ith flip being a head:

> $p_i(f_i = \text{Heads}) = \theta_i$

Here θ_i is the parameter that governs the outcome of *ith* flip. To signify that the probability distribution depends on θ_i, you can use conditioning as you saw earlier and write down the last equation to show the probability distribution function along with its dependence on theta_i.

> $p_i((f_i = \text{Heads}) | \theta_i)$

The probability of seeing heads in the ith flip , given theta_i

This makes sense so far, but raises a few confusing points: If the data depends on theta parameter, then the first ten coin flips f_1 to f_10 depend on theta_1 to theta_10 for i = 1 to 10. So looking at the outcome of first ten experiments, how can we extrapolate it to theta_11?

Here's how you can do this—if you say that random outcome of a sequence of flips is governed (or modeled) by the parameters theta_1 to theta_10, you can calculate the probability function based on observed data as:

> $P(\text{Heads, Heads, Tails, Tails, Tails, Heads, Tails, Heads, Heads, Heads}) | \theta_1 \theta_2 .. \theta_{10})$

This is where Maximum Likelihood Estimation steps into the equation. The problem you have now is that you need to find values of

This is where Maximum Likelihood Estimation steps into the equation. The problem you have now is that you need to find values of thetas 1 to 10. MLE helps find theta_i's such that that probability function shown above is **as high as possible** and this is the basic principle of MLE.

Likelihood - The probability of data

MLE looks at the probability of data and it tries to find those parameters (i.e. theta_1 through theta_10 in above case) that maximize the likelihood of this sequence occurring.

> With maximum liklihood estimation, we want to choose those parameters under which our observations become most **likely**.

Going back to our coin flip example. If in our understanding, the coin flips do not affect each other, i.e., they are independent (the outcome of first flip does not affect the outcome of the second flip):

> $P(H, H, T, T, T, H, T, H, H, H) | \theta_1 \theta_2 . . \theta_{10})$
>
> $= P(F_1 = H|\theta_1). P(F_2 = H|\theta_2). . P(F_{10} = H|\theta_{10})$
>
> $= \prod_{10 i=1} p_i(F_i = f_i|\theta_i)$ - The general case for coin flip

Note: ∏ signifies the product over a series, shown in the previous equation, just as Σ denotes summation over a series.

MLE assumptions

Note here that the **independence assumption** allows you to simplify the complex likelihood term into ten simpler factors that can be shown through a general notation in the last equation.

The independence assumption allows simplification of the likelihood term but you still don't have theta_11 in the equation.

There is another assumption you can introduce, based on the fact that the coin does not change significantly after each flip i.e.:

- **The flips are quantitatively same, i.e., they are identically distributed**.

This implies that the flips are taking place under similar circumstances, you can assume that the parameter governing the flips is one and same i.e. just the θ without any subscripts. Based on this assumption, you can rewrite above equation as :

> $\prod_{10 i=1} p_i(F_i = f_i|\theta_i) = \prod_{10 i=1} p(F_i = f_i|\theta)$

This assumption leads you to believe that the 10 flips are governed by the same parameter theta. You now have just one parameter governing the entire sequence of coin flips, and that includes the 11th flip as well.

This is how MLE allows you to connect first 10 coin flips to the 11th coin flip and is the key for inference.

> The two assumptions you made are used so often in Machine Learning that they have a special name together as an entity : "The i.i.d. assumption" i.e. Independent and Identically distributed samples.

This means that the 10 flips are independent and identically distributed which is great as it will allow you to explicitly write down the likelihood that you are trying to optimize.

Remember that theta was defined as the probability of the flip showing up heads; the probability of the sequence w.r.t. theta can now be formulated as:

$\prod_{10 i=1} p(F_i = f_i|\theta)$

$= \theta\theta(1 - \theta)(1 - \theta)(1 - \theta)\theta(1 - \theta)\theta\theta\theta$

$= \theta^6(1 - \theta)^4$

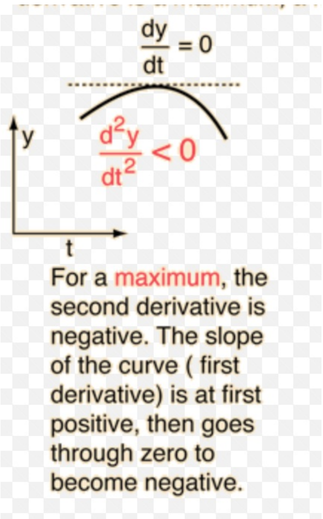
- theta = Probability of seeing a head
- 1 - theta = Probability of seeing a tail
- The sequence: H, H, T, T, T, H, T, H, H, H

You see here the i.i.d. assumptions simplifies the likelihood function to a simple polynomial; to a point where you can **start optimizing the function for the parameter theta**.

This simplified polynomial expression can be interpreted as a function of theta i.e.,

> f(θ)

Now you want to find out the maxima (maximum likelihood) of this function.



For a maximum, the second derivative is negative. The slope of the curve (first derivative) is at first positive, then goes through zero to become negative.

Following the intuition in the image above, you can achieve this theta by taking the derivative

> dfd(θ)

Set this zero, and solve for theta. Then verify the critical point i.e. maxima, by inserting it into the second derivative of f(θ). This is a simple approach, however, the application of product rule repeatedly in this process could be a technically challenging process. This calculation can be simplified using a monotonic function.

Monotonic function

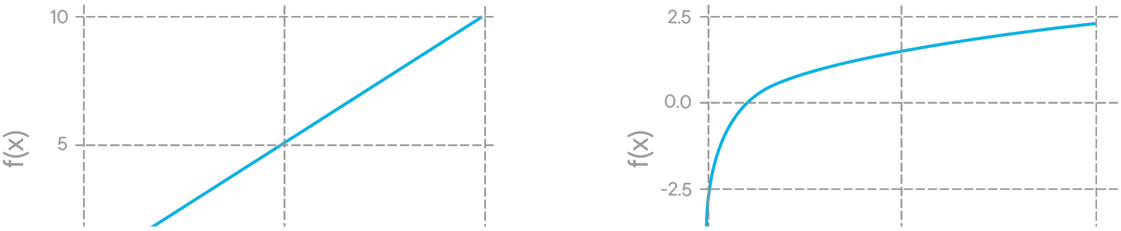
> In mathematics, a monotonic function (or monotone function) is a function between ordered sets that preserves or reverses the given order. This concept first arose in calculus, and was later generalized to the more abstract setting of order theory.

According to this theory, if you apply a monotonic function to another function, like the one you are trying to optimize above, this application will preserve the critical points (maxima in this case) of the original function. Logarithmic functions are normally used within the domain of machine learning to achieve the functionality of monotonicity. The logarithmic function is described as:

> $\log_b(x)$

- where b is any number such that b > 0, b ≠ 1, and x > 0
- The function is read "log base b of x"

The logarithm y is the exponent to which b must be raised to get x. The behavior of a log function can be understood from following image.



(a) f(x) = x

(b) f(x) = 1n(x)

This helps you realize that **log of f(θ) i.e. log(f(θ)) will have the save maxima as the likelihood function f(θ).** This is better known as the **log likelihood**.

Thus, the optimization function i.e. $\theta^6(1-\theta)^4$, that you're trying to optimize w.r.t. theta can be written down as:

> $\text{argmax}\theta\theta^6(1-\theta)^4$
>
> In mathematics, the arguments of the maxima (abbreviated arg max or argmax) are the points of the domain of some function at which the function values are maximized.

Remember that you are not concerned with the actual maximum value of the function. You want to **learn the value for theta** where the **function has the maximum value**.

Following the monotonicity principle, the argmax function can be written with natural log *ln* as:

> $\text{argmax}\theta\ln(\theta^6(1-\theta)^4)$
>
> $= \text{argmax}\theta 6(\ln(\theta)) + 4(\ln(1-\theta))$

Let's call our log likelihood function g(θ), take its derivative and set it to zero.

> $dd\theta_{[g(\theta)]} = |H|^{1\theta} + |T|^{11-\theta}(-1)$

|T| are the number of tails = 4 |H| are the number of heads = 6

You are simply solving for a general case here , so use |T| and |H|

> $|H|^{1\theta} + |T|^{11-\theta}(-1) = 0$

> $|H|(1-\theta) - |T|\theta = 0$
>
> $\theta = |H||H|+|T|$

This is the Maximum Likelihood Function θ_MLE for any given sequence of coins.

$\theta_{MLE} = |H|\,|H| + |T|$

For the initial problem, where H = 6 and T = 4, you get MLE for theta as 6/10 = 0.6 , or , 60% chance of seeing a head for the 11th coin given the data from first 10 coin flips.

> This maximum is called the **MLE for theta** as it makes the observed sequence **most likely**.

Limitations of MLE

Consider a scenario where you get this sequence by total chance: [T, T, T, T, T]. According to the derived MLE formula, the probability of seeing a head at 6th coin toss would be zero. This demonstrates how MLE heavily depends on past data to find the likelihood function. It also indicates that MLE is only a first step for parameter estimation. We shall come across more sophisticated approaches like Maximum Aposteriori Estimate (MAP) and Fully Bayesian Analysis.

Additional Resources

This section was pretty math heavy and included many new concepts like optimization, maximas and minimas, monotonicity, and log functions. With that, take some time to go through following resources to see more example of MLE calculation and get a deep dive into the underlying mathematical theory.

- Probability Concepts Explained: Maximum Likelihood Estimation - Example for calculating MLE with normal distributions.
- IID Statistics: Independent and Identically Distributed
- Monotonically Increasing and Decreasing function: An algebraic approach
- Logarithm Functions

Summary

In this lesson, you began to develop an intuition surrounding MLE. You saw how to use the principle of monotonicity to simplify complex probability calculations into simple arithmetic operations. You also looked at a simple example of a coin toss for MLE. You're well on your way to conducting further complex statistical experiments using Bayesian techniques!

Maximum A Posteriori Estimation (MAP) and Multinomial Bayes

Introduction

Maximum A Posteriori provides a means for estimating a parameter given some prior knowledge about a variable. In it, one assumes a given distribution for the variable and then estimates the parameter itself given additional information. In this lesson, you'll see how Bayes' theorem can be applied in this manner and then extended to multivariate cases.

Objectives

You will be able to

- Identify how Maximum A Posteriori Estimation is related to MLE

Maximum A Posteriori Estimation

Maximum A Posteriori Estimation (MAP) is similar to Maximum Likelihood Estimation but extends this concept by allowing one to also account for prior beliefs regarding the distribution of the variable in question. Recall Bayes' theorem:

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

The Bayesian interpretation of this formula is

$$\text{Posterior} = \frac{\text{Likelihood} \cdot \text{Prior}}{\text{Evidence}}$$

With MAP, you then attempt to optimize a parameter θ for the assumed distribution in order to maximize the posterior probability.

Multinomial Bayes

Multinomial Bayes also extends the notions within Bayes' theorem, allowing one to chain inferences. The primary assumption for this is assuming that your variables are independent of one another. Recall that if you assume two events A and B are independent of one another, then $P(A \cap B) = P(A) \cdot P(B)$. Similarly, if independence is assumed when extending Bayes theorem to a multivariate case, one can multiply the successive probability estimates. Mathematically, this can be summarized as:

$$P(Y|X_1, X_2, \ldots, X_n) = \frac{P(X_1|Y) \cdot P(X_2|Y) \cdot \ldots \cdot P(X_n|Y)}{P(X_1, X_2, \ldots, X_n)} P(Y)$$

Summary

This lesson briefly introduced the concept of Maximum A Posteriori Estimation and extending Bayes' theorem to multivariate cases. In later sections, you'll investigate these ideas in practice, working with practical examples and coding your own implementations to gain a full understanding.

Bayesian Statistics - Recap

Introduction

Well done! You covered a lot of ground in this section. From Bayes' theorem to Maximum Likelihood Estimation (MLE), you now have the foundation to dive into the world of Bayesians!

Bayes' Theorem

To start, you investigated Bayes' theorem and some hypothetical examples.

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

Bayesian Statistics

From there, you then went on to read more about some of the philosophical differences between Bayesians and Frequentists. Bayesians interpret probability as the level of confidence or belief in an event. In contrast, Frequentists view probability as the limit as the number of trials goes to infinity of successes versus trials.

MLE and MAP

In outlining the discussion of Bayesian techniques, you got an introduction to Maximum Likelihood Estimation and Maximum A Posteriori Estimation. In both, you saw methods for optimizing one's beliefs given certain information. This was used to estimate parameters for assumed distributions.

Summary

Again, quite a bit was covered in this section. There are certainly plenty of additional resources available if you wish to further dive into MLE, MAP, or other Bayesian techniques. Bayesian inference can provide a powerful framework for quantifying and reasoning with uncertainty that has continued to gain popularity with additional computing resources.