Lecture Notes: Naive Bayes for Text Classification

CSCI 375: NLP / Williams College / Fall 2024

1 Bayes theorem

First, we will derive **Bayes theorem**. Recall the definition of a conditional probability

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{1}$$

Rearranging terms, we have

$$P(A,B) = P(A|B)P(B) \tag{2}$$
$$= P(B|A)P(A) \tag{3}$$

Let's substitute Equation 3 into Equation 1,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{4}$$

This is called **Bayes theorem** and the various components in the equation are given names

$$\underbrace{P(A|B)}_{\text{posterior}} = \frac{\overbrace{P(B|A)}^{\text{likelihood}}\overbrace{P(A)}^{\text{prior}}}{\underbrace{P(B)}_{\text{evidence}}} \tag{5}$$

2 Naive Bayes

Now we will use Bayes theorem to derive an approach to text classification.

Let $x_1, x_2, \ldots x_n$ be observed pieces of text (e.g., sentences) in the training data.

Let $\mathcal{Y}$ be the set of class labels for the task, e.g., {spam, not spam} for spam detection or {Tolkien, Shakespeare, Austen} for author identification, and $y \in \mathcal{Y}$ be a specific class label, e.g., $y =$ spam.

The goal of text classification is to select a class label for each document i given a probabilistic model

$$\hat{y_i} = \underset{y \in \mathcal{Y}}{\text{argmax}}\ P(y|x_i) \tag{6}$$

Here, *argmax* is the "argument maximum" meaning it returns the argument, y, that gives the maximum value when plugged into $P(y|x_i)$.

Applying Bayes theorem to Equation 6, we obtain

$$\hat{y_i} = \underset{y \in \mathcal{Y}}{\text{argmax}}\ \frac{P(x_i|y)P(y)}{P(x_i)} \tag{7}$$

Because denominator is a constant across all y values, it does not influence the expression that is the maximum. So the equation above simplifies to

$$\hat{y_i} \propto \underset{y \in \mathcal{Y}}{\text{argmax}}\ P(x_i|y)P(y) \tag{8}$$

where $\propto$ means "proportional to".

The "naive" part of "Naive Bayes" is that we make the following assumption:

- A **conditional independence assumption** that all words in a document are independent conditional on the class label

$$P(x_i|y) \approx \prod_{k \in \text{index}(i)} P(w_k|y) \tag{9}$$

Here, $k \in \text{index}(i)$ is shorthand for saying we iterate through every position, or "index" in the document i and w_k is the word-type of the token at index k.

Substituting this assumption into Equation 8, we have

$$\hat{y}_i \approx \underset{y \in \mathcal{Y}}{\text{argmax}} \left(\prod_{k \in \text{index}(i)} P(w_k|y) \right) P(y) \tag{10}$$

When finding the argmax, we can apply a monotonic function (a function which is either entirely nonincreasing or nondecreasing) to our expression and the argmax will not change. To prevent underflow in coding implementations, we use logs which are monotonic functions. Thus,

$$\hat{y}_i = \underset{y \in \mathcal{Y}}{\text{argmax}} \ \log \left(P(y) \prod_{k \in \text{index}(i)} P(w_k|y) \right) \tag{11}$$

$$= \underset{y \in \mathcal{Y}}{\text{argmax}} \left(\log P(y) + \sum_{k \in \text{index}(i)} \log P(w_k|y) \right) \tag{12}$$

Now we'll calculate the **maximum likelihood estimates** from the training data. We'll denote an estimate with a hat, e.g., $\hat{P}(\cdot)$. Thus, the estimate of the *prior* is

$$\hat{P}(y) = \frac{\sum_{i=1}^{n} \mathbb{1}(y_i = y)}{n} \tag{13}$$

where $\mathbb{1}$ is the indicator function (it might help to think of the Python equivalent which is $y_i == y$) and n is the number of documents in the training data.

For any word type w_j, we calculate the estimates across *all documents*

$$\hat{P}(w_j|y) = \frac{\text{Count}(w_j, y)}{\sum_{l=1}^{|V|} \text{Count}(w_l, y)} \tag{14}$$

In the equation above, the numerator is the number of tokens for which the word type w_j co-occurs with class label y among all documents. The denominator is the number of tokens for *any* word type across the entire vocabulary V (a vocabulary that is constructed across all class labels in Y) co-occurs with the class label y among all documents.

In practice, we use **Laplace smoothing** to ensure that we have no $\hat{P}(w_j|y) = 0$. This changes Equation 14 to

$$\hat{P}(w_j|y) = \frac{\text{Count}(w_j, y) + 1}{\left(\sum_{l=1}^{|V|} \text{Count}(w_l, y) \right) + |V|} \tag{15}$$