Bayensian Learning

Homework 1: Naive Bayesian Classification

Emilie Krutnes Engen

Master in Big Data Analytics
Carlos III University of Madrid

Universidad
Carlos III de Madrid

Naive Bayesian Classification

In this assignment we want apply Bayes Theorem to a text classification problem. By using a naive Bayes classifier, we wish to classify a set of documents based on the content of the documents. Thus imagine that you have a set of documents where each document belong to a certain class. Each document class can be modeled as a set of words, where the independent probability of a word w_i from a given document occur in a document from class C_k is $p(w_i|C_k)$. In order to determine the class of a given document, we compute the probability that document D belongs to a given class C_k. This probability can be obtained, using Bayes theorem

$$p(C_k|D) = \frac{p(C_k \cap D)}{p(D)} = \frac{p(C_k)p(D|C_k)}{p(D)} \tag{1}$$

where $p(C_k|D)$ is referred to as the posterior probability, $p(C_k)$ the prior probability, $p(C_k \cap D)$ the likelihood and $p(D)$ the evidence. By assuming that all words in the document are randomly distributed, implying that they are independent of the length of the document, the position in the document and other words in the document, we can apply naive Bayes to construct a classifier for this problem. The naive Bayes probability model is given by following conditional distribution over the class C_k.

$$p(C_k|D) = \frac{p(C_k)}{p(D)} \prod_i p(w_i|C_k) \tag{2}$$

In order to obtain the naive Bayes classifier the probability model from Equation 2 is combined with a decision rule. The most common rule is the maximum posterior rule, which imply assigning the class with the highest probability. This classifier predicts a class label $\hat{y}$ according to the following condition.

$$\hat{y} = \max_{k \in K} \frac{p(C_k)}{p(D)} \prod_i p(w_i|C_k) \tag{3}$$

Here $p(D)$ is only a scaling factor as all the words in the documents used for constructing the classifier is known.

The Dataset

For the purpose of this problem I have chosen a collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. To reduce the computation time of this classification problem, the problem was limited to classifying documents in the Electronics class, *sci.electronics*, and the Medicine class, *sci.med*. This give us a total of 1974 documents. The two news categories and the associated number of documents are presented in Table 1.

Class	Documents
sci.electronics	984
sci.med	990
Total	1974

Table 1: Number of documents for the two classes used in the classification problem

Data Cleaning

Because the dataset was already partitioned into a training and test set, I first combined the two into a single dataset. Before constructing the classifier we first want to clean the documents. This is done by constructing a corpus, which is basically a collection of all the documents in the dataset. To enable the classifier to compare the words in the document collection, we have to ensure that the classifier is able to recognise equivalent words. This includes transforming all letters to lower case, removing punctuations, numbers, excess white space and stop words. The latter are common non-content words as *you*, *me*, *to*, *and* etc. These words are frequently used regardless of the document class, and should therefore not be used for prediction. By applying the function *stopwords()* we investigate the occurrence of stop words in the document collection from a set of 174 unique words. Finally the documents in the corpus are converted into plain text.

Explanatory Analysis

In order to investigate the differences between the two newsgroup classes, we can construct a word cloud for each class. A word cloud illustrates the most frequent terms for a given class. The word clouds for the Electronics and Medicine classes are presented in Figure 1 and 2, respectively.

For better visualization and interpretation the number of words appearing in the word clouds is restricted by a minimum frequency of 100 for each document class. From the word cloud we have that *edu*, *can*, *one*, *writes* and *use* are the most frequent words in the Electronics articles and *edu*, *can*, *one*, *com*, *writes* and *article* are the most frequent in the Medicine articles. The most frequently used words in the two classes are fairly general, as the two classes have similar high frequency terms.

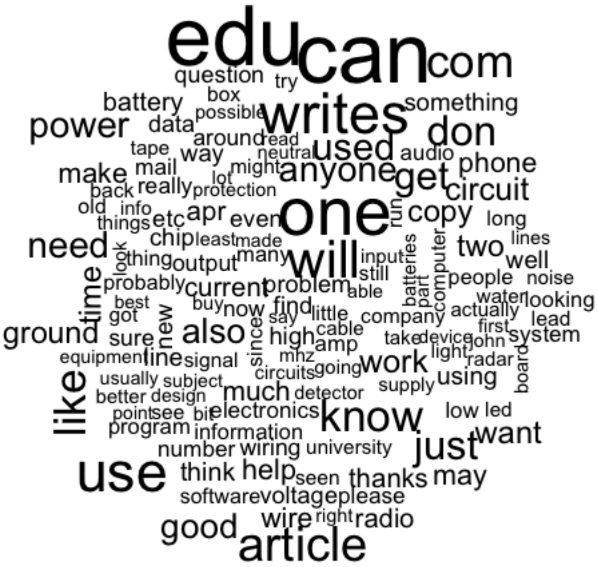

Figure 1: Word cloud of terms from the Electronics documents

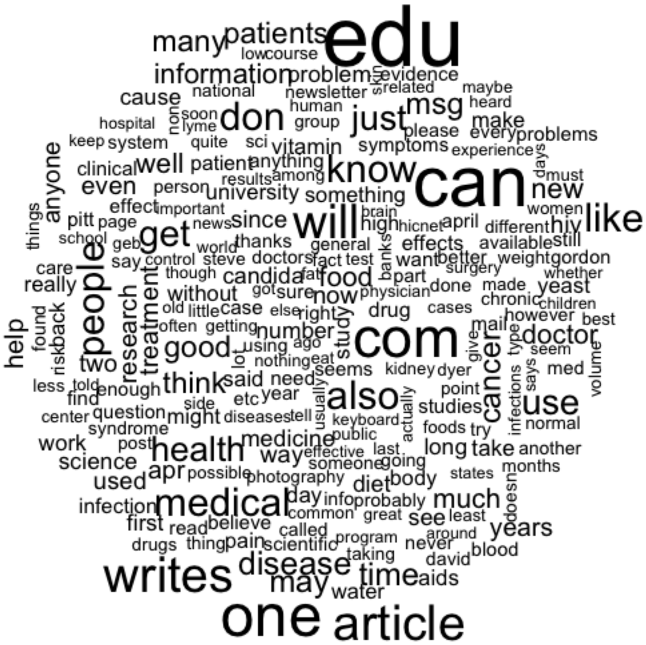

Figure 2: Word cloud of terms from the Medicine documents

In order to create accurate predictions we have to encounter more than just the most frequent words. We further have that the Electronics class has a lower occurrence of words with a frequency higher than 100 than the Medicine class, which may result in different prediction accuracy for the two classes.

Constructing Train and Test Samples

Before building the classifier, we first partition the dataset into a training and test set. The training set is used to build the classifier, while the test set is used for evaluating the performance of the constructed model. We divide the dataset using random sampling. The training set includes 75 % of documents in the total dataset and the test set the remaining 25 %. The proportions of the two newsgroup classes in the train and test sample is presented in Table 2.

	Medicine	Electronics
Original	50.15 %	49.85 %
Train	50.81 %	49.19 %
Test	48.18 %	51.82 %

Table 2: Proportions of the two classes in the original dataset, the train and the test sample

According to Table 2 the proportions in the two partitions are nearly the same as in the original dataset, although the proportion of Medicine documents in the test sample is slightly higher than in the original dataset.

Building a Classifier

The naive Bayesian classifier classify documents in the test set by calculating the probability that a certain document belong to a class given the words in the given document. Each document have a high number of words, where a large part only occur a few times. To get the best prediction we only want to include the words that are considered important in each document. Words with a frequency lower than five are therefore excluded. Finally the naive Bayes classifier is constructed using the words and classes for the documents in the training set. In this problem we only consider two document classes, Medicine M and Electronics E. Thus the naive Bayes probability model can be obtained by the following equations.

$$p(M|D) = \frac{p(M)}{p(D)} \prod_i p(w_i|M) \tag{4}$$

$$p(E|D) = \frac{p(E)}{p(D)} \prod_i p(w_i|E) \tag{5}$$

where $E = \neg M$.

By dividing the first equation by the other and then taking the natural logarithm we get:

$$\frac{p(M|D)}{p(\neg M|D)} = \frac{p(M)}{p(\neg M)} \frac{\prod_i p(w_i|M)}{\prod_i p(w_i|\neg M)} \tag{6}$$

$$ln \frac{p(M|D)}{p(\neg M|D)} = ln \frac{p(M)}{p(\neg M)} + \sum_i ln \frac{p(w_i|M)}{p(w_i|\neg M)} \tag{7}$$

Thus for

$$ln \frac{p(M|D)}{p(\neg M|D)} > 0 \tag{8}$$

we have that $p(M|D) > p(E|D)$ and the naive Bayes classifier will assign the document to the Medicine class.

Evaluation of Performance

The classifier is evaluated by comparing the predicted class to the true class of the instances in the test set. From the documents in the test set 295 documents are classified as Electronics and 199 are classified as Medicine, which corresponds to a proportion of 59.72 % and 40.28 %, respectively. We know that the true proportion of Electronics and Medisine documents in the test set are 51.82 % and 48.18 %. The error rate, or the percentage of incorrect classifications for the naive Bayes model is 8.3 %. The confusion matrix in Table 3 show the correct and incorrect classifications from the naive Bayes classifier.

	True classes	
Prediction	Electronics	Medicine
Electronics	255	40
Medicine	1	198

Table 3: Confusion matrix showing correct and incorrect classifications

From the confusion matrix we have the correct classifications on the diagonal, while the off-diagonal elements represent the incorrect classifications. We see that the Electronics class has a high accuracy, only one document from the Electronics class was incorrectly classified as Medicine. For the Medicine class, 40 documents were incorrectly classified as Electronics. The overall prediction accuracy of the naive Bayesian classifier is 91.70 %. It classifies 99.61 % of the Electronics documents correctly, while 83.19 % of the Medicine documents. This is considered a good balance.

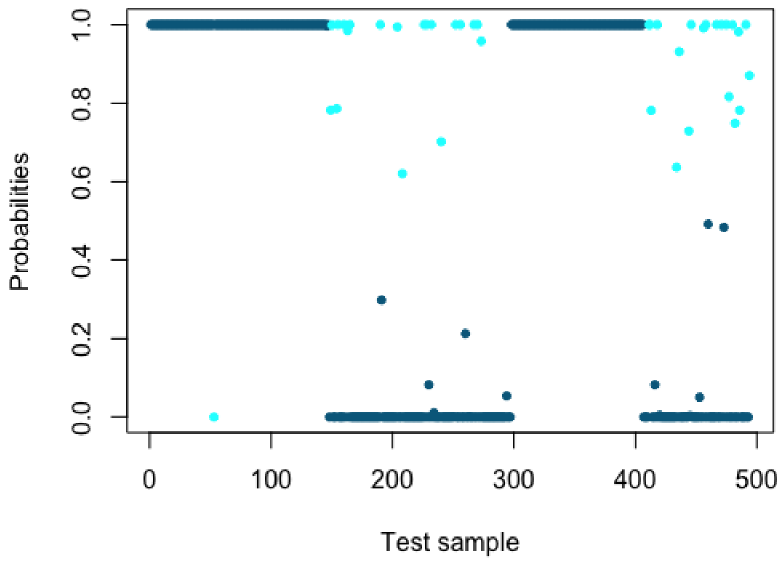

Figure 3: Probability plot of correct classifications in blue and incorrect classifications in cyan

We further calculated the class probability for each document instance in the test set. Figure 3 presents the probability of classifying the documents in the Medicine class. It illustrates the correct and incorrect classifications in the test sample, where the blue points are correct classifications and the cyan points are incorrect classifications. From this figure we can see that the in the predicted Medicine class, there is only one incorrect classification, represented by the cyan point in the lower left corner. However quit a lot of the documents with a probability of 1, which implies that the classifier strongly believes that the document belongs to the Electronics class, actually are Medicine documents.