Supervised learning

An introduction

Dr. Travis G. Coan
T.Coan@exeter.ac.uk

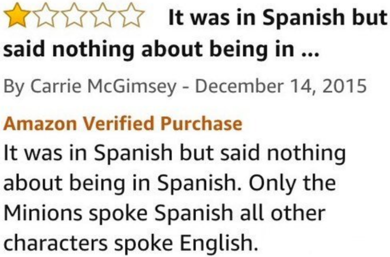
What is **supervised learning**?

Supervised learning is a set of algorithms that classify a text (e.g., a document, paragraph, or sentence) into predefined labels (or classes) based on human-annotated **training** data. These algorithms learn patterns from the associated with labeled classes in a **training set** and provide a model for predicting unseen data.

Supervised learning is often considered the “gold-standard” of text classification: if you have predefined classes and it is possible to acquire training data, then using a supervised method is the preferred options.

Getting a “feel” for supervised learning: movie reviews

Classifying movie reviews. The best way to get a feel for supervised learning is to jump right in! As an illustrative example, we will classify positive and negative movie reviews (i.e., sentiment) using the data from Pang and Lee (2014).



★☆☆☆☆ **It was in Spanish but said nothing about being in ...**

By Carrie McGimsey - December 14, 2015

Amazon Verified Purchase

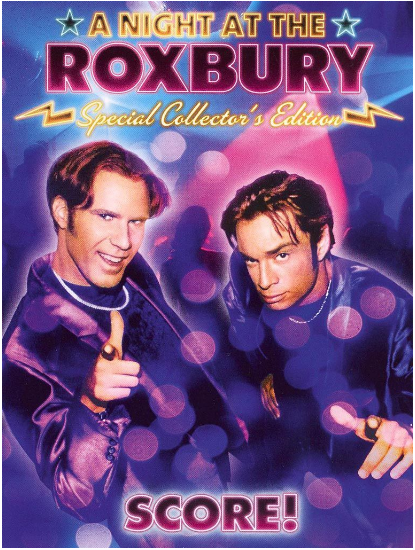
It was in Spanish but said nothing about being in Spanish. Only the Minions spoke Spanish all other characters spoke English.

A **positive** review

films adapted from comic books have had plenty of success , whether they're about superheroes (batman , superman , spawn) , or geared toward kids (casper) or the arthouse crowd (ghost world) , but there's never really been a comic book like from hell before ...

A **negative** review

two party guys bob their heads to haddawayś dance hit " what is love ? ... it's barely enough to sustain a three-minute _saturday_night_live_ skit , but _snl_ producer lorne michaels , _clueless_ creator amy heckerling , and paramount pictures saw something in the late night television institution's recurring " roxbury guys " sketch...



Getting a “feel” for supervised learning with Naive Bayes

Naive Bayes classification is a simple—but effective—model for supervised text classification. It starts with Bayes’ theorem for conditional probability:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Where $P(c|x)$ is the posterior probability of class c, $P(x|c)$ represents the likelihood, $P(c)$ is the prior class probability, and $P(x)$ is the evidence.

Getting a feel for “supervised” learning with Naive Bayes

The **naive Bayes classifier** extends Bayes’ formula to estimate the probability that a document is of a certain class, given it’s underlying words:

$$P(c|w_1, \ldots, w_n) = \frac{P(w_1, w_2, \ldots, w_n|c)P(c)}{P(w_1, \ldots, w_n)}$$

The problem with this formulation is that the likelihood is really hard to compute! To get around this, we make the usual **conditional independence assumption**.

Getting a feel for "supervised" learning with Naive Bayes

If we assume that words are (conditionally) independent—this puts the "naive" in "naive Bayes"—then we can re-write the Bayes classifier as follows:

$$P(c|w_1,\dots,w_n) = \frac{P(w_1|c)P(w_2|c)\dots P(w_n|c)P(c)}{P(w_1,\dots,w_n)}$$

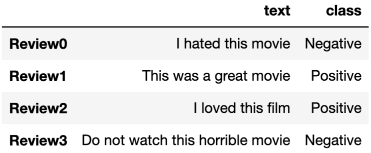
And finally,

$$P(c|w_1,\dots,w_n) \propto P(c)\prod_{i=1}^{n} P(w_i|c)$$

We can thus drop the denominator all together and simply recognize that the numerator is still proportional to the posterior probability of interest.

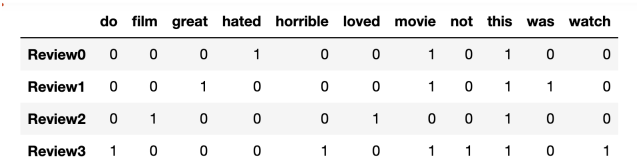
Naive Bayes "by hand"

Assume that we have the following movie reviews data:



	text	class
Review0	I hated this movie	Negative
Review1	This was a great movie	Positive
Review2	I loved this film	Positive
Review3	Do not watch this horrible movie	Negative

To estimate the probabilities needed to "train" our naive Bayes classifier, we first need to count the number of times that words appear. We typically organize these counts in **document-term matrix**:



	do	film	great	hated	horrible	loved	movie	not	this	was	watch
Review0	0	0	0	1	0	0	1	0	1	0	0
Review1	0	0	1	0	0	0	1	0	1	1	0
Review2	0	1	0	0	0	1	0	0	1	0	0
Review3	1	0	0	0	1	0	1	1	1	0	1

Naive Bayes "by hand"

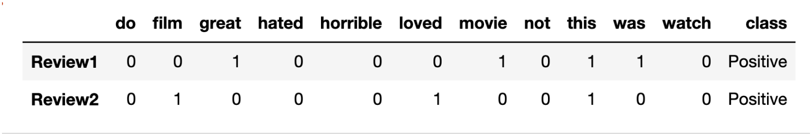
Again, our "model" for estimating the probability for each review class (i.e., positive or negative):

$$P(c|w_1, \ldots, w_n) \propto P(c) \prod_{i=1}^{n} P(w_i|c)$$

So we need estimates for the following:

1. $P(c = Positive)$: Prior probability that the class is positive.
2. $P(w_i|c = Positive)$: Probability of each word given that the class is positive.
3. $P(c = Negative)$: Prior probability that the class is negative.
4. $P(w_i|c = Negative)$: Probability of each word given that the class is negative.

Start with **positive** reviews



	do	film	great	hated	horrible	loved	movie	not	this	was	watch	class
Review1	0	0	1	0	0	0	1	0	1	1	0	Positive
Review2	0	1	0	0	0	1	0	0	1	0	0	Positive

The **prior probability** is straightforward to calculate:

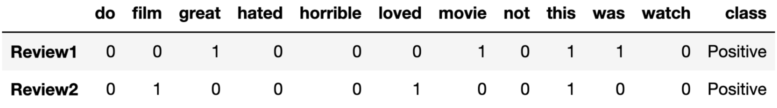
$$P(c = Positive) = \frac{n_{positive}}{n_{reviews}} = \frac{2}{4} = 0.50$$

The conditional word probabilities are also straightforward to calculate. We will use the following formula for the calculation:

$$P(w_i | c = Positive) = \frac{n_{i|+} + 1}{n_{w|+} + n_{vocab}}$$

Where $n_{i|+}$ is the number of times word i shows up in positive reviews, $n_{w|+}$ is the total number of words in positive reviews, and n_{vocab} is the number of unique words.

Start with **positive** reviews



	do	film	great	hated	horrible	loved	movie	not	this	was	watch	class
Review1	0	0	1	0	0	0	1	0	1	1	0	Positive
Review2	0	1	0	0	0	1	0	0	1	0	0	Positive

$$P(do|c = Positive) = \frac{0+1}{7+11} = .056 \qquad P(movie|c = Positive) = \frac{1+1}{7+11} = .11$$

$$P(film|c = Positive) = \frac{1+1}{7+11} = .11 \qquad P(not|c = Positive) = \frac{0+1}{7+11} = .056$$

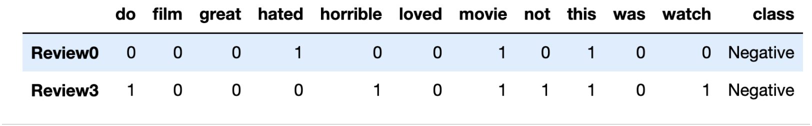
$$P(great|c = Positive) = \frac{1+1}{7+11} = .11 \qquad P(this|c = Positive) = \frac{2+1}{7+11} = .167$$

$$P(hated|c = Positive) = \frac{0+1}{7+11} = .056 \qquad P(was|c = Positive) = \frac{1+1}{7+11} = .11$$

$$P(horrible|c = Positive) = \frac{0+1}{7+11} = .056 \qquad P(watch|c = Positive) = \frac{0+1}{7+11} = .056$$

$$P(loved|c = Positive) = \frac{1+1}{7+11} = .11$$

Next **negative** reviews



	do	film	great	hated	horrible	loved	movie	not	this	was	watch	class
Review0	0	0	0	1	0	0	1	0	1	0	0	Negative
Review3	1	0	0	0	1	0	1	1	1	0	1	Negative

$$P(do|c = Negative) = \frac{1+1}{9+11} = .1 \qquad P(movie|c = Negative) = \frac{2+1}{9+11} = .15$$

$$P(film|c = Negative) = \frac{0+1}{9+11} = .05 \qquad P(not|c = Negative) = \frac{1+1}{9+11} = .1$$

$$P(great|c = Negative) = \frac{0+1}{9+11} = .05 \qquad P(this|c = Negative) = \frac{2+1}{9+11} = .1$$

$$P(hated|c = Negative) = \frac{1+1}{9+11} = .1 \qquad P(was|c = Negative) = \frac{0+1}{9+11} = .05$$

$$P(horrible|c = Negative) = \frac{1+1}{9+11} = .1 \qquad P(watch|c = Negative) = \frac{1+1}{9+11} = .1$$

$$P(loved|c = Negative) = \frac{0+1}{9+11} = .05$$

Classifying a movie review

Let's say we have the following new review:

The acting is horrible

We can now use our model and the "weights" estimated on the previous slides to estimate the probability the review is positive or negative:

$$\begin{aligned}
P(Positive|horrible) &= P(Positive) * P(horrible|c = Positive) \\
&= 0.50 * 0.056 \\
&= 0.028 \\
P(Negative|horrible) &= P(Negative) * P(horrible|c = Negative) \\
&= 0.50 * 0.10 \\
&= 0.05
\end{aligned}$$

To classify the review, we simply choose the class with the highest probability.

Naive Bayes in **Python**

While estimating our naive Bayes model by hand is possible, it is tedious for even this small problem and infeasible for any real-world problem. Python and sklearn to the rescue!

Let's head over to the supervised learning notebook now to see Python work it's magic.