CS 276 Programming Assignment 4
Rukmani Ravisundaram
Tayyab Tariq

Multivariate Naive Bayes Classifier:

The Bernoulli model for Naive Bayes classification produces 77% accuracy on the test set indicated. This is much lower than the performance of the Multinomial model on the same test set. From our results, we can conclude that the Bernoulli model typically make many mistakes because of the strong assumption of presence/absence of a term being an indicator of the class. As documents get longer, the Bernoulli model will perform worse, as the presence or absence of a single can mess up the classification because of some noise features. To avoid overfitting, as part of Deliverable 1, feature selection was performed. The method used was based on mutual information. The accuracy on the test set improves to 84.5% after using mutual information feature selection as a means to avoid overfitting.

Multivariate Naive Bayes Classifier with χ^2 :

A way to improve the Bernoulli model is to perform feature selection. To prevent the assumption of the occurrence/non-occurrence of every word being an indicator of the class from dominating the result of the Bernoulli model, we select only those words that are strong indicators for each class. This will weed out noisy features like common words that occur ubiquitously, rare words in documents that actually do not bear any relation to a class, words that occur only in one class but are in reality not strong indicators of the class. We performed χ^2 feature selection on the Bernoulli model and retained as features only words that appear among the top 300 χ^2 scores in each class. This gives a considerable improvement in performance to 88.5% on the same test set. Feature selection tends to improve a Bernoulli model, not on a multinomial Naive bayes model, as the frequency counts of a term sufficiently downweight noise features i.e words that are not strong indicators of belonging to a particular class.

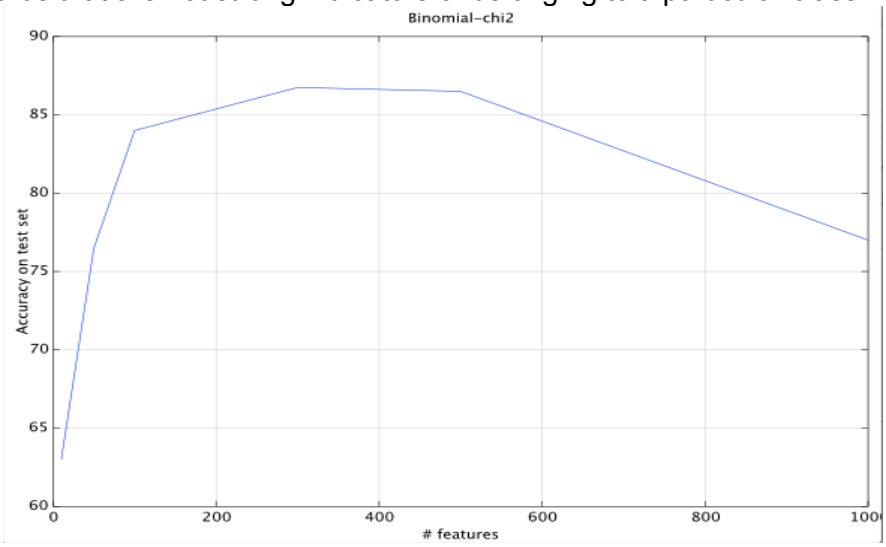

Figure1. Performance of Binomial Naive Bayes with χ^2 feature selection varying with the number of features used in classification

Multinomial Naive Bayes Classifier:

The multinomial Naive Bayes classifier performs better than the Multivariate Naive Bayes classifier because of its makes use of the actual number of times a word appears in the document. This allows the conditional probability distributions to be more representative of the generative process that created the document. This allows the Multinomial Naive Bayes classifier to achieve an accuracy of 96.75% which is much higher than that achieved with

Multivariate Naive Bayes classifier. Moreover, we experimented with different ways of using the text from the subject as the document body. We experimented with giving different weights to the the conditional probability distribution and the conditional probability distribution over subjects. However, concatenating the message body and subject gave us the best results.. We also applied a Laplacian smoothing with k=1. We performed χ^2 feature selection on the and retained as features only words that appear among the top 300 χ^2 scores in each class. Although this caused the accuracy to fall on the first 20 messages of each class, but it stayed stable on unseen data compared to multinomial without χ^2.

kFold Cross Validation:

For this part we implemented 10-fold cross validation on all the implemented techniques. The following table shows the average accuracy for all the implemented techniques for 10 fold cross validation.

Name	**Cross Validation**	**First 20**
Multivariate Bernoulli Naive Bayes (with correction of overfitting)	83.12%	84.5%
Multivariate Bernoulli Naive Bayes with χ^2 feature selection	81.2%	88.5%
Multinomial Naive Bayes	86.33%	96.75%
Multinomial Naive Bayes with χ^2 feature selection	83.9%	89.75%
TWCNB	86.2%	96%
Multinomial with Bigrams (Deliv 6)	87.7%	99.75%

Table 1 Results of 10 fold cross validation

The accuracy for all the compared techniques fell for cross validation. This is to be expected since the classifiers are expected to work better on seen data. The accuracy of Multinomial is better than multivariate NB model, however, this data shows that multivariate naive bayes generalizes better compared to multinomial. This is because it is less representative model and does not overfit training data too much.

Transformed Weight Normalized Complement Naive Bayes :

TWCNB improves Naive Bayes when the training data is not uniform, and words in classes are not independent (basically all the assumptions that Naive Bayes made). Since data from some classes are better represented than certain others, we can normalize this effect, by making the size of the training set for each class approximately the same using the complement idea introduced in the paper. CNB improves the multinomial Naive Bayes performance to 97.25%. This is a small improvement, but the test set being used is also not large, so the performance improvement cannot be discounted away. Weight normalization accounts for preferring certain classes that most violate the independence assumption. This gives a 96% accuracy, a slight reduction, as the training data given probably does not have any one single class dominating the weight vectors. We see that the document sizes and vocabulary sizes are almost the same across classes in the given training data and so the shortcoming that WCNB tries to address is not significant here, and so does not give significantly increased results. Finally applying the transforms as outlined in the paper give an accuracy of 96.5%. This

applies traditional tf-idf techniques and hence is expected to work well on any corpus that is bound to exhibit features of natural language, i.e frequently occurring common words and less frequent words that carry more information about classes. On the whole TWCNB does proved an improvement over the traditional multinomial model, by trying to correct for the inherent “naive” assumptions in Naive Bayes.

Deliverable 6: Multinomial with Bigrams

For this part, we changed the features set for the Multinomial Naive Bayes (our best performer so far) to bigrams instead of words. This allowed us to create a more representative model of the process that generated the messages. For the testing on the first 20 messages of each class, we obtained an accuracy of 99.75%. This high accuracy is partly because we are testing on part of the training set itself and partly because the model has better power to represent the data compared to Multinomal Naive Bayes with words as features. For ten fold cross validation, the accuracy dropped to 87.7%. This drop is in line with the drop experienced with Multinomial without bigram features.

Optional Deliverable 7: Support Vector Machines

We experimented with Support Vector Machines using LIBSVM (python version) as suggested in the handout. A comparison of Naive Bayes and SVMs are as follows. The first thing we noticed was that training a Naive Bayes classifier is roughly proportional to the time taken to read in the corpus. Whereas training an SVM takes many orders of magnitude longer than Naive Bayes. The performance of an SVM on this particular data set is 100%. Naive Bayes and any improvements to it (feature selection/complement) prove to be no match for an SVM on this data set. A linear kernel was used for the SVM in this experiment.

Optional Deliverable 8: kNN

For this part we implemented the kNN technique representing each document by its normalized tf.idf vector. We used words from both the body and subject of the document. We experimented with different values of k and settled for k=3 as it gave the best results, without making the value of k too large. We observed that representing the query document by a normalized tf vector got marginally better results compared to using tf.idf. This observation may be attributed to the doubling of idf when used in both the corpus and the query document, as discussed in the lecture. We also experimented by creating a training set using only a subset of the original document. We selected the square root of the number of documents in each class uniformly at random. However, this resulted in the accuracy falling by about 20%.

Accuracy on 400 message test set:

Note: The below accuracy figures for Deliverable 5 (CNB,WCNB,TWCNB) were calculated before the change in dataset and using the buggy version of message_features.py. Hence accuracy values obtained on the newer dataset may not be the same and are not directly comparable to those of Deliverable 1,2, and 6

Multivariate Bernoulli Naive Bayes (with correction of overfitting)	84.5%
Multivariate Bernoulli Naive Bayes with χ^2 feature selection	88.5%
Multinomial Naive Bayes	96.75%

Multinomial Naive Bayes with χ^2 feature selection	89.75%
CNB	97.25%
WCNB	96%
TWCNB	96%
Multinomial Naive Bayes (Deliverable 6)	99.75%
Support Vector Machines	100%
kNN	89.75%

Table 2. Results of Deliverables 1-6