Naive Bayes Classifier for Text Classification

Divya Agarwal, NUID: 001226349
Agarwal.d@husky.neu.edu

Reference: - Dan Jurafsky and James H. Martin, "Speech and Language Processing, 2nd Edition", Prentice Hall, 2009.

We represent the text classifier as a bag of words. Bag of words is an unordered set of words (position in the document is ignored) and frequency of the word in the document is recorded.

Naive Bayes is a probabilistic classifier, meaning that for a document d, out of all classes c ∈ C the classifier returns the class ˆc which has the maximum posterior ˆ probability given the document. (Equation 1)

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

By Bayes Rule, Posterior probability = P(A|B) = (Equation 2)

$$\frac{P(B|A)P(A)}{P(B)}$$

We use, Bayes rule to transform the Equation 1 into other probabilities that have some useful properties. (Equation 3)

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

We can drop the denominator P(d) since it is same for every possible class c.
We thus compute the most probable class ˆc given some document d by choosing the class which has the highest product of two probabilities: the prior probability prior probability likelihood of the class P(c) and the likelihood of the document P(d|c):

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

We can represent d as a set of features f1, f2, .... fn.

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, ...., f_n|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

Naïve Bayes Classifier makes two assumptions: -

1. Bag of words assumption; the order of occurrence of a word does not matter. Thus, the features only represent the word identity and not the position.
2. Naïve Bayes Assumption; Conditional independence assumption, the probabilities P(fi|c) are independent given the class c and thus,

$$P(f_1, f_2, ...., f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot ... \cdot P(f_n|c)$$

The equation now becomes,

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{f \in F} P(f|c)$$

To apply Navie Bayes to text, we consider occurrence of words, thus, (word positions, by simply walking an index through every word position in the document)

$$\begin{aligned} \text{positions} &\leftarrow \text{all word positions in test document} \\ c_{NB} &= \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{i \in positions} P(w_i|c) \end{aligned}$$

In the numerator, we are multiplying many probabilities together. We could end up with reall small numbers in 0<P<1 and the computer might round it to zero. To avoid that we take log on both sides.

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

Thus, the document prior can be given by,

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

The maximum likelihood estimate, that is the frequency of wi given the document c is given by,

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

When a word in the training set is missing from a class (Spam or Ham), the count of the word in the class becomes zero and so is the conditional probability. Since naïve bayes multiplies all the feature likelihoods together, zero probability for any class will cause the probability of the entire class to be zero. To avoid this, we apply Laplace Smoothing.

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