

Introduction to Data Science

Naïve Bayes Classifier

Gordon Anderson

A little background and notation, then we'll look at the naïve Bayes classifier.

Recall: Independent/Dependent Events

- Independent events: the occurrence of event A has no bearing on the occurrence of event B.
 - Example: The outcome of a dice roll does not influence the outcome of the next roll.
- Dependent event: If the occurrence of event A affects the probability of the occurrence of event B.
 - Example: Owning a sports car increases the likelihood of getting a speeding ticket.
- Question: is the probability of a word occurring in a document independent of the probability of a different word occurring?

Independent/Dependent Events

- Probability of two independent events happening at the same time:
- $P(A \text{ and } B) = P(A) \times P(B)$
- Probability of two dependent events happening: Well, they can't happen at the same time- so, say A happens first:
- $P(A \text{ and } B) = P(A) \times P(B \text{ after } A \text{ happened})$

Joint and Conditional Probability

Joint probability is the probability of two or more events happening together. It can be written several ways:

$$P(A \text{ and } B) = P(A \cap B) = P(A, B) = P(AB)$$

Conditional probability:

$$P(A|B) = P(AB)/P(B)$$

$$P(AB) = P(A|B)P(B)$$

Conditional independence

1. If A and B are independent: $P(A|B) = P(A)$
2. And, if A is independent of B given C: $P(A|B, C) = P(A|C)$
3. Combining the previous slide with this one, if we have a joint prob:

$$P(A, B, C, D)$$

Then:

$$P(A, B, C, D) = P(A|B, C, D)P(B|C, D)P(C|D)P(D)$$

Conditional independence

From the previous slide:

$$P(A, B, C, D) = P(A|B, C, D)P(B|C, D)P(C|D)P(D)$$

If we assume A, B, C are independent (a “naïve” assumption), then:

$$P(A, B, C, D) = P(A|D)P(B|D)P(C|D)P(D)$$

Now we'll look at how we use Baye's rule to build the classifier for spam emails.

Bayes Rule

Our “theories” are that an email is spam or not:

$$P(spam) = 1 - P(\neg spam)$$

The evidence is the occurrence of a keyword, such as “viagra”. This can be computed from frequency counts.

$$P(spam|word = "viagra") = \frac{P(word = "viagra"|spam)P(spam)}{P(word = "viagra")}$$

The denominator (left off the “viagra” part for brevity):

$$P(word) = P(word|spam)P(spam) + P(word|\neg spam)P(\neg spam)$$

Data Representation

Let's say we have emails represented by binary vectors, such that a 1 indicates the presence of a word in the email and a 0 indicates its absence:

words:	meeting	viagra	office	money	memo	click	...
Email1	1	0	1	0	1	0	...
Email2	0	1	0	1	0	1	...
Email3	1	0	1	0	0	0	...
etc...							

Notation

We refer to each of these emails as a vector consisting of n variables w_j , with the variable a 1 if the j^{th} word exists in the email, and 0 otherwise. The i^{th} email is: $\vec{W}_i = w_1, w_2, w_3, \dots, w_j, \dots, w_n$

The “theory”, or label, is that an email is spam or it is not spam:

$$label = \{spam, \neg spam\}.$$

So, a training set would consist of pairs for each row in the training set, this would be the k^{th} pair: $\langle label_k, \vec{W}_k \rangle$

Bayes Rule

The Bayes rule for all words in the i^{th} vector for spam and not spam:

$$P(\text{spam}|\vec{W}_i) = \frac{P(\vec{W}_i|\text{spam})P(\text{spam})}{P(\vec{W}_i)}$$

$$P(\neg\text{spam}|\vec{W}_i) = \frac{P(\vec{W}_i|\neg\text{spam})P(\neg\text{spam})}{P(\vec{W}_i)}$$

We want to know which left-hand side is greater to classify the email.

Bayes Classifier

$$P(spam|\vec{W}_i) = P(\vec{W}_i|spam)P(spam)$$

$$P(\neg spam|\vec{W}_i) = P(\vec{W}_i|\neg spam)P(\neg spam)$$

The denominator, $P(\vec{W}_i)$, is the same for both spam and not spam, so we leave it off. We calculate the probability of the words in an email given the theory its spam or not.

Bayes Classifier

Calculate the probability a single word occurs in a spam email:

$$\hat{\theta}_{jspam} = P(w_j|spam)$$

We use the greek letter theta to represent a parameter the model learns to estimate. Theta is the probability that a word is in a spam email. The fact that it is estimated is why it wears the “hat”, so we call it “theta hat”. The subscript means it is the parameter for the jth word for the theory “spam”. There would be another “theta hat” for word j for not-spam.

Bayes Classifier

The calculation that includes all words in an email, indexed by j , can be summarized by the following:

$$P(\vec{W}_i | spam) = \prod_j \theta_{js}^{x_j} (1 - \theta_{js})^{(1-x_j)}$$

This is a product, since we are treating all words as independent occurrences (naively). Each word either occurs or not, so the exponents select the correct probability estimate (theta) for that word. This is a product of n Bernoulli trials (given the word vectors are length n).

Bayes Classifier

The easy way to calculate the product:

$$\prod_j \theta_{js}^{x_j} (1 - \theta_{js})^{(1-x_j)}$$

Is to take the log of each side. Recall that $\log(xy) = \log(x) + \log(y)$.

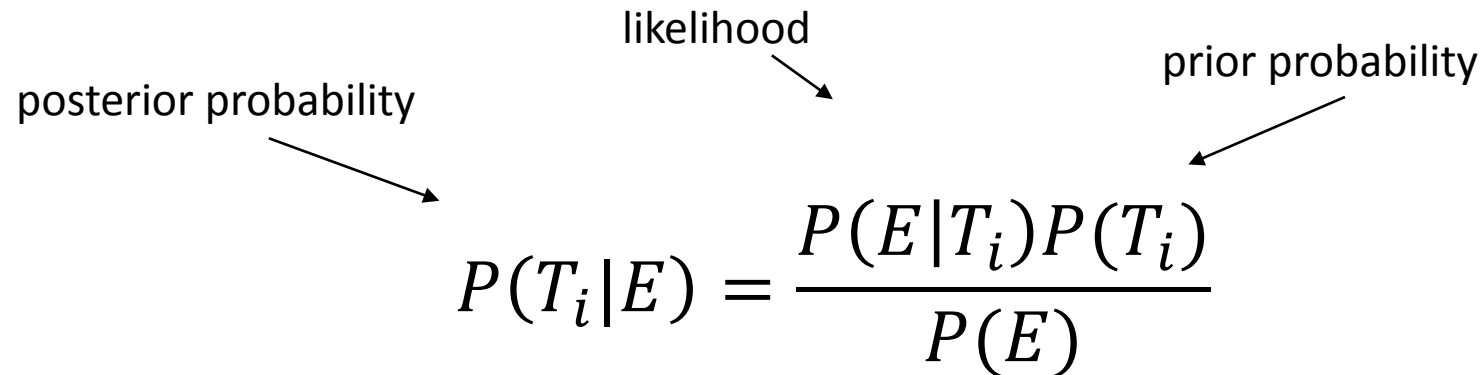
It's easier to calculate a sum than a product.

Also, the book mentions Laplace smoothing. This is a cool way to tweak the model and avoid the problem that a zero probability would bring.

The idea of hyper parameters is beyond our scope, but look further if you are interested.

Bayes Classifier in general

Suppose we have n theories \vec{T} and evidence E :



The diagram shows the Bayes' theorem formula with three labels and arrows pointing to its components: 'posterior probability' points to $P(T_i|E)$, 'likelihood' points to $P(E|T_i)$, and 'prior probability' points to $P(T_i)$.

$$P(T_i|E) = \frac{P(E|T_i)P(T_i)}{P(E)}$$

The above is a Bayesian *probability model*.

Bayes Classifier

Bayes rule for “theory” i:

$$P(T_i|E) = \frac{P(E|T_i)P(T_i)}{P(E)}$$

Calculate the likelihood term:

$$P(E|T_i) = \prod_j^n P(e_j|T_i)$$

There are n terms in the “evidence”, indexed by j.

Bayes Classifier

Add the prior term to the likelihood term:

$$P(E|T_i)P(T_i) = P(T_i) \prod_j^n P(e_j|T_i)$$

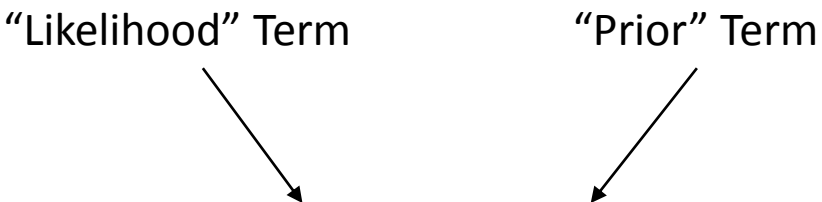
This is the numerator of the Bayes rule. It is the Bayes classifier for the i^{th} theory.

If we add a selection process, to assign the theory with the highest posterior probability, \hat{T} , to the evidence, then we have a Bayesian *classifier*:

$$\hat{T} = \text{ARGMAX}_i = P(E|T_i)P(T_i)$$

Bayes Rule

One advantage of Bayesian analysis is that we can use a “prior” belief to boost the model, thus incorporating knowledge about the domain.



“Likelihood” Term “Prior” Term

$$\hat{T} = \mathit{ARGMAX}_i = P(E|T_i)P(T_i)$$