Text Mining and Sentiment Analysis

Prof. Annamaria Bianchi
A.Y. 2024/2025

Lecture 14

8 April 2025

Outline

SA with Machine Learning: Naive Bayes

Statistical Learning

It is a vast set of tools for understanding the data. The tools can be classified as supervised or unsupervised:

- **Supervised learning**: build a statistical model for predicting or estimating an output based on one or more inputs. In the case of classification (ex. Sentiment Analysis), a dataset with labels is used to *train* and *test* the model before implementing the analysis on unlabeled data.

- **Unsupervised learning**: a labelled dataset is not available. The objective is to learn relationship and structure from the data. An example is Topic Modeling.

Here we focus on supervised learning for Sentiment Analysis.

Supervised classification

Sentiment Analysis is viewed as a **classification task** and this can be done via supervised machine learning.

Formally, the task of **supervised classification** is to take an input document *d* and return a predicted class *c* among a set of possible classes *C.*

In the supervised situation we have a training set of *N* documents that have each been hand-labeled with a class:

$$(d_1, c_1),\ldots\ldots, (d_N, c_N)$$

Goal: learn a classifier that is capable of mapping from a new document *d* to its correct class *c*ϵ*C*. A **probabilistic classifier** will tell us the probability of the document being in the class.

The Naive Bayes Classifier

The Naïve Bayes is the most famous supervised probabilistic classifier.

It is considered a **generative classifier** as it builds a model of how a class could generate some input data. Given a document is returns the class most likely to have generated it.

The Naïve Bayes classifier ignores the position of the words (**bag-of-word** assumption)

The name comes from a simplifying (naïve) assumption about how the features (words) interact.

The Naive Bayes Classifier

Given a document *d*, denote by *P(c|d)* the probability that it belongs to class $c \in C$ (posterior probability).

The Naïve Bayes returns the class $\hat{c}$ which has the maximum posterior probability given the document.

Our estimate $\hat{c}$ of the class for document d is

$$\boxed{\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d)}$$

The Naive Bayes Classifier

Let us consider a training set of N_{doc} documents which are labeled with a class *C*= $\{positive, negative\}$. The goal is to train a classifier which is able to tell the probability of the document being in that class.

We need to re-write the probability *P(c/d)* in a way that can be estimated from the training data. This is based on Bayes' rule.

In general, given a conditional probability *P(x/y)*, Bayes' rule allows to reverse the conditioning and gives a way to break down the conditional probability into three other probabilities

$$P(x|y) = \frac{P(y|x) \cdot P(x)}{P(y)}.$$

The Naive Bayes Classifier

The posterior probability *P(c|d)* can then be written as

$$P(c|d) = \frac{P(d|c) \cdot P(c)}{P(d)} \propto P(d|c) \cdot P(c)$$

where $P(c|d)$ is the posterior and it is proportional to the product between the $P(c)$ prior and the $P(d|c)$ likelihood. It is proportional because the denominator $P(d)$ does not change for each document.

The Naive Bayes Classifier

The document can be represented as a set of features, i.e., words. The Naive Bayes classifier is based on the **bag-of-words text representation**. This means that we consider the words and their frequency in the documents but we ignore the position of the words or syntactic features. The "naive" assumption of this classifier is that the probability of observing a word is independent of each other given the class, i.e.,

$$P(d|c) = P(w_1, w_2, \dots, w_n|c) = P(w_1|c) \cdot P(w_2|c) \cdot \ldots \cdot P(w_n|c)$$

Thus, rewriting the formula we obtain:

$$P(c|d) \propto P(c) \cdot [P(w_1|c) \cdot P(w_2|c) \cdot \ldots \cdot P(w_n|c)]$$

The predicted class for document *d* is the one maximizing $P(c) \cdot [P(w_1|c) \cdot P(w_2|c) \cdot \ldots \cdot P(w_n|c)]$

The Naive Bayes Classifier

From the training set, we estimate the class prior *P(c)* as

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

where

N_c is the number of documents in the c Class

N_{doc} is the number of documents

The Naive Bayes Classifier

We estimate $P(w_i|c)$ as the fraction of times the word w_i appears among all words in all document of class *c*:

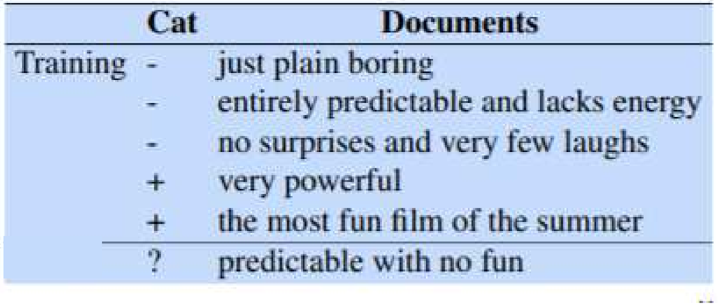
$$\hat{P}(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} count(w, c) + |V|}$$

where

V is the vocabulary (collection of unique words) and $|V|$ indicates the vocabulary size.

The +1 and +|V| avoid to have a probability equal to zero.

Worked example



	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
	?	predictable with no fun

The **prior probabilities** for the two classes are:

$$\hat{P}(-) = \frac{3}{5} \text{ and } \hat{P}(+) = \frac{2}{5}$$

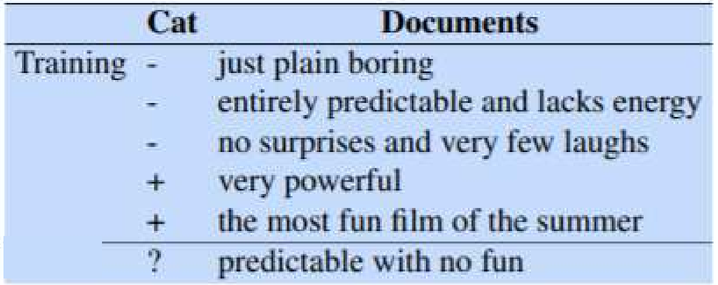
The word *with* doesn't occur in the training set, so we drop it from the data.

The **likelihoods** are computed as follows:

$$\hat{P}(\text{"predictable"}| -) = \frac{1+1}{14+20}$$

- $count(\text{predictable}, -) = 1$
- $\sum_{w \in V} count(w, -) = 14$
- $|V| = 20$

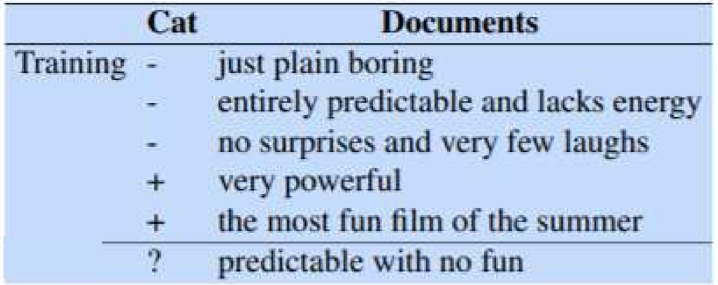
Worked example



	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
	?	predictable with no fun

Exercise. Compute the other likelihoods.

Worked example



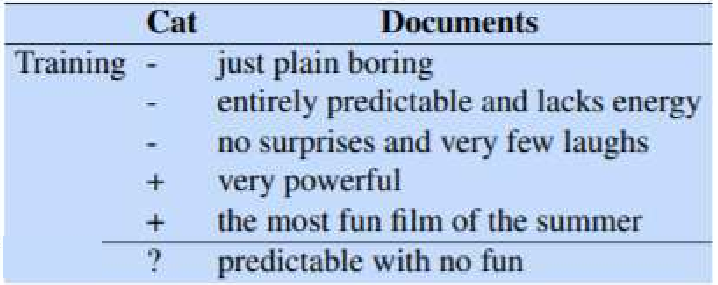
	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
	?	predictable with no fun

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

Worked example



	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
	?	predictable with no fun

For the sentence S=«predictable with no fun», after removing the word *with*, the posterior probabilities are computed as follows:

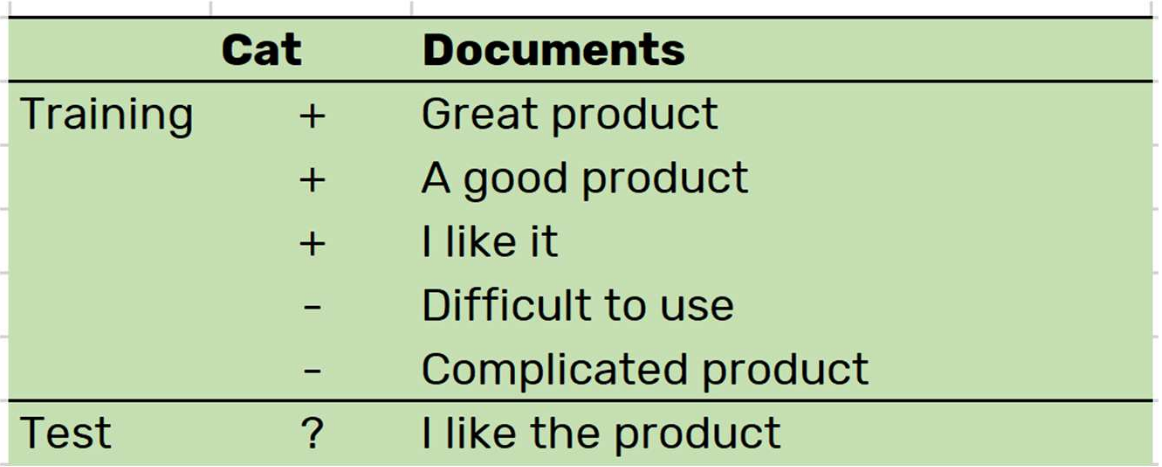
$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Thus, the model predicts that the class is **negative.**

Exercises for you

Exercise 1. Consider the following miniature training and test documents simplified from actual product reviews.

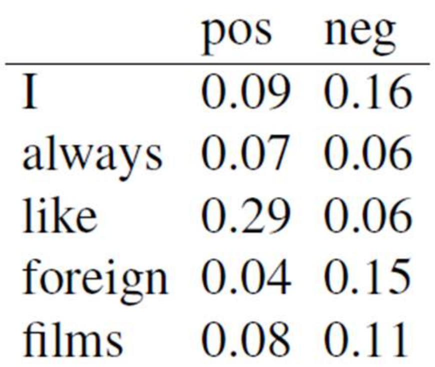


	Cat	Documents
Training	+	Great product
	+	A good product
	+	I like it
	-	Difficult to use
	-	Complicated product
Test	?	I like the product

What class will Naive Bayes assign to the test data?

Exercises for you

Exercise 2. Assume the following likelihoods for each word being part of a positive or negative movie review, and equal prior probabilities for each class



	pos	neg
I	0.09	0.16
always	0.07	0.06
like	0.29	0.06
foreign	0.04	0.15
films	0.08	0.11

What class will Naive Bayes assign to the sentence «I always like foreign films.»?