CS276 Practical Exercise #2

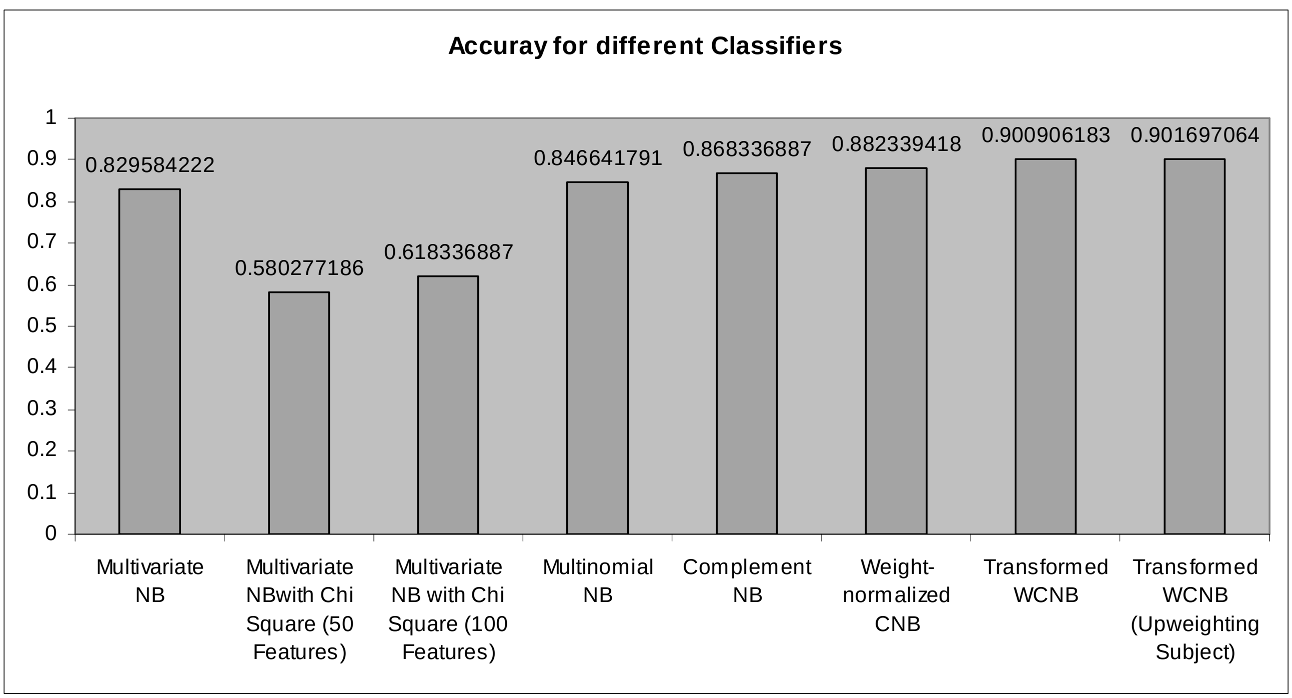
Rashmi Raj, Ashutosh Saxena

Abstract: In this report, we use a supervised learning approach to classify Usenet newsgroup messages using various techniques. Besides extra credit classification using SVM and nearest neighbor, we proposed a new version of nearest neighbor algorithm, accuracies for which are reported in the text.

Implementing Naïve Bayes and its different variants: We implemented different variants of Naïve Bayes and applied improvement as suggested by Rennie et al paper. For verifying the result we implemented K-fold cross fold (K=10) by dividing the data in K-subset and using one subset for testing and rest for training. We repeated this for each subset and collected the accuracy for each K. Here are the results tabulated for 10-fold cross-validation:

	Multivariate NB	Multivariate NB with Chi Square (50 Features)	Multivariate NB with CHI Square(100 Features)	Multinomial NB	CNB	WCNB	TWCNB	TWCNB(Up weighting Subject)
K=0	0.8310235	0.581023454	0.608742004	0.8422175	0.86141	0.87342	0.8971215	0.89823457
K=1	0.826226	0.584754797	0.618336887	0.8443497	0.8678	0.88453	0.8965885	0.8965891
K=2	0.8288913	0.607675906	0.632196162	0.8555437	0.87687	0.87894	0.902452	0.90326798
K=3	0.8278252	0.57249467	0.615671642	0.8400853	0.86354	0.88321	0.8939232	0.89478671
K=4	0.8406183	0.577292111	0.620469083	0.8432836	0.8726	0.89255	0.9104478	0.91045672
K=5	0.8400853	0.583688699	0.630063966	0.8544776	0.87047	0.87984	0.8997868	0.89980459
K=6	0.8182303	0.57782516	0.62206823	0.8464819	0.86834	0.88125	0.9083156	0.90832789
K=7	0.8331557	0.577292111	0.609275053	0.8416844	0.86301	0.88308	0.8971215	0.89832457
K=8	0.8192964	0.567164179	0.600746269	0.848081	0.8694	0.87987	0.8987207	0.89883562
K=9	0.8304904	0.573560768	0.625799574	0.8502132	0.86994	0.8867	0.9045842	0.90834289
Mean	0.8295842	0.580277186	0.618336887	0.8466418	0.86834	0.88234	0.9009062	0.90169706

Table 1: Accuracy result for different values of K



Plot 1: Comparison of accuracy for different classifiers

Accuracy discussion and Interesting Findings for Naïve Bayes classifier and its variants:

- Naïve Bayes methods assume conditional independence of the data. I.e. they assume that P(word_1 | class) is independent of P(word_2 | class). Although, this is a huge modeling assumption, these methods perform reasonably well. We could reach accuracy ranging from 80% to 90% with different variants of Naïve Bayes.
- We tried **different smoothing parameters** for Multinomial Naïve Bayes (different value of extent of smoothing (m) as discussed in the class-Lecture 10) and got best result for smoothing by adding 1 to numerator and adding k to denominator where k is number of values of terms. In the case of Multivariate Naïve Bayes k = 2 because term is either present or absent.
- Cutting down the number of features using Chi-square resulted in significant degradation of the performance. We attribute this to the fact that just 50 features are not enough to classify the documents with high accuracy. We increased features to 100 and got increased accuracy. We also tried with other values of features and found that **accuracy increased monotonically with the increase in features** though it reached close to maximum (approximately 72% accuracy) for number of features as 1000.
- Improvement suggested in Rennie et al paper gave significant improvement in accuracy.
- We also tried **combination of regular and complement classification rules (one-vs-all-but-one MNB)** and got almost similar accuracy as with CNB.
- For improving the accuracy we tried different domain specific features. We got almost .1percent increase in accuracy by **upweighting subject field (See the table 1 above)** by the factor of two. **Stemming caused small decrease in accuracy.**

Support Vector Machines for classification

We used the matlab library for SVM. Alain Rakotomamonjy [1] implemented SVM as a quadratic program [2]. We used one-against-all multiclass SVM for this purpose. We experimented with various kernels:

1. Polynomial
2. RBF

It was found that **RBF kernel gave better accuracy than Polynomial kernel**. This can be attributed to the fact that in a *very* high-dimensional space, the data is mostly linearly separable. Radial Basis function puts a prior on the data and helps improve the generalization performance.

Kernel	Mean Accuracy
SVM using polynomial kernel	0.8207
SVM using RBF kernel	0.8434

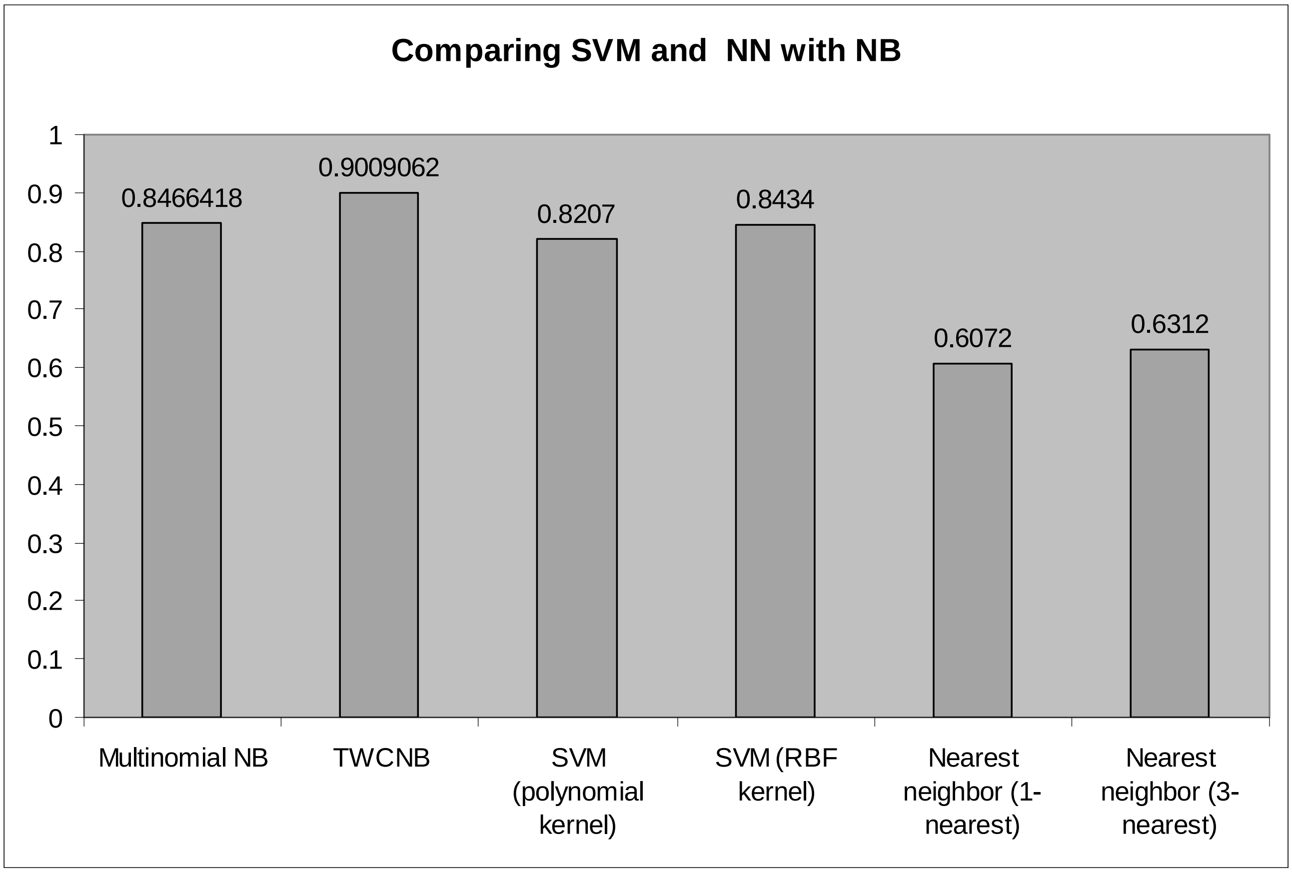
Table 2: SVM with different Kernels

Nearest Neighbor

We simply found the nearest neighbor in Euclidean space and tested using nearest neighbor and 3-nearest neighbor methods. Using 3-nearest neighbors improved the accuracy. In 3-nearest neighbors, if there was no consensus (because of the multi-class nature), we took the nearest neighbor. Because of computational reasons, the results are given on a fixed test set, without 10-fold cross validation.

K	Accuracy
Nearest neighbor (1-nearest)	0.6072
Nearest neighbor (3-nearest)	0.6312

Table 3: Nearest Neighbor for different values of K



Plot 2: **Comparison of SVM and Nearest Neighbor with Naïve Bayes**

References

[1] Alain Rakotomamonjy, SVM and Kernel Methods Matlab Toolbox. Available Online: http://asi.insa-rouen.fr/~arakotom/toolbox/index.html

[2] Stephen Boyd and Lieven Vandenberghe, Convex Optimization, Cambridge University Press.