LIN7076 – Foundations of Computational Linguistics

Naive Bayes

Adèle Hénot-Mortier

07/10/2025

Queen Mary University of London

Binary classification: some applications

- Positive? Negative?
- Neutral? Hateful?
- Neutral? Sarcastic?
- Real news? Fake news?
- Unbiased? Biased?

Binary classification: the task

- Given a bunch of text, predict if it belongs to class 0 (e.g. neutral) or to class 1 (e.g. hateful).
- Supervised learning *via* a set of (text, class) training examples.
- A few techniques used for binary classification: Naive Bayes, logistic regression, support vector machines, and of course, neural nets.[1]
- This semester, we'll focus on Naive Bayes and logistic regression, and today specifically on **Naive Bayes**.
- We will focus on binary classification, but in fact Naive Bayes extends to more than 2 classes!

[1] There's also *k*-nearest neighbors, but it's unsupervised.

Naive Bayes: core intuition

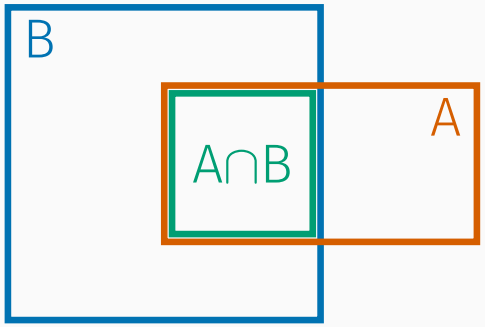
- Different pieces of text have different characteristics, for instance, they vary in length, capitalization, use of punctuation, and of course, in the words they use.
- Some of these features seems very relevant to certain classification tasks; for instance, a text in all caps with lots of strong punctuation and offensive words, is probably hateful. A text without these features is more likely to be neutral.
- To determine which class a text belongs to, one can then think of how likely it is to display the feature it has, were it from class 0, or from class 1.
- **A piece of text belongs to the class that makes its features more likely!**

Conditional probabilities, again!

- The **conditional probability** of an event A, given an event B, can be computed using the joint probability of A and B, and B's probability.
- Symmetrically, the conditional probability of B, given A, can be computed using the same joint probability, and A's probability.

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

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



Classification as conditional probability optimization

- Suppose we have a piece of text ("document") d, that we want to classify as neutral (c_0) or hateful (c_1).
- We want to figure out which class is the most likely to represent d, meaning, we want to compute $P(c_0|d)$ and $P(c_1|d)$, and assign d the class that is more likely.

$$\hat{c}(d) = \begin{cases} c_0 \text{ if } P(c_0|d) > P(c_1|d) \\ c_1 \text{ if } P(c_0|d) < P(c_1|d) \end{cases}$$

- We can "compress" and generalize this with the argmax notation: $\text{argmax}_x f(x)$ returns the element x that maximizes the function f; here, the class c that maximizes the function $P(c|d)$ for a fixed d.

$$\hat{c}(d) = \text{argmax}_{c \in \{c_0, c_1\}} P(c|d)$$

Deriving the Bayes formula for document classification

- To compare $P(c_0|d)$ and $P(c_1|d)$, we need to approximate them!
- For any class c, we know that $P(c|d)$ is $\frac{P(c \cap d)}{P(d)}$, but the joint probability $P(c \cap d)$ is equally hard to approximate...
- Here is the trick: $P(d|c)$, the probability of a document, *given a class*, is probably easier to compute (based on d's features).

$$\begin{aligned} P(d|c) &= \frac{P(c \cap d)}{P(c)} \\ P(d \cap c) &= P(d|c)P(c) \end{aligned}$$

- We can then express our target probability $P(c|d)$ as a function of $P(d|c)$, $P(d)$, and $P(c)$!

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

Analyzing the Bayes formula

$$P(c_0|d) = \frac{P(d|c_0)P(c_0)}{P(d)} \qquad P(c_1|d) = \frac{P(d|c_1)P(c_1)}{P(d)}$$

- To compare the above conditional probabilities, we can conveniently ignore the denominators $P(d)$.

$$\underbrace{P(c|d)}_{posterior} \propto \overbrace{P(d|c)}^{likelihood} \underbrace{P(c)}_{prior}$$

- The above equation says that the probability of a class, after observing a document (**posterior** $P(c|d)$), is proportional to:
  - how likely the document is, given that class (**likelihood** $P(d|c)$);
  - and to how probable the class is before observing the document (**prior** $P(c)$).

Illustrating the interplay between prior and likelihood

- Suppose harmful (c_1) documents represent .01% of all documents. $P(c_1) = .0001$, $P(c_0) = .9999$.
- Suppose we have a document d, whose features make it very likely to be harmful, say $P(d|c_1) = .99$. Still it's not completely impossible d is neutral, say $P(d|c_0) = .05$.
- Which class should we assign d?

$$P(c_0|d) \propto .05 * .999 \simeq .05 \qquad P(c_1|d) \propto .99 * .0001 \simeq .0001$$

- Counterintuitively, we should predict d to be neutral according to Naive Bayes!
- This illustrates how priors influence more "rational" decision making, sometimes against what likelihood suggests. We humans are notoriously bad at this!

Introducing features

$$\begin{aligned} P(c|d) &\propto P(d|c)P(c) \\ \hat{c}(d) &= \text{argmax}_{c \in \{c_0, c_1\}} P(c|d) \\ &= \text{argmax}_{c \in \{c_0, c_1\}} P(d|c)P(c) \end{aligned}$$

- Our predicted class for which d is the most likely, weighted by how probable the class is independently of d.
- How to measure how likely a document is given a class? We suggested that documents with certain features, tend to belong to certain classes.

  $P(d|c) = P(F(d)|c)$ with $F(d) = (f_1, ..., f_k)$ d's relevant features

- For instance, harmful documents contain harmful words... so the f_i can be words. This may sound idle, but dealing with individual words will help us further break down $P(d|c)$.

The Bag of Word assumption

- We assume that documents are “featurized” as the multiset of their words. A “multiset” is unordered like a set, but keeps track of multiple occurrences.
- It’s like writing each individual word of your document on separate pieces of paper, and tossing them all into a bag. For any document d, we call $BoW(d)$ its corresponding **Bag of Words**.

$$
\begin{aligned}
P(d|c) &\simeq P(BoW(d)|c) \\
\hat{c}(d) &= \text{argmax}_{c \in \{c_0, c_1\}} P(BoW(d)|c)P(c)
\end{aligned}
$$

- Can you think of a classification task and document for which loosing positional information would likely affect classification?

The independence assumption

$$\begin{aligned} P(d|c) &\simeq P(BoW(d)|c) \\ \hat{c}(d) &= \text{argmax}_{c \in \{c_0, c_1\}} P(BoW(d)|c)P(c) \end{aligned}$$

- To further decompose $P(BoW(d)|c)$, we take that the words in our document are produced independently from each other.
- This means that knowing a word is in the bag does not inform us about how likely other words are in the bag.
- So our likelihood becomes a convenient product of conditional probabilities, one per feature, i.e. one per word in the bag.

$$P(d|c) \simeq P(BoW(d)|c) = \prod_{w \in BoW(d)} P(w|c)$$

- Can you give arguments from syntax or semantics showing why the independence assumption is “naive”?

Summary of the model

$$\begin{aligned}\hat{c}(d) &= \text{argmax}_{c \in \{c_0, c_1\}} P(c|d) \\ &= \text{argmax}_{c \in \{c_0, c_1\}} P(c) \prod_{w \in BoW(d)} P(w|c)\end{aligned}$$

- A document belongs to the class that maximizes the likelihood of its constitutive words ($P(w|c)$), weighted by the class' prior ($P(c)$).
- How to estimate $P(c)$ and $P(w|c)$ for any c and w?

content...