A Naive Bayes Primer / Naive Bayes for Text Categorization

Nelson Liu nfliu@uw.edu
Jonathan Lee jlee27@uw.edu

November 14, 2016

1 Motivating Problem

Consider the following (hypothetical) problem:

In our dataset, Hillary Clinton has tweeted 4190 times and Donald Trump has tweeted 4246 times. 30% of Clinton tweets contain the word "climate", while only 15% of Trump tweets contain the word "climate". What is the probability that a tweet is written by Trump or Clinton, given that it contains the word "climate"?

We like to assume that our dataset is fairly accurate sample of the population, so we'll say that the numbers of tweets we have for each person is an accurate representation of their tweeting frequency. Let H be the event that a tweet is written by Hillary Clinton, T be the event that a tweet is written by Donald Trump. Thus, we can estimate the probability that an arbitrary tweet is written by Clinton or by Trump from our dataset.

$$\mathbb{P}(H) = \frac{\text{\# Hillary tweets}}{\text{\# total tweets}} = \frac{4190}{4190 + 4246} \approx 0.497$$
$$\mathbb{P}(T) = \frac{\text{\# Trump tweets}}{\text{\# total tweets}} = \frac{4246}{4190 + 4246} \approx 0.503$$

Let C be the event that the tweet contains the word "climate". We want to calculate $\mathbb{P}(H|C)$, the probability that a tweet is written by Clinton given that we know that it contains the word "climate". We can calculate this probability (called the posterior probability) by applying Bayes' Theorem (presented in the next section, so don't worry if you don't know what this is).

$$\begin{aligned}\mathbb{P}(H|C) &= \frac{\mathbb{P}(C|H)\mathbb{P}(H)}{\mathbb{P}(C|H)\mathbb{P}(H) + \mathbb{P}(C|T)\mathbb{P}(T)} \\ &= \frac{0.3 \times 0.497}{0.3 \times 0.497 + 0.15 \times 0.503} \\ &\approx 0.66\end{aligned} \tag{1}$$

It's pretty simple to calculate this probability when we're only examining one word. We would like to extend this sort of thinking to classify entire tweets as either being written by Hillary Clinton or Donald Trump by examining every word in the tweet. This is where naive bayes classifiers come in.

2 Derivation of Bayes' Theorem

This section seeks to give the reader a theoretical understanding of Bayes' Theorem. It's difficult to understand how Bayes' theorem works without knowing what conditional probability is, so we'll begin by giving a recap on that.

2.1 Conditional Probability: A Recap

Given two events A and B, the probability of A happening given that we know that B happens (denoted $\mathbb{P}(A|B)$) is defined by:

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

$\mathbb{P}(A|B)$ is read "the probability of A given B". This equation makes intuitive sense; since you are trying to calculate the probability that A happens and you know that B has happened, you put the probability that B happens in the denominator (this is your "sample space") and the probability that A and B happen in the numerator. Dividing these thus yields the probability that A occurs given that B occurs.

Remark. *Notation may shift occasionally; note that* $\mathbb{P}(AB) = \mathbb{P}(A, B)$

2.2 Derivation

Before we can derive Bayes' theorem, we need a preliminary result, the law of total probability. We present it without proof for the sake of brevity, the the proof can be found online quite easily.

Theorem 2.1 (Law of Total Probability)**.** *Let* $A_1, \ldots, A_n$ *represent events that span the entire sample space (that is, the possible outcome events are* $A_1, \ldots, A_n$*). For any event* B*,*

$$\mathbb{P}(B) = \sum_{i=1}^{k} \mathbb{P}(B|A_i)\mathbb{P}(A_i)$$

In our derivation of Bayes' Theorem, let's continue considering the events $A_1, \ldots, A_n$ where $\mathbb{P}(A_i) > 0$ for each i. With the formula for conditional probability, we rearrange the terms a bit and see that:

$$\mathbb{P}(A_iB) = \mathbb{P}(B) \times \mathbb{P}(A_i|B)$$

This signifies that the probability of A_i and B occurring is the probability of B occurring multiplied by the probability of A_i occurring given that B occurs.

We can similarly rearrange this the other way around:

$$\mathbb{P}(A_i B) = \mathbb{P}(A_i) \times \mathbb{P}(B|A_i)$$

This signifies that the probability of A_i and B occurring is the probability of A_i occurring multiplied by the probability of B occurring given that A_i occurs, which is also a reasonable thing to say.

Because the two equations above equal $\mathbb{P}(A_i B)$, it follows that:

$$\mathbb{P}(B) \times \mathbb{P}(A_i|B) = \mathbb{P}(A_i) \times \mathbb{P}(B|A_i)$$

Now, by dividing $\mathbb{P}(B)$ on both sides, we get the "simple formulation" of Bayes' Theorem:

$$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(A_i) \times \mathbb{P}(B|A_i)}{\mathbb{P}(B)}$$

The simple formulation shows us that the probability of A given B is equal to the probability of A happening multiplied by the probability of B given A, all divided by the probability of B. In reality however, the probability of B is calculated by applying the aforementioned law of total probability. Thus, we can apply the law of total probability to the "simple formulation" to yield the formulation of Bayes' Theorem that is commonly found in literature (simply "Bayes' Theorem" from here on out).

Theorem 2.2 (Bayes' Theorem)**.** *Let $A_1, \ldots, A_k$ represent the events that span the entire sample space such that $\mathbb{P}(A_i) > 0$ for each i. If $\mathbb{P}(B) > 0$ then, for each $i = 1, \ldots, k$,*

$$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(A_i) \times \mathbb{P}(B|A_i)}{\sum_{i=1}^{k} \mathbb{P}(B|A_i)\mathbb{P}(A_i)}$$

Remark. *We call $\mathbb{P}(A_i)$ the **prior probability of** A and $\mathbb{P}(A_i|B)$ the **posterior probability of** A.*

3 Theory behind Naive Bayes Classifiers

Let's represent a tweet as the set of distinct words $(x_1, \ldots, x_n)$ contained in the text. We are interested in computing

$$\mathbb{P}(H|x_1, \ldots, x_n)$$

where H is the event that the tweet is written by Hillary, and T is the event that the tweet is written by Trump. Note that the probability of T is equivalent to H^c, or the probability that the tweet is not written by Hillary (since we're only deciding between two classes, a task called binary classification). By Bayes' theorem, this is equivalent to

$$\begin{aligned} \mathbb{P}(H|x_1, \ldots, x_n) &= \frac{\mathbb{P}(H) \times \mathbb{P}(x_1, \ldots, x_n|H)}{\mathbb{P}(x_1, \ldots, x_n)} \\ &= \frac{\mathbb{P}(H)\mathbb{P}(x_1, \ldots, x_n|H)}{\mathbb{P}(x_1, \ldots, x_n)} \\ &= \frac{\mathbb{P}(H)\mathbb{P}(x_1, \ldots, x_n|H)}{\mathbb{P}(x_1, \ldots, x_n|H)\mathbb{P}(H) + \mathbb{P}(x_1, \ldots, x_n|T)\mathbb{P}(T)} \end{aligned} \tag{2}$$

Ignoring the denominator for a minute, by definition of conditional probability,

$$\mathbb{P}(H)\mathbb{P}(x_1, \ldots, x_n|H) = \mathbb{P}(x_1, \ldots, x_n, H)$$

To simplify this equation, we need the chain rule of probability. As a brief recap, the chain rule is a rearrangement of the generalization of conditional probability formula to n terms, and not just 2. We can get the "product rule" by rearranging the formula for conditional probability:

$$\mathbb{P}(A, B) = \mathbb{P}(A|B)\mathbb{P}(B)$$

Extending this for three variables, we can see that:

$$\mathbb{P}(A, B, C) = \mathbb{P}(A|B, C)\mathbb{P}(B, C) = \mathbb{P}(A|B, C)\mathbb{P}(B|C)\mathbb{P}(C)$$

Applying this to n variables leads to the chain rule.

Theorem 3.1 (The Chain Rule of Probability)**.**

$$\mathbb{P}(A_1, A_2, \ldots, A_n) = \mathbb{P}(A_1|A_2, \ldots, A_n)\mathbb{P}(A_2|A_3, \ldots, A_n) \ldots \mathbb{P}(A_{n-1}|A_n)\mathbb{P}(A_n)$$

Applying the chain rule to decompose the probability equation we had above,

$$\begin{aligned}
\mathbb{P}(x_1, \ldots, x_n, H) &= \mathbb{P}(x_1|x_2, \ldots, x_n, H)\mathbb{P}(x_2, \ldots, x_n, H) \\
&= \mathbb{P}(x_1|x_2, \ldots, x_n, H)\mathbb{P}(x_2|x_3, \ldots, x_n, H)\mathbb{P}(x_3, \ldots, x_n, H) \\
&= \ldots \\
&= \mathbb{P}(x_1|x_2, \ldots, x_n, H)\mathbb{P}(x_2|x_3, \ldots, x_n, H) \ldots \mathbb{P}(x_{n-1}|x_n, H)\mathbb{P}(x_n|H)\mathbb{P}(H)
\end{aligned} \tag{3}$$

This is a correct formula, but it's not a particularly useful one for us! The problem is that terms like $\mathbb{P}(x_1|x_2, \ldots, x_n, H)$ have so many conditions, that it's generally not possible to accurately calculate their probability.

Think about what that term is really trying to calculate, though: what's the probability that the word x_1 occurs in a Hillary tweet, given that $(x_2, \ldots x_n)$ also occur? Unless there is another Hillary tweet with all of those words $(x_2, \ldots x_n)$ in it that you've already looked at (which is unlikely unless you have absurd amounts of data), there's no way to know that probability already. We need to simplify the problem!

Naive Bayes "fixes" this by making the assumption that the words in an tweet (and more generally, features in data) are **conditionally independent of each other, given that we know whether or not the tweet was written by Hillary**. In other words, it assumes that knowing that an tweet has the word "climate" in it, doesn't tell us anything about whether it has the word "change" in it, **if we already factor in whether the tweet was written by Hillary or not**. This is why it's called Naive Bayes: it naively assumes independence where it might not actually (and likely doesn't!) exist.

This assumption, of course, is **not true** in reality: words often appear together, and English is not just a random walk through the dictionary. But it's a useful simplification of the problem that can lead to practical results.

Let's rewrite those chain rule probabilities using the conditional independence assumptions.

$$\begin{aligned}\mathbb{P}(x_1, \ldots, x_n, H) &= \mathbb{P}(x_1|H)\mathbb{P}(x_2|x_3, \ldots, x_n, H) \ldots \mathbb{P}(x_{n-1}|x_n, H)\mathbb{P}(x_n|H)\mathbb{P}(H) \\ &\approx \mathbb{P}(x_1|H)\mathbb{P}(x_2|H) \ldots \mathbb{P}(x_{n-1}|H)\mathbb{P}(x_n|H)\mathbb{P}(H) \\ &= \mathbb{P}(H) \prod_{i=1}^{n} \mathbb{P}(x_i|H)\end{aligned} \tag{4}$$

By a similar argument,

$$\mathbb{P}(x_1, \ldots, x_n, T) \approx \mathbb{P}(T) \prod_{i=1}^{n} \mathbb{P}(x_i|T)$$

Putting it all together,

$$\mathbb{P}(H|x_1, \ldots, x_n) \approx \frac{\mathbb{P}(H) \prod_{i=1}^{n} \mathbb{P}(x_i|H)}{\mathbb{P}(H) \prod_{i=1}^{n} \mathbb{P}(x_i|H) + \mathbb{P}(T) \prod_{i=1}^{n} \mathbb{P}(x_i|T)}$$

Thus, we can calculate the probability of a tweet being authored by Hillary if we know the underlying probability of tweet being a Hillary-tweet, and the probabilities of seeing a particular word in either a Hillary or Trump tweet!

4 How Trump-ish is a word?

We have a way to compute the probability of an tweet being from Hillary Clinton or Donald Trump using relatively simple terms, but it's not yet clear how to compute those conditional probabilities. This classifier works by taking a large number of tweets that have already been hand-labelled as *Hillary* or *Trump* and uses that data to compute word-Trump (or word-Hillary) probabilities, by counting the frequency of each word.

Imagine we're given a labelled training set of 2000 Hillary tweets and 3000 Trump tweets (so $\mathbb{P}(H) = 0.4$, and $\mathbb{P}(T) = 0.6$). We'd like to calculate $\mathbb{P}(\text{"}woman\text{"}|H)$ and $\mathbb{P}(\text{"}woman\text{"}|T)$. The easiest thing to do would be to count how many Hillary tweets have "woman" and divide that by the total number of Hillary tweets (and do the same thing for Trump). So, if there were 216 Hillary tweets with "woman" and 12 Trump tweets with "woman", we'd say

$$\mathbb{P}(\text{"}woman\text{"}|H) = \frac{216}{2000} \approx 11\%$$

$$\mathbb{P}(\text{"}woman\text{"}|T) = \frac{12}{3000} \approx 0.4\%$$

5 Important Practical / Engineering Considerations

While the theory is all dandy and fine, there are some problems that we may run into when actually implementing the classifier. We detail a few of them below, and how they are solved.

5.1 Smoothing

Our theoretical framework and formulation of the solution has the right idea, but there's a small problem that we have yet to address: what if there's a word (say, "Pokemon") that we've only ever seen before in Trump tweets, and not Hillary tweets? In that case, $\mathbb{P}(\text{"}Pokemon\text{"}|H) = 0$, and the entire Hillary probability will go to zero, because we're multiplying all of the word probabilities together and we've never seen "Pokemon" in a Hillary tweet before. Thus, any tweet that we get that has "Pokemon" in it would be classified as a Trump tweet for sure! We would like to be robust to words we haven't seen before, or at least words we've only seen in one setting.

The solution is to never let any word probabilities be zero, by smoothing them upwards. **Instead of starting each word count at 0, start it at 1**. This way none of the counts will ever have a numerator of 0. This overestimates the word probability, so we need to **add 2 to the denominator**. (We add 2 because we're implicitly keeping track of 2 things: the number of tweets that contain that word, and the number that don't. The sum of those two things should be in the denominator, and the 2 accounts for starting both the counters at 1.) Essentially, you're hallucinating that you've seen the novel word at least once in your train set.

The smoothed word probabilities for the previous example are now

$$\mathbb{P}(\text{"}woman\text{"}|S) = \frac{217}{2002} \approx 11\%$$

$$\mathbb{P}(\text{"}woman\text{"}|H) = \frac{13}{3002} \approx 0.43\%$$

And our estimate for "Pokemon" in Hillary tweets would now be $\frac{1}{2002}$, instead of 0.

This technique is called **Laplacian smoothing** and was used in the 18th century to estimate the probability that the sun will rise tomorrow!

5.2 The Naive Bayes Classification algorithm

1. Iterate over the labelled Hillary tweets, and for each word w seen, count how many of the Hillary tweets contain w. Compute $\mathbb{P}(w|H) = \frac{|\text{\# Hillary tweets containing w}|+1}{|\text{\# Hillary Tweets}|+2}$.
2. Compute $\mathbb{P}(w|T)$ the same way for Trump tweets.
3. Compute $\mathbb{P}(H) = \frac{|\text{\# Hillary tweets}|}{|\text{\# Hillary tweets}|+|\text{\# Trump tweets}|}$

4. $\mathbb{P}(T) = \frac{|\#\text{ Trump tweets}|}{|\#\text{ Hillary tweets}|+|\#\text{ Trump tweets}|}$

5. Iterate over the unlabelled test tweets:

   (a) Create a set $(x_1, \ldots, x_n)$ of the distinct words in the tweet. Ignore the words that you haven't seen in the labelled training data.

   (b) Compute

$$\mathbb{P}(H|x_1, \ldots, x_n) \approx \frac{\mathbb{P}(H)\prod_{i=1}^{n}\mathbb{P}(x_i|H)}{\mathbb{P}(H)\prod_{i=1}^{n}\mathbb{P}(x_i|H) + \mathbb{P}(T)\prod_{i=1}^{n}\mathbb{P}(x_i|T)}$$

   (c) If $\mathbb{P}(H|x_1, \ldots, x_n) > 0.5$, output "Hillary", else output "Trump"

5.3 Avoiding floating point underflow

Multiplying a bunch of small probabilities together will probably result in floating point underflow, where the numbers will become too small to represent and will go to 0. Instead of calculating

$$\mathbb{P}(H)\prod_{i=1}^{n}\mathbb{P}(x_i|H)$$

(which may underflow), consider computing the logarithm of this,

$$\log\left(\mathbb{P}(H)\prod_{i=1}^{n}\mathbb{P}(x_i|H)\right)$$

which can be written equivalently as

$$\log(\mathbb{P}(H)) + \sum_{i=1}^{n}\log(\mathbb{P}(x_i|H))$$

Which will certainly not underflow. Then, realize that if

$$\log(\mathbb{P}(H)) + \sum_{i=1}^{n}\log(\mathbb{P}(x_i|H)) > \log(\mathbb{P}(T)) + \sum_{i=1}^{n}\log(\mathbb{P}(x_i|T))$$

then, because in general $\log(x) > \log(y)$ means $x > y$,

$$\mathbb{P}(H)\prod_{i=1}^{n}\mathbb{P}(x_i|H) > \mathbb{P}(T)\prod_{i=1}^{n}\mathbb{P}(x_i|T)$$

and therefore

$$\mathbb{P}(H|x_1, \ldots, x_n) > 0.5$$

and thus the tweet should be classified as "Hillary" (or as "Trump", if the sum of the logarithms for "Trump" was greater).