CS 276: Programming Exercise II

Sandeep Sripada(ssandeep) & Gautam Kumar Parai(gkparai)

1 Introduction

In this project, we implemented a family of supervised machine learning methods to classify Usenet newsgroup messages. The methods include variants of Naïve Bayes Classifier like multinomial, multivariate, complement multinomial, weight-normalized complement multinomial, and transformed weight-normalized complement multinomial classifiers. We implemented several feature selection methods like chi-square, mutual information, KL Divergence, dKL Divergence along with an array of domain specific features. We compared our results with other classifier implementations in WEKA, SVM and decision trees. The data set considered for training was a 20 Newsgroups collection with 18828 messages in all and the testing set contained the top 20 messages from each class (comprising a total of 400).

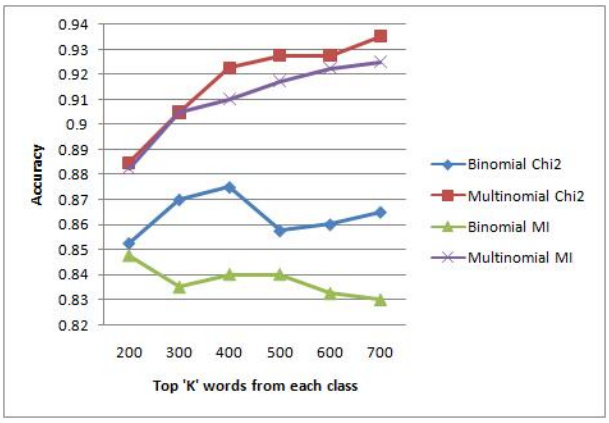

Figure 1: Comparison of feature selection methods in MVB and MNB as K varies

2 Deliverable I - Multivariate Naïve Bayes Classifier

We implemented a multivariate naïve bayes classifier (MVB) and trained it on the entire set of 18828 documents to obtain the model. The classifier took into account (a) underflow prevention issue by using log probabilities and (b) over-fitting issue by considering the Add-one or Laplace smoothing. The accuracy of MVB on a test set of 20 documents from each class is reported in Table 1.

Table 1: MNB, MVB, and MVB with chi-square

Classifier	Matches	Accuracy
Multinomial NB	383	95.75%
Multinomial NB with χ^2	362	90.5%
Multivariate NB	339	84.75%
Multivariate NB with χ^2	348	87%

3 Deliverable II - Feature selection with χ^2

Over all messages in the training set, χ^2 feature selection was employed and top 300 best discriminating features from each class were selected and merged to form a new feature set of 5613 words (total vocabulary size: 103584). The results of the retrained classifier on the new feature set is reported in Table 1. Also, the comparison of this feature selection method with mutual information can be seen in Figure 1 as the value of 'K' is varied.

Observations: The use of χ^2 as a feature selection method was helpful in case of MVB as the bernoulli model does not take into account the word frequency information present in the training data. So using χ^2 as a measure to select the most discriminating features gives this technique the edge over simple MVB. However, the χ^2 selection method is not as effective in case of a MNB because of the fact that reducing the feature set has actually stripped the model of potential information rather than adding to it.

From Figure 1 we can clearly see that the value of 'K' plays an important role in determining the accuracy of the classifier and the overall training time. There is a trade-off both on the accuracy and performance front between the value of 'K' and the accuracy. A small value of 'K' tended to remove the most discriminating features which are essential for classification whereas a large 'K' did not give high accuracies as it leads to overfitting. We found that the accuracy is highest around the top 400 mark. We compared χ^2 method with other feature selection methods like mutual information (MI), KL Divergence, dKL Divergence and found that the χ^2 method outperformed the other methods in most cases.

Table 2: k-fold cross validation (correct predictions)

Classifier	k=5	10	15	20	40
MNB	348	351	353	352	346
MNB with χ^2	339	339	341	340	333
MVB	302	304	303	304	303
MVB with χ^2	328	326	328	330	323

4 Deliverable III - Multinomial Naïve Bayes Classifier

We implemented another member of the naïve bayes classifier family called multinomial naïve bayes classifier (MNB). The classifier was trained on the entire set of 18828 documents to obtain the model. The accuracy of MNB on a test set of 20 documents from each class is reported in Table 1.

Observations: We observed that the Multinomial Naïve Bayes Classifier performs much better than the Multivariate Classifier (Bernoulli), both with and without the feature selection. The accuracies mentioned as part of this experiment are very high because the testing was done on a subset of the data used for training. More analysis on this is mentioned in the Section 5.

The classifier gave 383 correct classifications on the test data set of 400 documents with an accuracy of 95.75%. The Multinomial Classifier had a much higher accuracy than the Bernoulli classifier. This is expected as the MNB classifier took into account the position of the words in the documents while this was completely ignored in the MVB classifier.

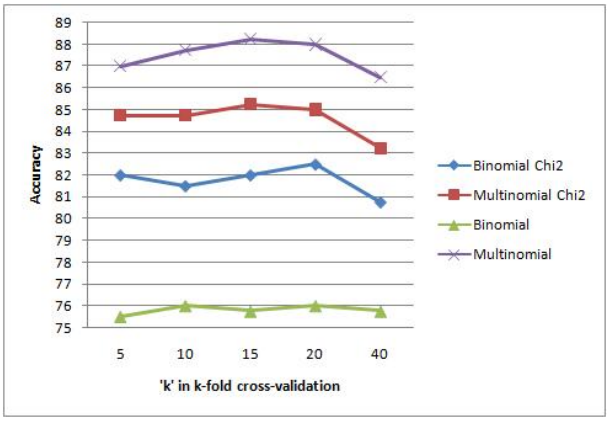

Figure 2: Comparison of cross-validation accuracies as k varies on MNB and MVB with and without χ^2 selection

Table 3: Improvements to Naïve bayes classifier (LN: Length Normalization

Stage	Correct Predictions	Accuracy
MNB	383	95.75%
CNB	383	95.75%
WCNB	383	95.75%
TWCNB(no LN)	389	97.25%
TWCNB(with LN)	377	94.25%

5 Deliverable IV - k-fold cross-validation

The accuracies in the previous sections are high because of the fact that the testing was done on a subset of the data used for training. This bias can be corrected using the *k-fold cross-validation* technique where the training set is converted into 'k' subsets and for each of the 'k' iterations one of the subsets is used as testing dataset and the others as training sets. This would test the classifier in its ability to classify unseen samples.

We implemented the cross-validation technique by randomly choosing samples and placing them into 'k' bins. As the samples are chosen at random rather going sequentially through each class and filling the bins, it would be a better test of the robustness of the classifier to unseen data. We performed k-fold cross-validation for different values of 'k' on the MNB and MVB classifiers with and without feature selection. The accuracies averaged over all iterations are reported in Table 2 and the plot is shown in Figure 2.

Observations: We performed the tests on various values of 'k' i.e. 5,10,15,20,40 and observed that the average cross-validation accuracies are low because the testing was done on unseen data unlike the previous sections. The accuracy also seemed to improve as 'k' increases but also fell after a certain value of 'k'. As 'k' increases each of the subsets formed have $|docs|/k$ documents which results in a smaller testing set and larger training sets (relatively with previous 'k') causing a case of over-fitting to come into picture and thereby decreasing the accuracy slightly. Also, for very high values of 'k' performance is an issue and we found that the value of 'k' is optimal around 10-20 subsets on our dataset.

6 Deliverable V - Improving the Naïve Bayes Classifier

We implemented and experimented with the improvements to the Multinomial Naïve Base Classifier as described in [Rennie2003]. The experiment was carried out using 18828 documents as the training set and 400 doc-

uments (20 from each class) as the test set. The implementation comprised of 5 sub stages, the results of which are depicted in Table 3.

- Term frequency transform (TF)
- Inverse document frequency transform (IDF)
- Length normalization (LN)
- Complement naïve bayes (CNB)
- Weight normalization (TWCNB)

Observations: We observed that the accuracy of the classifier improved quite a lot after implementing the changes as mentioned in [Rennie2003]. The accuracy of CNB and WCNB was same as that of the MNB and the accuracy of the TWCNB without length normalization was higher and by far the best accuracy attained so far. However, the inclusion of length normalization resulted in a decreased accuracy of 94.25% from 97.25% as the normalization worsened the parameters in a fairly uniform length message dataset. TWCNB also performed better that MVB as it takes into account various factors like term frequency which were completely ignored by MVB. It also did very well compared to the classifiers that included feature selection methods like χ^2, mutual information, KLD, and dKLD.

With the changes incorporated, TWCNB, is very close in terms of accuracy to various other state-of-the-art classifiers like SVMs, Decision trees etc. TWCNB performed comparably to most of the classifiers but was way ahead in terms of the performance (training time). It was faster than SVMs with radial and polynomial kernels, Decision Tables and most other sophisticated classifiers. The simple changes including traditional IR concepts proved effective as shown by the performance of TWCNB.

The following section details the list of domain-specific features we tried on MNB. To show the improvements more evidently we have chosen MNB as TWCNB with already such high accuracy would not help us in analyzing the improvements effectively. However combining these techniques to TWCNB would definitely be helpful in improving its accuracy. We have also compared the results with the accuracy ratings in MVB.

7 Deliverable VI - Experimenting with different techniques

There are two broad types of techniques that can be employed to improve the accuracy of a classifier (a) Domain specific feature selection and (b) Domain independent (statistical) feature selection.

Under (a), we have experimented with the following: (i) lower casing and word stemming, (ii) upweighting zones, (iii) number processing, (iv) email processing, (v) signature processing, (vi) hyperlink processing and (vii) bigram words.

Under (b), we tried the following: (i) χ^2, (ii) Mutual Information, (iii) KLD, and (iv) dKLD.

We have implemented all the above mentioned domain specific features on the MVB and MNB classifiers. The experiment was carried out using 18828 documents as the training set and 400 documents (20 from each class) as the test set.

Observations:

A.1 We observed that for MNB classifier with/without lowercasing and stemming there is no improvement in the classification accuracy whereas for MVB classifier there is a slight improvement in the classification accuracy.

A.2 We observed that upweighting the subject zone by an upweightfactor of 2 in the documents improves the classification accuracy for both the MNB and MVB classifiers.

A.3 We performed two types of number processing a) replace all numbers with the same token 'NUMBERID' b) ignore all numbers. We observed that in both the cases the performance of MNB and MVB classifiers doesnt improve at all.

A.4 We performed two types of email processing a) replace all emailIDs with the same token 'EMAILID' b) ignore all emailIDs. We observed that in both the cases the performance of MNB and MVB classifiers doesnt improve at all.

A.5 We observed that in many messages, users had put their signatures at the bottom of the messages, which are not related to the email content. Therefore, we performed signature stripping in all the messages. We used the following rules a) Search for patterns of special characters (that usually appeared in signatures) in the last 5 lines of the messages and remove the lines following them. b) Search for patterns of special characters (that usually appeared in signatures) in the last 8 lines of the messages and remove the last 5 lines from the message. Although, our method for signature stripping is not robust it performed reasonably well as we found by manually checking the stripped messages randomly. We observed that the accuracy of both MNB and MVB classifiers slightly decreased, when signature stripping was applied.

A.6 We performed two types of hyperlink processing a) replace all hyperlinks with the same token 'HYPERLINKID' b) ignore all hyperlinks. We observed that in both the cases the accuracy of MNB remains the same whereas the accuracy of MVB classifier improves slightly.

A.7 We observed that including word bigrams greatly improves the accuracy of the MNB classifier. We found that it beats all other classifiers except SVMs. However, the accuracy of the MVB classifier was decreased.

B.1 Same as Section 3.

B.2 Over all messages in the training set, mutual information feature selection was employed and top 300 best

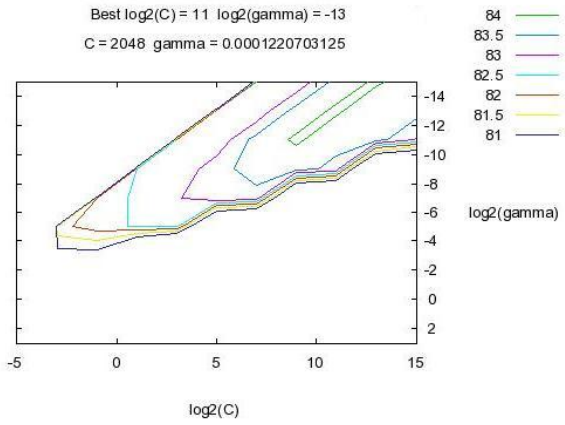

Figure 3: Grid search for optimal values of (C, γ) in RBF kernel

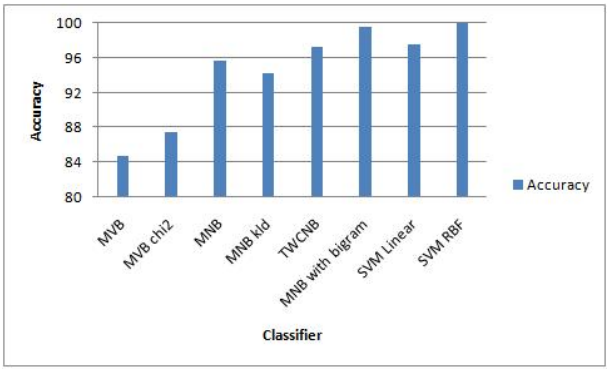

Figure 4: Comparison of accuracies of various classifier and best extensions

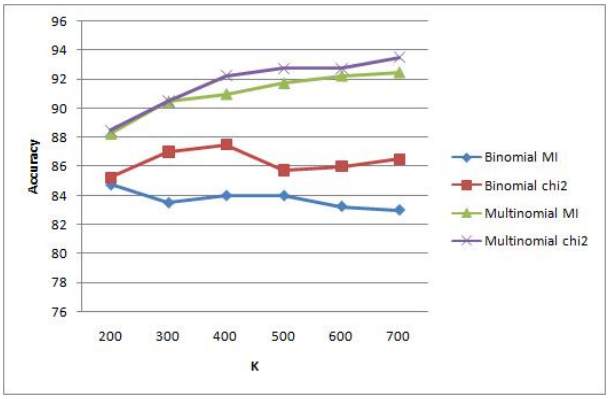

Figure 5: Feature selection with various methods

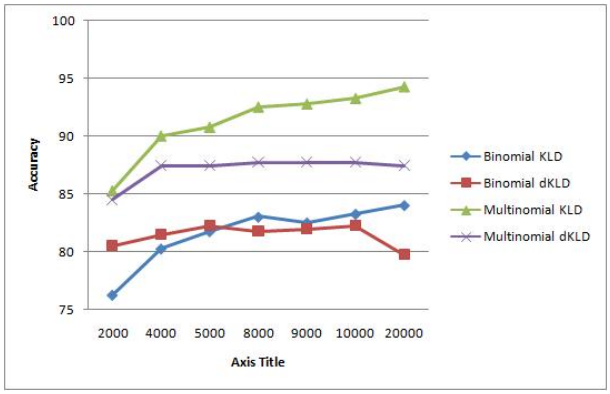

Figure 6: KLD and dKLD methods

Table 4: Support Vector Machines

Kernel	**C**	γ	**Mode**	**Acc**
Linear	15	-	logProb	87%
Linear	200	-	logProb	93.5%
Linear	1000	-	logProb	97.5%
Linear	15	-	TF	78.75%
Linear	200	-	TF	83%
Linear	1000	-	TF	88.25%
Radial	15	5	TF	100%
Radial	1	1	logProb	100%
Radial(Best)	2048	$1.7E^{-3}$	logProb	100%

discriminating features from each class were selected and merged to form a new feature set of 4549 words(total vocabulary size: 103584). This was done for comparison with mutual information. The results of the retrained classifier on the new feature set is reported in Figure 5. We found that mutual information performs worse than chi-square.

The best combination of these features is: Bigrams, hyperlink processing, upweighting (lowercasing and stemming) where the accuracy is 99.5% in MNB and 75.25% in MVB.

8 Deliverable VII - Support vector machines

We also looked at multi-class Support Vector Machines (SVMs) in classification using the $SVM^{multiclass}$ implementation[1]. Support Vector Machines are state-of-the-art in classification. Using multi-class SVMs we conducted experiments with different kernels and corresponding kernel parameters.

The training data consisted of words selected using χ^2 feature selection method by taking top 300 from each class and merging them resulting in a 5613 dimensional

[1] `http://svmlight.joachims.org/svm_multiclass.html`

feature vector. The training samples included all 18828 documents from the 20 Newsgroups and the test samples included 400 samples with the same feature vector dimension as the training data. This would allow us to analyze the accuracies by comparing them to MNB and MVB. The SVM performed exceptionally well for certain kernels although the training time was extremely high in some cases like the polynomial kernel. We have used both term frequency and multinomial log probabilities as feature values in the experiments. The accuracies are reported in Table 4.

Observations: The accuracies were quite high with the use of SVMs especially in the case of Radial Basis function. We tried various values of C and γ for RBF and the values 'best' were obtained using the grid search method as described in [Chih09]. We tried exponentially growing sequences of (C,γ). C ranged from 2^{-5} to 2^{15} and γ ranged from 2^{-15} to 2^{3}. The best contour obtained after searching gave the optimal values of (C,γ) which were (2048, $1.7E^{-3}$) as shown in 3. The accuracy attained was 100% on the test set but the extremely high training times of these machines could be a severe drawback in cases with large sample data and features. The comparison of the accuracies with other classifiers and their best extensions is shown in 4.

9 Deliverable VIII - Other classifiers using WEKA

We performed experiments on the Usenet newsgroup dataset using the classifiers provided by the Weka Library. The Weka Library is an off-the-shelf machine learning library providing implementations of various classifiers. We used a training set of 400 documents with 20 taken from each class and a 100 document testing set with 5 taken from each class. The reason for this reduction in training and testing set is due to the amount of memory and taken to learn the models of various classifiers. We generated the arff files and used WEKA GUI (using java -jar weka.jar) to perform the experiments.

We have experimented with a) REPtree with bagging, b) Decision Table, c) Decision Tree, d) M5P, e) IBK (K nearest neighbors), f) MNB, and g) CNB.

A. Bagging We performed bagging using the REPtree class on the datasets. The accuracy was around 87% for both the Term frequency and the log probabability feature values. Bagging also took care of efficient dataset usage and performed quite well on the dataset.

B. Decision Table We used a majority class implementation in the decision table and ran the tests on the datasets. Another parameter that was configured was the maximum number of stale states to consider before stopping which was set to 5. The accuracy was quite low at 36% and could be possibly because of the small dataset and noise.

C. Decision Tree We applied the Decision Tree Classifier (decision tree is a simple structure where non-terminal nodes represent tests on one or more attributes and terminal nodes represent decision outcomes). We used the J48 implementation which WEKA uses. The accuracy we obtained was around 83% using this classifier. Parameter: C=0.25, M=2.

D. M5P This is a variant of the algorithm invented by Quinlan. We achieved a co-relation coefficient of 0.54 between the actual classes and predictions with an mean absolute error of 4.0283 and a root mean square error of 4.8582 on our datasets. Parameter: M=4.

E. IBk We tested our data using Weka's IBK classifier varying the k value between 2 and square root of the number of samples. The IBk Classifier uses the K-nearest neighbor classification algorithm. The accuracy for the classification decreased as 'k' increases as it tends to look at a high number of neighbors before assigning a class. The accuracies are as follows: k=2, Acc=85.5%, k=3, Acc=85%, k=20, Acc=64.5%, k=137 (sqrt(18828)), Acc=43%. For this experiment, we ran the test on the entire dataset of 18828 samples as a reduced data set would reduce the accuracy because of lower number of samples to choose from as neighbors.

F. MNB We ran the MNB in WEKA on the dataset and found the accuracy to be 90.5%. The reduced value maybe because of the dataset considered.

G. CNB We ran the CNB in WEKA on the dataset and found the accuracy to be 88.5%. The reduced value maybe because of the dataset considered.

10 Deliverable IX - Additional feature selections

KL Divergence - Over all messages in the training set, KL divergence feature selection as mentioned in [Schneider04] was employed and top K best discriminating features were used to form the new feature set.(total vocabulary size: 103584). This was done for comparison with mutual information. The results of the retrained classifier on the new feature set is reported in Figure 6. We found that the best accuracy for KL Divergence and ChiSquare for (almost) same number of features is comparable.

dKL Divergence - Over all messages in the training set, dKL divergence feature selection as mentioned in [Schneider04] was employed and top K best discriminating features were used to form the new feature set.(total vocabulary size: 103584). This was done for comparison with mutual information. The results of the retrained classifier on the new feature set is reported in Figure 6. We found that dKL divergence has lower classification accuracy as compared to KL divergence.

For the MVB classifier we observed that dKL divergence performed better than KL divergence on smaller feature sets whereas for the MVB classifier dKL divergence performed worse than KL divergence on all feature sets. Both KL Divergence make strong assumptions (i) the number of occurrences of a word is the same in all documents that contain the word, (ii) all documents in the same class have the same length. which we found in violation on our training data which contributed to its lower accuracy in comparison to chi2.

References

[Rennie2003] Tackling the Poor Assumptions of Naïve Bayes Text Classifiers.
Jason D. M. Rennie, Lawrence Shih, Jaime Teevan and David R. Karger.
Proceedings of the Twentieth International Conference on Machine Learning (ICML). 2003.

[Schneider04] A New Feature Selection Score for Multinomial Naïve Bayes Text Classification Based on KL-Divergence.
Karl-Michael Schneider.
42nd Meeting of the Association for Computational Linguistics (ACL 2004).

[Chih09] A Practical Guide to Support Vector Classification.
Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin.