Bayes' Theorem

Machine Learning

Generic - Sem I

06/02/2020

Presented by: Ms. Sonia Ahlawat

Bayes' Theorem

Bayes' Theorem is formula that converts human belief, based on evidence, into predictions. It was conceived by the Reverend Thomas Bayes, an 18th-century British statistician who sought to explain how humans make predictions based on their changing beliefs. To understand his theorem, we will start by learning its notation.

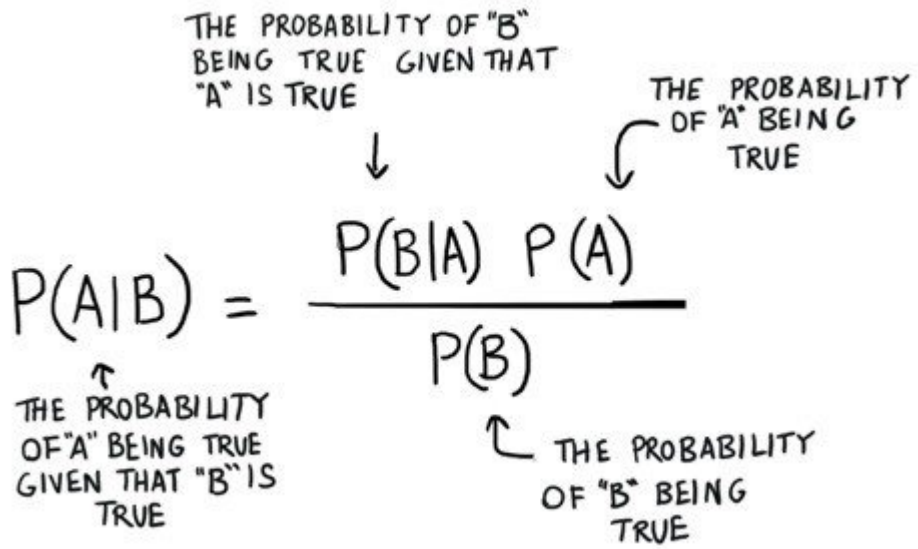
Bayesian Notation

Here's how to read Bayesian notation:

- P(A) means "the probability that A is true."
- P(A|B) means "the probability that A is true *given that B is true*."

In this case, it's easiest to think of B as the symptom and A as the disease; i.e. B is a skin rash that includes tiny white spots, and A is the probability of the measles. So we use phenomena or evidence that is easily visible to calculate the probability of phenomena that are hidden.
What you can see enables you to predict what you can't see.

It turns out that the probabilities of A and B are related to each other in the following manner:



That is Bayes Theorem: that you can use the probability of one thing to predict the probability of another thing. But Bayes Theorem is not a static thing. It's a machine that you crank to make better and better predictions as new evidence surfaces.

An interesting exercise is to twiddle the variables by assigning different speculative values to P(B) or P(A) and consider their logical impact on P(A B)

For example, if you increase the denominator P(B) on the right, then P(A|B) goes down.

Concrete example:

A runny nose is a symptom of the measles, but runny noses are far more common than skin rashes with tiny white spots.

That is, if you choose P(B) where B is a runny nose, then the frequency of runny noses in the general population decreases the chance that runny nose is a sign of measles. The probability of a measles diagnosis goes down with regard to symptoms

that become increasingly common; those symptoms are not strong indicators. Likewise, as measles become more common and P(A) goes up in the numerator on the right, P(A|B) goes up necessarily, because the measles are just generally more likely regardless of the symptom that you consider.

Naming the Terms in the Theorem

The terms in the Bayes Theorem equation are given names depending on the context where the equation is used.
It can be helpful to think about the calculation from these different perspectives and help to map your problem onto the equation. Firstly, in general, the result P(A|B) is referred to as the **posterior probability** and P(A) is referred to as the **prior probability**.

- P(A|B): Posterior probability.
- P(A): Prior probability.

Sometimes P(B|A) is referred to as the **likelihood** and P(B) is referred to as the **evidence**.

- P(B|A): Likelihood.
- P(B): Evidence.

This allows Bayes Theorem to be restated as:

- Posterior = Likelihood * Prior / Evidence

We can make this clear with a smoke and fire case.

What is the probability that there is fire given that there is smoke?

Where P(Fire) is the Prior, P(Smoke|Fire) is the Likelihood, and P(Smoke) is the evidence:

- P(Fire|Smoke) = P(Smoke|Fire) * P(Fire) / P(Smoke)

You can imagine the same situation with rain and clouds.

Now that we are familiar with Bayes Theorem and the meaning of the terms, let's look at a scenario where we can calculate it.

Worked Example for Calculating Bayes Theorem

Bayes theorem is best understood with a real-life worked example with real numbers to demonstrate the calculations.
First we will define a scenario then work through a manual calculation, a calculation in Python, and a calculation using the terms that may be familiar to you from the field of binary classification.

1. Diagnostic Test Scenario
2. Manual Calculation
3. Python Code Calculation
4. Binary Classifier Terminology

Diagnostic Test Scenario

An excellent and widely used example of the benefit of Bayes Theorem is in the analysis of a medical diagnostic test.

Scenario: Consider a human population that may or may not have cancer (Cancer is True or False) and a medical test that returns positive or negative for detecting cancer (Test is Positive or Negative), e.g. like a mammogram for detecting breast cancer.

***Problem**: If a randomly selected patient has the test and it comes back positive, what is the probability that the patient has cancer?*

Manual Calculation

Medical diagnostic tests are not perfect; they have error.
Sometimes a patient will have cancer, but the test will not detect it. This capability of the test to detect cancer is referred to as the **sensitivity**, or the true positive rate. In this case, we will contrive a sensitivity value for the test. The test is good, but not great, with a true positive rate or sensitivity of 85%. That is, of all the people who have cancer and are tested, 85% of them will get a positive result from the test.

- P(Test=Positive | Cancer=True) = 0.85

Given this information, our intuition would suggest that there is an 85% probability that the patient has cancer.

Our intuitions of probability are wrong.

This type of error in interpreting probabilities is so common that it has its own name, it is referred to as

Bayes Theorem for Modeling Hypotheses

Bayes Theorem is a useful tool in applied machine learning.
It provides a way of thinking about the relationship between data and a model.
A machine learning algorithm or model is a specific way of thinking about the structured relationships in the data. In this way, a model can be thought of as a hypothesis about the relationships in the data, such as the relationship between input (*X*) and output (*y*). The practice of applied machine learning is the testing and analysis of different hypotheses (models) on a given dataset.

Bayes Theorem provides a probabilistic model to describe the relationship between data (*D*) and a hypothesis (h); for example:

- P(h|D) = P(D|h) * P(h) / P(D)

Breaking this down, it says that the probability of a given hypothesis holding or being true given some observed data can be calculated as the probability of observing the data given the hypothesis multiplied by the
probability of the hypothesis being true regardless of the data, divided by the probability of observing the data regardless of the hypothesis.

Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.

Under this framework, each piece of the calculation has a specific name; for example:

- P(h|D): Posterior probability of the hypothesis (the thing we want to calculate).
- P(h): Prior probability of the hypothesis.

This gives a useful framework for thinking about and modeling a machine learning problem. If we have some prior domain knowledge about the hypothesis, this is captured in the prior probability. If we don't, then all hypotheses may have the same prior probability.

If the probability of observing the data P(D) increases, then the probability of the hypothesis holding given the data P(h|D) decreases. Conversely, if the probability of the hypothesis P(h) and the probability of observing the data given hypothesis increases, the probability of the hypothesis holding given the data P(h|D) increases.

The notion of testing different models on a dataset in applied machine learning can be thought of as estimating the probability of each hypothesis (h1, h2, h3, … in H) being true given the observed data.

The optimization or seeking the hypothesis with the maximum posterior probability in modeling is called maximum a posteriori or MAP for short.

Any such maximally probable hypothesis is called a maximum a posteriori (MAP) hypothesis. We can determine the MAP hypotheses by using Bayes theorem to calculate the posterior probability of each candidate hypothesis.

Bayes Theorem for Classification

Classification is a predictive modeling problem that involves assigning a label to a given input data sample.
The problem of classification predictive modeling can be framed as calculating the conditional probability of a class label given a data sample, for example:

- P(class|data) = (P(data|class) * P(class)) / P(data)

Where P(class|data) is the probability of class given the provided data.
This calculation can be performed for each class in the problem and the class that is assigned the largest probability can be selected and assigned to the input data.
The priors for the class and the data are easy to estimate from a training dataset, if the dataset is suitability representative of the broader problem.
The conditional probability of the observation based on the class P(data|class) is not feasible unless the number of examples is extraordinarily large, e.g. large enough to effectively estimate the probability distribution for all different possible combinations of values. This is almost never the case, we will not have sufficient coverage of the domain.
As such, the direct application of Bayes Theorem also becomes intractable, especially as the number of variables or features (n) increases.

Naive Bayes Classifier

The solution to using Bayes Theorem for a conditional probability classification model is to simplify the calculation.
The Bayes Theorem assumes that each input variable is dependent upon all other variables. This is a cause of complexity in the calculation. We can remove this assumption and consider each input variable as being independent from each other.
This changes the model from a dependent conditional probability model to an independent conditional probability model and dramatically simplifies the calculation.
This means that we calculate P(data|class) for each input variable separately and multiple the results together, for example:

- P(class | X1, X2, ..., Xn) = P(X1|class) * P(X2|class) * ... * P(Xn|class) * P(class) / P(data)

We can also drop the probability of observing the data as it is a constant for all calculations, for example:

- P(class | X1, X2, ..., Xn) = P(X1|class) * P(X2|class) * ... * P(Xn|class) * P(class)

This simplification of Bayes Theorem is common and widely used for classification predictive modeling problems and is generally referred to as Naive Bayes.

Imagine 100 people at a party, and you tally how many wear pink or not, and if a man or not, and get these numbers:

Bayes' Theorem is based off just those 4 numbers!
Let us do some totals:

	Pink	notPink
Man	5	35
notMan	20	40

Bayes' Theorem is based off just those 4 numbers!

Let us do some totals:

	Pink	notPink	
Man	5	35	40
notMan	20	40	60
	25	75	100

And calculate some probabilities:

- the probability of being a man is P(Man) = *40*/**100** = 0.4
- the probability of wearing pink is P(Pink) = *25*/**100** = 0.25
- the probability that a man wears pink is P(Pink|Man) = *5*/**40** = 0.125
- the probability that a person wearing pink is a man **P(Man|Pink)** = **...**

But alll your data is **ripped up**! Only 3 values survive:

- **P(Man) = 0.4, P(Pink) = 0.25 and P(Pink|Man) = 0.125**

Can you discover **P(Man|Pink)** ?
Imagine a pink-wearing guest leaves money behind ... was it a man? We can answer this question using Bayes' Theorem:
P(Man|Pink) = (*P(Man) * P(Pink|Man)*)/**P(Pink)**
P(Man|Pink) = (*0.4 × 0.125*)/**0.25** = 0.2
*Note: if we still had the raw data we could calculate directly 5/**25** = 0.2*

https://www.youtube.com/watch?v=gJvpOOLnBpg

https://slideplayer.com/slide/16240789/