Naïve Bayes Classifier

In this lecture. . .

- A recap of Bayes Theorem
- Applying Bayes Theorem to Natural Language Processing
- Example: Analysing political speeches

Introduction

Naïve Bayes classifier is a supervised-learning technique. We have samples of data representative of different classes. Then using Bayes Theorem we calculate the probability of new data being in each class.

What Is It Used For?

Naïve Bayes classifier is used for

- Classifying data, often text, and analysing sentiment
- Example: Natural Language Processing (NLP), getting the computer to understand humans
- Example: Classify whether a piece of news is good or bad
- Example: Predict in which direction tweets on twitter are going to influence an election or referendum
- Example: Determine if said tweets are from a Russian bot

Naïve Bayes Classifier is often explained using examples of book reviews. By analysing the statistics of a review and comparing with a catalogue of other reviews the algorithm learns which words appear in good reviews and which in bad reviews.

- First we need a large collection of good reviews, and a large collection of bad reviews
- We count up incidences of certain words appearing in each of these categories. And thus the probability of each word appearing in each class
- Then if we are given a new review a straightforward use of Bayes Theorem will give us the probability that the new review is either good or bad

The reason that the method is called naïve is that the assumptions are usually quite unrealistic.

The main assumption being that words are independent of each, which is clearly not the case.

However one always reads that this doesn't seem to matter all that much in practice. I'll mention some possible improvements later.

Using Bayes Theorem

In our NBC example shortly we shall be applying Bayes Theorem to political speeches. An example of Bayes Theorem would be

$$P(\text{Politician is left wing}|\text{Uses the word ``Comrade''}) =$$

$$\frac{P(\text{Uses the word ``Comrade''}|\text{Politician is left wing})P(\text{Politician is left wing})}{P(\text{Uses the word ``Comrade''})}.$$

So if we hear a politician using the word "Comrade" then we can calculate the probability of him being left wing by studying the speeches of left-wing politicians, and knowing how often politicians generally use the word, and the probability of any random politician being left wing.

Application Of NBC

But we are going to apply NBC not just to a single word but to whole phrases and ultimately entire speeches. And also we don't calculate the exact probability of a politician being left wing. Instead we *compare* probabilities for being left wing and right wing. You will see when I go through the details in a moment.

Suppose we hear a politician saying "Property is theft, Comrade" and want to know whether he is left or right wing. We'd like to know

$$P(\text{Left}|\text{Says, "Property is theft, Comrade"}). \qquad (1)$$

And similar for being right wing.

We use Bayes Theorem as follows. We first write (1) as

$$P(\text{Left}|\text{Says, ``Property is theft, Comrade''}) = \frac{P(\text{Says, ``Property is theft, Comrade''}|\text{Left})P(\text{Left})}{P(\text{Says, ``Property is theft, Comrade''})}.$$

And we'd write something similar for the probability of being right wing.

We compare for Left and Right and the larger determines which way our politician swings.

But note that we don't need to calculate the denominator in the above. The denominator does not depends on the classes, whether Left or Right.

And so all we need to do is calculate the numerator for Left and Right and then compare, to see which is the larger.

We just have to estimate

$$P(\text{“Property is theft, Comrade”}|\text{Left})P(\text{Left})$$

and

$$P(\text{“Property is theft, Comrade”}|\text{Right})P(\text{Right}).$$

Now comes the naïve part. We shall assume that

$$P(\text{Says, "Property is theft, Comrade"} \,|\text{Left}) =$$

$$P(\text{Says, "Property"} \,|\text{Left}) \times P(\text{Says, "is"} \,|\text{Left}) \times$$

$$P(\text{Says, "theft"} \,|\text{Left}) \times P(\text{Says, "Comrade"} \,|\text{Left}).$$

That is, all the words are independent. (Which is clearly not true for most sentences.)

From training data we would know how often politicians of different persuasions use the words "property," "theft," etc. (Not so much the "is." That's a stop word that we would drop.) Thus each of those probabilities, for both class of politician, is easily found from many speeches of many politicians.

Let's write all this in symbols.

In Symbols

The text, or whatever, we want to classify is going to be written as $\mathbf{x}$, a vector consisting of entries x_m for $1 \leq m \leq M$, so that there are M words in the text. We want to find

$$P(C_k|\mathbf{x}) \tag{2}$$

for each of the K classes (political persuasions) C_k.

And whichever probability is largest gives us our decision (that's just maximum likelihood).

Bayes tells us that (2) can be written as

$$\frac{P(C_k)\ P(\mathbf{x}|C_k)}{P(\mathbf{x})}.$$

The denominator we can ignore because it is common to all classes (and we are only *comparing* the numbers for each class, the exact number doesn't matter).

If the features are all independent of each other then this simplifies — if they're not then this is much harder — and the numerator becomes

$$P(C_k) \prod_{m=1}^{M} P(x_m|C_k).$$

This is what we must compare for each class. The term $P(x_m|C_k)$ we will get from the data, just look at the speeches of other politicians and look at the probabilities of words, the x_ms, appearing and which political direction they lean.

Finally because we are here multiplying potentially many small numbers we usually take the logarithm of this expression. This doesn't change which class gives the maximum likelihood just makes the numbers more manageable:

$$\ln(P(C_k)) + \sum_{m=1}^{M} \ln(P(x_m|C_k)).$$

Example: Political speeches

I am going to work with the speeches or writings of some famous politicians or people with political connections. And then I will use this training set to classify another speech.

I have taken eight speeches/writings. For example one is:

A spectre is haunting Europe-the spectre of Communism. All the Powers of old Europe have entered into a holy alliance to exorcise this spectre: Pope and Czar, Metternich and Guizot, French Radicals and German police-spies. . .

This is immediately recognisable as the start of the *Communist Manifesto* by Marx and Engels.

I have also used the opening thousand or so words from each of

- Churchill: Beaches
- JFK: Inaugural address
- Benn: Maiden speech as MP
- Thatcher: The Lady's not for turning
- May: Syria strikes
- Corbyn: Post-Brexit-Referendum speech
- Trump: State of the Union

And I have classified them as either Left or Right.

Here's a selection of words used by the left and their probabilities and log probabilities:

Word	**Prob.**	**Log Prob.**
. . .		
ablaze	0.013%	-8.983
able	0.050%	-7.597
abolish	0.025%	-8.290
abolished	0.013%	-8.983
abolishing	0.019%	-8.578
abolition	0.088%	-7.037
about	0.094%	-6.968
above	0.019%	-8.578
abroad	0.019%	-8.578
absence	0.013%	-8.983
absolute	0.031%	-8.067
absolutely	0.013%	-8.983
. . .		

And by the right:

Word	**Prob.**	**Log Prob.**
. . .		
add	0.029%	-8.130
addiction	0.015%	-8.824
additional	0.015%	-8.824
address	0.015%	-8.824
administration	0.081%	-7.119
admits	0.015%	-8.824
adopt	0.015%	-8.824
advance	0.015%	-8.824
advantage	0.022%	-8.418
advantages	0.015%	-8.824
adverse	0.015%	-8.824
advert	0.015%	-8.824
. . .		

There are a couple of minor tweaks we have to do before trying to classify a new speech:

1. Remove all 'Stop' words from the texts. Stop words are words like 'the,' 'a,' 'of,' etc., words that don't add any information but could mess up the statistics.

2. There will be words in our new to-be-classified speech that do not appear in any of our training data. In order for them not to give a log probability of minus infinity we give them a default log probability of zero.

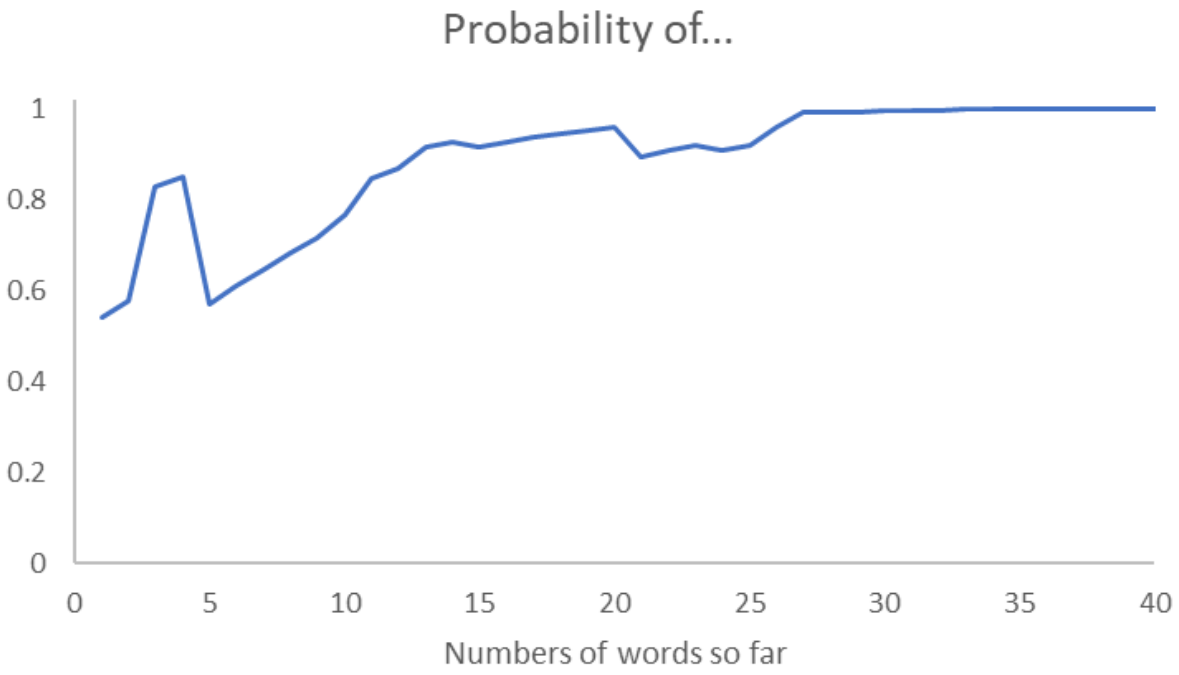
And so I took the following speech,

happy join today down history greatest demonstration freedom history nation years great American symbolic shadow stand today signed. . .

You will have noticed that I have removed a lot of stop words. So it's not as immediately recognisable as it might be!

I then for a bit of fun I took the speech and imagined doing a real-time analysis of it, in the sort of way you might listen to the report by a CEO and try to read into his presentation whether the news is good or bad before putting in your buy or sell order.

By 15 important words the speech has already set its tone. (40 words of actual speech.)



This is supposed to be the probability that the speaker is right wing. Can you guess who it is?

Well, the person is not exactly famous for being right wing. And he was not exactly a politician. But he is famous for giving good speech. It is, of course, the "I Have A Dream" speech by Martin Luther King Jr. What does this tell us? Maybe MLK was right wing. Maybe this analysis was nonsense, too little data, too little follow up. Or maybe the classifier has found something I wasn't looking for, it's lumped in MLK with Churchill and Thatcher, and not with Jeremy Corbyn, because of his powers of rhetoric.

Summary

Please take away the following important ideas

- It is easy to apply Bayes Theorem to text
- Some preprocessing of the text is usually necessary
- And you'll need lots of data