

CS 276 Programming Assignment 2: Newsgroup Classification

Todd Sullivan
todd.sullivan@cs.stanford.edu

Ashutosh Kulkarni
ashuvk@stanford.edu

1 Introduction

In this project we implemented many variants of the Naïve Bayes classifier, including multivariate, multinomial, complement multinomial, weight-normalized complement multinomial, and transformed weight-normalized complement multinomial. We implemented several feature selection methods including Chi-Square, KL and dKL Divergence, and a baseline frequency-based method. We also thoroughly explored preprocessing methods and domain-specific features, and compared our classifiers with classifiers in the WEKA library, including SVMs and decision trees.

2 Multivariate and Multinomial Naïve Bayes

We implemented the multivariate (MV) and multinomial (MN) classifiers using Laplace's smoothing. The Underfitting was taken care of by using log probabilities instead of the rational numbers. Laplace smoothing took care of overfitting – by assigning some probability to the features that are not seen in the training dataset for a particular class.

The accuracy of multivariate, multinomial classifiers on a training dataset of top 20 documents is shown in Table 2. We would like to note that we sorted the data based on classnames and based on file names for each of the classes. The results here may differ slightly than when run on other machines without doing this type of sorting. The implementation submitted though does not contain sorting, and thus results should match with the usual results from the machine.

Table 2: Classifier Accuracy			
	MV	MN	TWCNB
First 20 documents	0.8225	0.945	0.9225
10-fold cross validation	0.7537	0.8799	0.8804

3 Feature Selection using χ^2

Our feature selection routines using Chi-Square allows us to select the top K words for each class (with words ranked by the Chi-Square calculation) and to use the union of these words as our feature set. Appendix §3 shows the top 20 words for each class as chosen by our Chi-Square routine when using the dataset that is generated from the unmodified MessageParser code. The top 20 words chosen by Chi-Square for each newsgroup are quite good and are able to summarize the category well.

In the world of text classification, using some form of feature selection will generally improve accuracy. Of course, being too aggressive and removing too many features can be detrimental to performance. At the same time, with a large number of textual features using no feature selection method can easily lead to overfitting. Aside from improving performance, feature selection can also make more complex classification methods have a feasible overall training time.

We explore all of these ideas further in Section 8 and Section 9. Section 8, which includes the other feature selection methods that we implemented, contains a thorough comparison on the various feature selection methods as K is varied. The Chi-Square method turns out to be our best and is required in Section 9 where we compare our Naïve Bayes classifiers

with other types of classifiers from WEKA that cannot handle hundreds of thousands of features.

4 Built for Speed

All of our classifiers were implemented with efficiency in mind, and do not take any more than 30 seconds to perform 10-fold cross validation. Using the original dataset as generated by the unmodified MessageParser code (which contains 103,584 features), our multinomial classifier takes 2.4 seconds for 10-fold CV while our multivariate takes 4.9 seconds and our CNB, WCNB, TWCNB classifiers in later sections take 2.8, 3.3, and 8.8 seconds respectively.

5 K-fold Cross Validation

We implemented standard k-fold cross validation. The routine places documents randomly into K buckets and then performs training/testing K times where on each iteration one of the K buckets is used for testing and the other K – 1 are used for training. Table 5 shows the average classification accuracy by newsgroup for our multivariate and multinomial classifiers when using 10-fold cross validation on the unmodified MessageParser code dataset. The columns with χ^2 in their header used Chi-Square feature selection with the top 300 words from each class used as features. The full dataset classifiers (first two columns) used 103,584 features while the Chi-Square classifiers used 5,613 features (not 6,000 due to overlap in the top 300 lists).

5.1 Full Dataset

A quick glance at Table 5 shows that the multivariate classifier is good at distinguishing classes that have little overlap with others, such as rec.motorcycles and rec.sport.baseball. These newsgroups are rather specific and contain a limited vocabulary in comparison to other newsgroups. For newsgroups that can include a wide range of discussions, such as talk.politics.misc and talk.religion.misc, the multivariate classifier exhibits terrible per-

mance. This can also be seen when comparing alt.atheism with soc.religion.christian. The Christian newsgroup is much more narrow (only covering a subarea of religion) while the atheism newsgroup includes discussions about all religions as well as various philosophical areas and humanism.

The multinomial classifier is able to overcome many of the multivariate’s deficiencies that arise from only looking at the presence/absence of words. The multinomial’s incorporation of word frequency allows the classifier to make marked gains over its multivariate sibling, with a jump from 6% to 53% in talk.religion.misc, 47% to 87% in talk.politics.misc, and 65% to 89% in alt.atheism.

Newsgroup	MV	MN	MV χ^2	MN χ^2
alt.atheism	0.65	0.89	0.86	0.89
comp.graphics	0.6	0.88	0.78	0.82
comp.os.ms-windows.misc	0.68	0.74	0.82	0.79
comp.sys.ibm.pc.hardware	0.88	0.82	0.75	0.77
comp.sys.mac.hardware	0.91	0.85	0.89	0.86
comp.windows.x	0.75	0.88	0.81	0.82
misc.forsale	0.87	0.74	0.84	0.74
rec.autos	0.86	0.92	0.87	0.92
rec.motorcycles	0.93	0.95	0.93	0.94
rec.sport.baseball	0.91	0.96	0.94	0.95
rec.sport.hockey	0.89	0.98	0.93	0.96
sci.crypt	0.87	0.97	0.88	0.94
sci.electronics	0.81	0.84	0.82	0.80
sci.med	0.76	0.95	0.85	0.92
sci.space	0.78	0.96	0.83	0.93
soc.religion.christian	0.85	0.95	0.87	0.91
talk.politics.guns	0.81	0.95	0.88	0.92
talk.politics.mideast	0.74	0.97	0.80	0.92
talk.politics.misc	0.47	0.87	0.75	0.81
talk.religion.misc	0.06	0.53	0.50	0.60
Macro-average	0.75	0.88	0.83	0.86

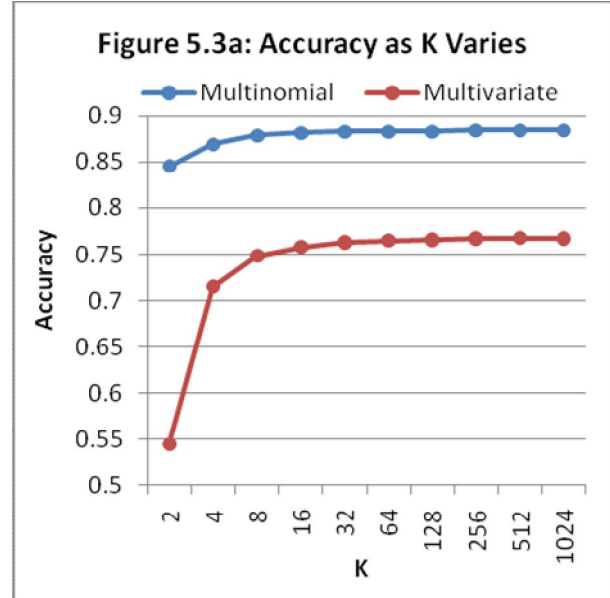
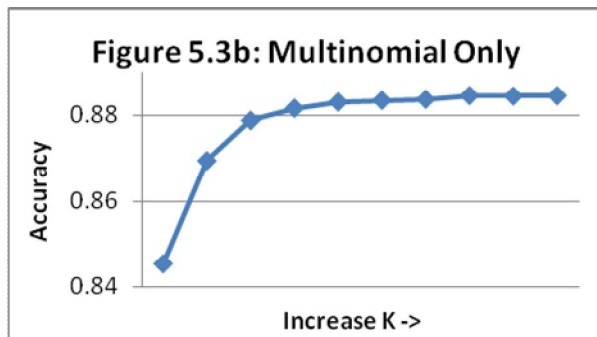
5.2 Chi-Square(300) Dataset

Decreasing the number of features from 104,000 to 5,600 caused a slight drop in per-

formance in the multinomial classifier. This does not mean that the feature selection method is bad, because we were able to decrease the number of features by a factor of 18 while only losing 2 percentage points in macro-averaged accuracy. The drastic cut in features significantly improved the performance of the multivariate classifier. The talk.religion.misc newsgroup accuracy had the largest gain from 6% up to 53%! Not all classes benefitted though: comp.sys.ibm.-pc.hardware dropped from 88% down to 75%.

5.3 Changing K

Figure 5.3a shows the macro-averaged accuracies of both classifiers using the full dataset as K is varied. We see a large change in accuracy when moving from 2 to 4 folds, or 4 to 8 folds, but the accuracy levels off quickly after 8. K-fold cross validation helps with the problem of wasting data when training/testing, decreasing variance and making us less likely to overfit. The accuracy for each classifier appears to be converging, but for each doubling of K after 8 or 16 (which doubles the amount of times one must train/test) we see very little change in score. With 1024 folds, the multinomial's accuracy is 88.42% and the multivariate's is 76.72%. The values in Table 5 from 10-fold CV are close enough to the 1024-fold CV, which explains why K=10 is used widely. Figure 5.3b, a close-up of the multinomial accuracy, shows that it also exhibits the same trend as the multivariate.



6 Transformed Weight-normalized Complement Naïve Bayes

We implemented the improvements in multinomial classifier in three steps – Complement Naïve Bayes (CNB), Weight normalized CNB (WCNB) and transformed WCNB (TWCNB). The accuracy on the training dataset for the first 20 documents from each class is as shown in Table 2. Note that, using the first 20 documents as a test dataset isn't a good measure of the classifier since that data is already used as training data for the classifiers. A better measure is to use k-fold cross validation for comparison. Indeed as seen in the k-fold cross validation comparison, CNB, WCNB and TWCNB work better than multinomial classifier. On the original dataset as generated by the unmodified MessageParser code, the MV, MN, CNB, WCNB, and TWCNB macro-averaged accuracies using 10-fold cross validation are 0.7537, 0.8799, 0.8882, 0.8848, and 0.8804 respectively

6.1 Complement Naïve Bayes:

Complement Naïve Bayes improved the performance of the Multinomial classifier. The most important reason being more training data per class. The fact that the training data is

more evenly distributed also helped improve the performance.

6.2 Weight normalization

We found that weight normalization does not really help improve the performance of the system. Weight normalization tries to remove the bias that is caused by the independence assumption of the multinomial Naïve Bayes classifier. [1] mentions that, weight normalization tries to reduce the bias of MNB towards dependency between words (giving more influence to classes that most violate independent assumption).

After going through our specific dataset, we realized that each newsgroup’s overall topic can be easily described by single independent words. Thus our dataset does not have the same issues as described in [1] with the topics of “Boston” and “San Francisco”. Weight normalization thus did not really help improve the performance and instead added more noise to CNB by trying to account for a dependency assumption that does not exist.

6.3 Transformation

We used the three types of transforms described in the paper. The comparison of the transforms with WCNB is as shown in the Table 6.3. Again, the results are from the dataset generated by the unmodified MessageParser code.

We see that using 10-fold cross validation, the term frequency transform and IDF transform work well and perform better than WCNB with no transformations. The length transform on the other hand decreases the performance of the classifier. We notice that length normalization is generally used to deal with the negative effects caused by longer documents. But in our specific dataset, the longer documents are usually the replies to the previous posts from other users and contain the part of the previous post that is carried forward. Thus, the new text usually being written by a new author is fairly independent

of the text in the previous post – even though it is in the context of the post. Therefore the assumption that if a word occurs in the document, it is more likely to appear again is not effective in this particular dataset.

According to our results, we suggest that the best model for this particular dataset would be to use CNB with Term frequency and IDF transform and without weight normalization. Indeed the 10-fold cross validation average accuracy of this model is better than the others – 89.61 %. (0.896124).

Table 6.3: Classifier Accuracy					
	W CNB	TF	IDF	TF- IDF	Length
First 20 docs	0.95	0.9475	0.97	-	0.8575
10-fold CV	0.8804	0.8880	0.8874	0.8894	0.8373

7 Preprocessing Techniques and Domain-Specific Features

Preprocessing each newsgroup email and including domain-specific features can give modest performance gains. We employed a variety of preprocessing methods and tested each combination on our classifiers. Table 7 lists each preprocessor or domain-specific feature that we tried.

We include five stemming options. The Porter2 stemmer is a modified Porter stemmer that Dr. Porter created in 2001/2002. The Lovins Iterated stemmer repeatedly applies the Lovins stemmer until there are no additional changes.

Table 7: Preprocessing and Extra Features	
Method	Possible Values
Stemming	None, Porter, Porter2, Lovins, Lovins Iterated
Lowercasing	On / Off
Stop List	On / Off
sbSame	On / Off
Weight	1:1, 2:1, 1:2
From	On / Off
Bigrams	On / Off
Noun N-Grams	On / Off
Bible Quotes	On / Off

The sbSame option indicates whether or not the subject and body words are considered the same features. If sbSame is set to Off, then the word “Cow” in the subject of an email is considered a separate feature from the word “Cow” in the body. If sbSame is set to On, we can give the two zones equal weight with Weight = 1:1, give the subject twice its actual counts with 2:1, etc. Our experiments found no benefit to giving unequal weights. Thus all results presented in this section have Weight = 1:1.

With the From option set to On, we include the From headers as part of the subject zone, otherwise the From headers are ignored. With Bigrams set to On we include all bigrams for tokens that are determined to be words and not numbers. The Noun N-Grams option sets as features all n-grams formed by successive words that have their first letter capitalized. Only the longest n-gram is used as a feature, so if the full n-gram is “Better Business Bureau”, “Better Business” will not be added as a feature.

Our Bible Quotes feature was developed as an attempt to improve the dismal performance in the talk.religion.misc newsgroup. Many discussions about religion include references to religious texts where the quote will be followed by a reference to the text such as “(John 1:18)”. When Bible Quotes is turned on, an occurrence of this pattern increments the counter for a special “<BIBLE>” token.

Unfortunately, our preliminary investigation showed our Bible Quotes feature to have no improvement on performance. We also tried other features such as counting all numbers as a single “<NUM>” token, and using similar features for hyperlinks and email addresses. We found that none of these feature changes improved performance, and thus all results in this section do not include these features.

7.1 Results

After eliminating a few of our feature ideas such as Weight and Bible Quotes, we were left with 320 different combinations of the methods in Table 7. We processed the newsgroup data using all combinations, resulting in 320 different datasets. Since our CNB classifier was our best-performing classifier in our previous trials, we used CNB with 10-fold cross validation on all 320 datasets.

To determine which methods in general improved accuracy, for each setting of each method we calculated the average macro-averaged accuracy and average total feature count from using CNB with all datasets that were generated using the setting of the given method. Thus for Stemming set to None, which we will denote Stemming(None), we calculated the averages over all datasets where no stemming was used. Table 7.1 presents each setting of each method ordered by average macro-averaged accuracy.

Table 7.1: Average Accuracy and Feature Count by Method/Setting

Method/Setting	Accuracy	Feature Count
Stemming(None)	0.9226	411,797
Bigrams(On)	0.9195	531,559
From(On)	0.9146	338,857
sbSame(Off)	0.9132	340,193
Stemming(Porter2)	0.9122	360,452
Stop List(Off)	0.9122	415,393
Stemming(Porter)	0.9119	355,910
Lowercasing(Off)	0.9115	342,479
Noun N-Grams(On)	0.9109	346,426
Average from All Datasets	0.9098	331,665
Noun N-Grams(Off)	0.9088	316,905
Lowercasing(On)	0.9082	320,852
Stop List(On)	0.9075	247,938
sbSame(On)	0.9065	323,138
From(Off)	0.9051	324,474
Stemming(Lovins)	0.9037	277,453
Bigrams(Off)	0.9002	131,772
Stemming(LovinsIterated)	0.8989	252,716

We see that the most important methods are not stemming, including bigrams, and including the From headers in the subject zone. It is best to not use a stop list and not lower-case the text. The Noun N-Grams method also gives a slight boost. If one must use stemming, the Porter2 stemmer is best, quickly followed by the original Porter stemmer. The Lovins and Lovins Iterated stemmers are two of the worst options to use.

The downside to all of the accuracy-improving options is that they all increase feature count. Indeed, a lack of stemming, including bigrams, including the From headers, treating the subject and body words as separate features, not using a stop list, not lower-casing text, and including noun n-grams all increases the feature count. While our Naïve Bayes classifiers are fast to train, other classifiers have much worse training time. A larger number of features can also increase the likelihood of overfitting.

As an interesting example, four of our top datasets with macro-averaged accuracies of 93.1% have up to 964,000 features, with an average of 895,371 features. All four of these datasets had Stemming(None), Stop List(Off), From(On), Bigrams(On), and sbSame(Off). The datasets differ in that they include all four combinations of Lowercasing On/Off with Noun N-Grams On/Off.

Interestingly, four other datasets that have 232,000, 175,000, 154,000, and 208,000 features respectively all achieve 99.98% macro-averaged accuracy. These datasets all include Stemming(None), Lowercasing(Off), Stop List(Off), From(On), and Bigrams(Off) with the datasets differing by the four combinations of Noun N-Grams On/Off and sbSame On/Off. All of these datasets have below 100% accuracy for the sci.electronics newsgroup, two have below 100% accuracy for talk.religion.misc, and one has below 100% accuracy for comp.sys.ibm.pc.hardware.

Normally we are concerned when achieving results this close to perfect because it indi-

cates that our train/test routines are flawed or that somehow each document's truth value is being shared with the classifier during training. We do not see how this could be the case since all 360 datasets were generated with a single pass of an automated program that operates based on the flags set for each of the methods. It is unlikely that the exact set of flags that all four datasets have in common would exhibit such a bug. Additionally, the same routine uses all of the datasets to train/test the CNB classifier and, being completely unaware of the specific dataset it is using, is not likely to be giving erroneous results.

We posit that this extreme performance is because the From headers include the email addresses of the senders of each email, and many senders do not participate in all newsgroups. With Bigrams(On), all of the additional features created noise and caused the accuracy to reach only 93.1%. By throwing away the hundreds of thousands of additional features, the From headers' influence became greater, which led to accuracy improvements.

A close examination shows that this appears to be the case. Our multinomial classifier achieves 99.8% accuracy on these datasets. We trained our multinomial classifier using one of the datasets, and then outputted the top 20 features for each class (in terms of the conditional probability of the feature given the class). All of the classes have at most only 3 features from the body zone. All other features in the top 20 are from the subject. Most of the features are email addresses. For example, alt.atheism gives 1.6% of the class' conditional probability mass to a single email address, and that email address only exists in alt.atheism emails. Two other features are the first and last name of the person with the previous mentioned email address, and they garner 2.3% and 1.6% of the probability mass respectively.

In a different dataset that includes bigrams, the massive amount of features drowns

out the From headers and our multinomial classifier only achieves 89.6% accuracy. When looking at the top 20 features for each class by probability mass in the condition distributions, we find that all of the top 20 features are from the body zone and generally none of the features has 1% or more of the probability mass. Appendix §7.1 shows the output of these two datasets.

8 Feature Selection Methods

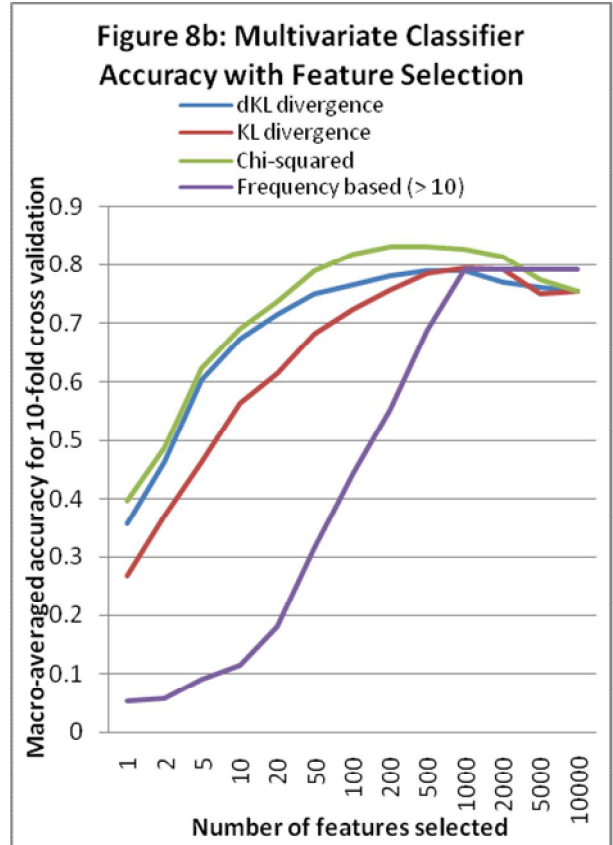
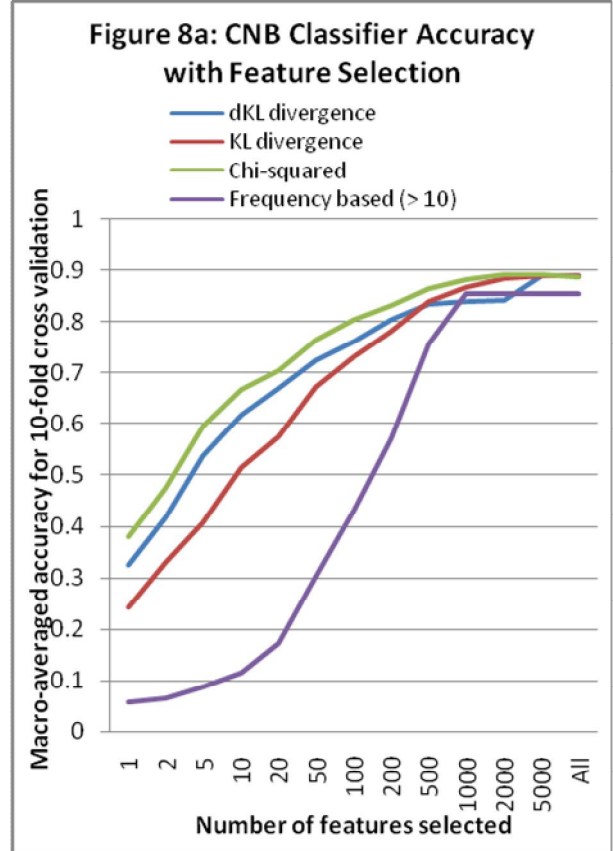
We implemented 4 feature selection methods.

1. Chi-squared feature selection (Section 3)
2. KL divergence feature selection [2]
3. dKL divergence feature selection [2]
4. Frequency-based feature selection (for baseline purposes)

The comparison of the 4 different feature selection methods for our CNB classifier is as shown in Figure 8a.

We implemented our own frequency-based feature selection classifier for comparison purposes. The selection method removes features with a frequency of 10 or lower and then finds the median of frequencies of the words. After determining the median, the method returns K features from the feature set such that the median is maintained. For example, if 3 features are to be returned, the algorithm returns features with frequencies - median-1, median, median+1.

As shown in the graph, the chi-square classification worked better than the KL and dKL divergence methods. Since the chi-square feature selection method is a class-based scoring method, it returns the features that are most differentiating for each of the classes, thus modeling the classes in a better way. KL-divergence on the other hand overlooks the class-based score. Thus, even though the words returned are differentiating as a whole, they generally belong to a subset of classes. Figure 8b shows the curves for our multivariate classifier, showing the peak at around 500 features.



9 Other Classifiers

Using WEKA, we compared our classifiers with SVMs and decision trees. Since SVMs are extremely slow compared to Naïve Bayes classifiers, we limited our feature set to the top 10 words for each class as determined by the Chi-Square method. We used the original dataset generated by the unmodified Message-Parser code and 10-fold cross validation. Due to overlap in the top 10 lists, the Chi-Square method resulted in 193 different features. Results are presented in Table 9.

The LibSVM RBF classifier is an SVM with an RBF kernel with C set to 2048 and gamma set to 2^{-11} . Similarly, LibSVM Linear uses a linear kernel with $C = 8192$. For the RBF kernel, we followed the LibSVM Guide ([3]) for determining the parameters C and gamma, which suggests trying all C and gamma pairs with $C = 2^{-5}, 2^{-3}, \dots, 2^{15}$ and gamma = $2^{-15}, 2^{-13}, \dots, 2^3$. For the linear kernel we only used the C range. We did not try other kernels because the additional parameters were prohibitive due to our time constraints. We trained each classifier using the raw feature counts, which had the best result, as well as normalizing the counts by document length. The results presented in Table 9 are for using the raw counts.

DMNBtext is the Discriminative Multinomial Naïve Bayes classifier as described in [4]. It combines both generative and discriminative methods for parameter estimation. The classifier had no parameters to tune. The C4.5 decision tree classifier used Laplace smoothing for predicting probabilities and the default parameters for everything else.

As Table 9 shows, the SVMs and DMNBtext classifiers performed best. The decision tree was a distant last, and was also the slowest classifier by far. Out of our classifiers, the multivariate performed best. Taking into account the much longer train/test times of the SVMs (1,200 to 2,500 times slower than our classifiers), the slight gain in accuracy from SVMs is not worth the time required to train

and make predictions. Our Naïve Bayes classifiers definitely have a better accuracy to time ratio than the other classifiers that we tested.

Table 9: Classifier Performance

Classifier	Time (seconds)	Macro-averaged Accuracy
LibSVM RBF	377	0.6929
DMNBtext	3	0.6926
LibSVM Linear	748	0.6914
Multivariate NB (ours)	0.3	0.6901
Multinomial NB (ours)	03	0.6831
CNB (ours)	0.3	0.6667
WCNB (ours)	0.3	0.6613
TWCNB (ours)	0.6	0.6588
C4.5 Decision Tree	2,486	0.6573

10 Conclusion

In this project we developed many variants of the Naïve Bayes classifier and applied these classifiers to the newsgroup classification. Our best performing classifier was CNB with a TF-IDF transformation. We tested several feature selection methods and found that Chi-Square performed best, improving the accuracy of the our multivariate classifier significantly. Other classifiers such as SVMs were shown to perform marginally better than our classifiers, but they took substantially longer for training/testing and could not handle as many features. Our most important extra feature/preprocessing step was to add the words in the From header into the document's subject zone. This enabled our classifiers to achieve 99.98% accuracy for four different datasets. Overall, Naïve Bayes classifiers have the best performance to time ratio on text classification than all other examined classifiers.

11 References

- [1] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger, Tackling the poor assumptions of Naïve Bayes Text Classifier, In ICML 2003.
- [2] Karl-Michael Schneider, A New Feature Selection Score for Multinomial Naïve Bayes Text Classification Based on KL-Divergence.
- [3] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, A Practical Guide to Support Vector Classification, <http://www.csie.ntu.edu.tw/~cjlin>
- [4] Jiang Su, Harry Zhang, Charles X. Ling, Stan Matwin, Discriminative Parameter Learning for Bayesian Networks, In ICML 2008.

Appendix §3: Spelling Correction Parameters
alt.atheism
atheist, atheism, rushdi, islam, livesei, moral, benedikt, solntz, livesey@solntze.wpd.sgi.com, keith, beauchain, wpd, bobbe@vice.ico.tek.com, jaeger, jaeger@buphy.bu.edu, buphi, mozumd, ico, god, keith@cco.caltech.edu
comp.graphics
graphic, imag, gif, tiff, polygon, pov, jpeg, format, file, viewer, tga, textur, siggraph, raytrac, pcx, cview, program, algorithm, ftp, geometr
comp.os.ms-windows.misc
window, microsoft, cica, file, ini, driver, bmp, directori, louray@seas.gwu.edu, lourai, download, panayiotaki, font, ftp, app, 3, workgroup, zip, gwu, win
comp.sys.ibm.pc.hardware
isa, card, motherboard, scsi, bio, vlb, 486, gatewai, jumper, drive, irq, eisa, floppi, cach, board, cmo, adaptec, disk, nanao, dma
comp.sys.mac.hardware
mac, appl, centri, quadra, iisi, powerbook, lciii, duo, 610, simm, fpu, nubu, iici, monitor, 040, adb, 68030, vram, hade, upgrad
comp.windows.x
motif, widget, xterm, server, xlib, window, openwindow, client, displai, suno, xdm, applic, xview, sparc, pixmap, compil, mwm, olwm, colormap, lib
misc.forsale
sale, ship, forsal, offer, sell, condit, write, obo, articl, cod, price, stereo, manual, mint, 00, cassett, brand, mutant, email, kou
rec.autos
car, ford, auto, mustang, toyota, automot, nissan, engin, dealer, dumbest, callison@uokmax.ecn.uoknor.edu, calli-son, chevi, uokmax, sedan, camaro, tauru, eliot, wagon, tranni
rec.motorcycles
bike, dod, ride, motorcycl, rider, biker, bmw, ama, yamaha, harlei, helmet, wheeli, counterst, egreen@east.sun.com, honda, behanna, egreen, cage, moto, dog
rec.sport.baseball
basebal, pitch, pitcher, hitter, player, game, bat, yanke, brave, team, sox, hit, batter, season, leagu, rbi, ball, jai, phil-li, cub
rec.sport.hockey
hockey, playoff, nhl, team, game, bruin, leaf, player, goali, penguin, season, cup, plai, fan, goal, detroit, score, stan-lei, coach, wing
sci.crypt
clipper, encrypt, kei, crypto, escrow, chip, nsa, wiretap, tap, secur, algorithm, cryptographi, pgp, sternlight, privati, secret, decrypt, strnlght@netcom.com, rsa, strnlght
sci.electronics
circuit, voltag, amp, resistor, electron, signal, amplifi, frequenc, volt, inqmind, puls, wire, batteri, solvent, diod, shack, bison, capacitor, output, ohm
sci.med
medic, doctor, patient, diseas, geb, medicin, diet, treatment, infect, geb@cadre.dsl.pitt.edu, dsl, physician, chastiti, gordon, cadr, diagnos, clinic, intellect, symptom, skeptic
sci.space
space, orbit, shuttl, nasa, launch, moon, spacecraft, mission, henry@zoo.toronto.edu, solar, satellit, lunar, zoo, nsmca, sky, henri, spencer, prb, flight, payload
soc.religion.christian
rutger, atho, christian, god, christ, church, jesu, sin, scriptur, bibl, geneva, cathol, doctrin, 1993, faith, heaven, lord, vers, spiritu, spirit
talk.politics.guns

gun, firearm, atf, waco, ranch, batf, dividian, fbi, weapon, survivor, handgun, cdt, fire, compound, burn, assault, cdt@vos.stratus.com, cdt@rocket.sw.stratus.com, arm, stratu
talk.politics.mideast
israel, isra, arab, turkish, armenia, armenian, turk, muslim, argic, serdar, jew, palestinian, occupi, extermin, sahak, appressian, melkonian, ohanu, 1920, 1919
talk.politics.misc
cramer, optilink, clayton, gai, cramer@optilink.com, homosexu, percentag, consent, molest, kaldi, steveh@thor.isc-br.com, steveh, hendrick, clinton, hallam, mutual, promiscu, dscomsa, democrat, sexual
talk.religion.misc
christian, sandvik, jesu, newton, kent, kendig, royalroad, sandvik@newton.apple.com., ksand, ceccarelli, mlee@post.royalroads.ca, mlee, alink, 9615, brian@lpl.arizona.edu, god, malcolm, bskendig@netcom.com, mor-mon, bskendig

Appendix §7.1: Result of Two Pre-processing/Feature Sets			
99.8% Dataset: Stemming(None), Lowercas-ing(Off), Stop List(Off), From(On), Bi-grams(Off), Noun N-Grams(Off), sbSame(Off)		89.6% Dataset: Stemming(None), Lowercas-ing(Off), Stop List(On), From(On), Bi-grams(On), Noun N-Grams(On), sbSame(Off)	
Feature	Conditional Probability Mass	Feature	Conditional Probability Mass

alt.atheism			
s!-LRB-	0.145484	b!n't	0.002869
s!-RRB-	0.145447	b!writes	0.001447
s!Keith	0.023337	b!people	0.001186
s!Schneider	0.016089	b!article	0.00112
s!keith@cco.caltech.edu	0.016086	b!God	0.001104
s!Allan	0.016086	b!time	5.93E-04
s!Jon	0.013325	b!religion	5.39E-04
s!Livesey	0.013325	b!Jesus	5.04E-04
s!livesey@solntze.wpd.sgi.com	0.013325	b!atheism	4.99E-04
s!Robert	0.009551	b!evidence	4.74E-04
b!-RRB-	0.00926	b!make	4.72E-04
s!kmr4@po.CWRU.edu	0.006993	b!moral	4.62E-04
s!Ryan	0.006993	b!good	4.41E-04
s!I3150101@dbstu1.rz.tu-bs.de	0.006437	b!atheists	4.37E-04
s!Benedikt	0.006437	b!point	4.32E-04
s!Rosenau	0.006437	b!system	4.13E-04
s!Beauchaine	0.006193	b!god	4.09E-04
s!bobbe@vice.ICO.TEK.COM	0.006022	b!argument	3.94E-04
s!mathew@mantis.co.uk	0.005839	b!things	3.81E-04
comp.graphics			
s!mathew	0.005839	b!true	3.74E-04
s!-RRB-	0.165999	b!image	0.00161
s!-LRB-	0.165988	b!n't	0.001271
s!Robert	0.005661	b!JPEG	0.001141
s!David	0.005212	b!file	0.001005
b!-RRB-	0.003641	b!images	8.38E-04
s!Chris	0.003358	b!graphics	7.64E-04
s!Mark	0.002958	b!files	7.20E-04
s!Steve	0.002948	b!format	6.93E-04
s!Michael	0.002912	b!program	6.67E-04
s!John	0.002784	b!color	6.24E-04
b!the	0.002781	b!writes	5.95E-04
s!Peter	0.002655	b!GIF	5.81E-04
s!Allen	0.002622	b!article	5.76E-04
s!ab@nova.cc.purdue.edu	0.002622	b!software	5.71E-04
s!Kenneth	0.00227	b!version	5.56E-04
s!Thomas	0.002148	b!data	5.41E-04

s!zyeh@caspian.usc.edu	0.002101	b!2	4.35E-04
s!yeh	0.002101	b!information	4.31E-04
s!zhenghao	0.002101	b!package	4.23E-04
s!Lee	0.00206	b!Graphics	4.20E-04
comp.os.ms.windows.misc			
s!-RRB-	0.159203	b!MAX	0.007775
s!-LRB-	0.158928	b!7	0.003253
b!-RRB-	0.035012	b!1	0.00261
b!-LRB-	0.009514	b!2	0.002531
s!Michael	0.005158	b!0	0.002404
s!David	0.003928	b!4	0.001804
s!Mike	0.003536	b!6	0.001766
s!Tom	0.003357	b!9	0.001754
s!Thomas	0.003149	b!3	0.001722
s!Dave	0.00309	b!8	0.001695
s!John	0.003031	b!Windows	0.001644
s!Panayiotakis	0.00302	b!5	0.001606
s!louray@seas.gwu.edu	0.00302	b!n't	0.001471
s!Richard	0.002404	b!file	9.34E-04
s!Tony	0.002376	b!writes	8.02E-04
s!Paul	0.002315	b!GIZ	7.37E-04
b!the	0.002258	b!windows	7.27E-04
s!Peter	0.002166	b!BHJ	7.13E-04
s!Daniel	0.00209	b!DOS	6.76E-04
s!Brian	0.001984	b!article	6.36E-04
comp.sys.ibm.pc.hardware			
s!-RRB-	0.170283	b!drive	0.001534
s!-LRB-	0.170283	b!n't	0.00147
s!Wayne	0.004604	b!SCSI	9.83E-04
s!Robert	0.004564	b!card	8.17E-04
s!Gordon	0.004011	b!IDE	7.24E-04
s!Mark	0.003827	b!system	7.02E-04
s!Lang	0.003609	b!writes	6.61E-04
s!glang@slee01.srl.ford.com	0.003609	b!controller	6.46E-04
s!Michael	0.003356	b!2	5.82E-04
b!-RRB-	0.003266	b!drives	5.80E-04
s!Sam	0.003238	b!article	5.77E-04
s!ab245@cleveland.Freenet.Edu	0.003238	b!disk	5.62E-04
s!Latonia	0.003238	b!bus	5.60E-04
s!David	0.003145	b!1	5.06E-04
s!Mike	0.002981	b!hard	4.89E-04
s!bgrubb@dante.nmsu.edu	0.002933	b!problem	4.68E-04
s!GRUBB	0.002933	b!work	4.37E-04
s!John	0.002848	b!-LCB-	4.33E-04
b!the	0.00282	b!ve	3.86E-04
s!Smith	0.002609	b!time	3.80E-04
comp.sys.mac.hardware			
s!-RRB-	0.162555	b!n't	0.001333
s!-LRB-	0.162548	b!Mac	8.74E-04
s!Brian	0.006048	b!Apple	8.48E-04
s!David	0.005924	b!writes	7.13E-04
s!Hughes	0.004403	b!drive	6.46E-04
s!hades@coos.dartmouth.edu	0.004166	b!problem	5.66E-04
s!Jon	0.00337	b!article	5.61E-04
s!Mark	0.00333	b!monitor	4.66E-04
s!Michael	0.003304	b!system	4.04E-04
b!-RRB-	0.003211	b!ve	3.92E-04
s!Robert	0.003118	b!work	3.70E-04
s!Peter	0.002971	b!software	3.61E-04
b!the	0.002864	b!card	3.48E-04
s!Kuo	0.002677	b!power	3.28E-04
s!Guy	0.002677	b!disk	3.26E-04
s!guykuo@carson.u.washington.edu	0.002677	b!problems	3.18E-04
s!Thomas	0.002531	b!speed	3.07E-04

s!Eric	0.002391	b!time	3.05E-04
s!d88-jwa@hemul.nada.kth.se	0.002359	b!2	3.04E-04
s!Adams	0.001988	b!SCSI	2.98E-04
comp.windows.x			
s!-RRB-	0.167692	b!n't	0.001286
s!-LRB-	0.167689	b!window	0.001211
s!David	0.007297	b!file	0.001026
b!-RRB-	0.005551	b!-RCB-	9.15E-04
s!Andre	0.004512	b!DOS	8.81E-04
s!Beck	0.004315	b!-LCB-	8.73E-04
s!beck@irzr17.inf.tu-dresden.de	0.00431	b!program	8.17E-04
b!-LRB-	0.003741	b!0	6.99E-04
s!Michael	0.003563	b!server	6.93E-04
s!Brian	0.003532	b!entry	6.06E-04
s!Robert	0.003493	b!1	5.90E-04
b!the	0.003488	b!Motif	5.79E-04
s!Richard	0.003059	b!application	5.61E-04
s!Lee	0.003026	b!writes	5.46E-04
s!Ken	0.002952	b!set	5.44E-04
s!John	0.002689	b!output	5.39E-04
s!Patrick	0.002508	b!problem	5.12E-04
s!Mark	0.002437	b!code	4.87E-04
s!Mike	0.002336	b!running	4.66E-04
s!Tom	0.002327	b!widget	4.65E-04
misc.forsale			
s!-RRB-	0.169081	b!1	0.002188
s!-LRB-	0.169075	b!2	0.0014
s!David	0.005678	b!3	7.61E-04
s!John	0.00365	b!sale	5.00E-04
s!Michael	0.003645	b!offer	4.80E-04
s!Chen	0.00289	b!n't	4.71E-04
s!Robert	0.002852	b!4	4.67E-04
s!Wilson	0.002771	b!5	4.60E-04
s!Mike	0.002617	b!10	4.36E-04
s!Mark	0.002604	b!shipping	4.36E-04
s!Jeff	0.002601	s!sale	4.01E-04
s!DOUGLAS	0.002526	b!interested	3.71E-04
s!KOU	0.002526	b!drive	3.66E-04
s!Peter	0.002349	b!condition	3.64E-04
s!Eric	0.002245	b!sell	3.43E-04
s!Dave	0.002191	b!DOS	3.40E-04
s!Samuel	0.002142	b!6	3.16E-04
s!02106@ravel.udel.edu	0.002142	b!price	3.16E-04
s!Ross	0.002142	b!email	2.97E-04
s!Brian	0.002121	b!appears	2.75E-04
rec.autos			
s!-LRB-	0.169241	b!car	0.002057
s!-RRB-	0.16924	b!n't	0.002039
s!John	0.005729	b!writes	0.001293
s!Andrew	0.005218	b!article	0.001168
b!-RRB-	0.005146	b!cars	8.41E-04
s!Craig	0.00465	b!engine	5.77E-04
s!Boyle	0.004302	b!good	5.64E-04
s!Robert	0.003989	b!ve	4.36E-04
s!boyle@cactus.org	0.003966	b!people	3.94E-04
s!Mark	0.003789	b!time	3.83E-04
s!James	0.003446	b!Ford	3.45E-04
s!Jim	0.003428	b!back	3.24E-04
s!Chen	0.003397	b!make	3.18E-04
b!the	0.00327	b!dealer	3.09E-04
s!cka52397@uxa.cso.uiuc.edu	0.003183	b!7	3.09E-04
s!uiuc	0.003183	b!price	3.00E-04
s!Matthew	0.003097	b!problem	2.98E-04
s!eliot	0.003076	b!oil	2.95E-04
s!Tom	0.002816	b!drive	2.95E-04

s!callison@uokmax.ecn.uoknor.edu	0.002686	b!speed	2.80E-04
rec.motorcycles			
s!-RRB-	0.170501	b!n't	0.001681
s!-LRB-	0.170498	b!writes	0.00145
s!Michael	0.006259	b!article	0.001274
s!Chris	0.006184	b!bike	0.001239
s!Woodward	0.005341	b!DoD	9.08E-04
s!Mike	0.00507	b!'ve	5.32E-04
s!John	0.004961	b!ride	4.33E-04
s!David	0.004935	b!back	3.87E-04
s!BeHanna	0.004475	b!good	3.76E-04
s!Pixel	0.004282	b!time	3.69E-04
s!Cruncher	0.004282	b!re	3.47E-04
s!Green	0.004282	b!BMW	3.39E-04
s!azw@aber.ac.uk	0.004142	b!riding	3.28E-04
s!Andy	0.004142	b!l	3.19E-04
b!-RRB-	0.004074	b!bikes	3.01E-04
s!Dave	0.004072	b!make	2.93E-04
s!Nick	0.003839	b!motorcycle	2.90E-04
s!Pettefar	0.003839	b!thing	2.90E-04
s!npet@bnr.ca	0.003839	b!dog	2.82E-04
s!behanna@syl.nj.nec.com	0.003796	b!front	2.70E-04
rec.sport.baseball			
s!-LRB-	0.165577	b!n't	0.002412
s!-RRB-	0.165571	b!writes	0.001357
s!David	0.01377	b!year	0.001131
s!Michael	0.008758	b!article	0.001111
s!Ted	0.007291	b!game	9.59E-04
s!Edward	0.00522	b!1	8.41E-04
s!The	0.005014	b!team	8.32E-04
b!-RRB-	0.004994	b!0	7.97E-04
s!tedward@cs.cornell.edu	0.004574	b!games	6.91E-04
s!Fischer	0.004574	b!good	6.82E-04
s!Mike	0.004396	b!baseball	6.15E-04
s!John	0.004271	b!time	6.02E-04
s!Smith	0.004078	b!2	5.99E-04
s!Eric	0.003989	b!3	5.81E-04
s!luriem@alleg.edu	0.00392	b!players	5.79E-04
s!Lurie	0.00392	b!hit	5.48E-04
s!Liberalizer	0.00392	b!runs	4.43E-04
s!Mark	0.003859	b!season	4.36E-04
s!Steve	0.003835	b!4	4.05E-04
s!Robert	0.003488	b!years	3.94E-04
rec.sport.hockey			
s!-LRB-	0.160474	b!0	0.005499
s!-RRB-	0.160472	b!1	0.004861
s!golchowy@alchemy.chem.utoronto.ca	0.009507	b!2	0.003388
s!Olchowy	0.009507	b!n't	0.002028
s!Gerald	0.009507	b!3	0.002015
s!Deepak	0.005372	b!4	0.001706
s!Chhabra	0.005372	b!game	0.001481
s!dchhabra@stpl.ists.ca	0.005372	b!team	0.001191
b!-RRB-	0.005168	b!5	0.001169
s!Roger	0.005155	b!6	0.001028
s!Maynard	0.005152	b!7	9.74E-04
s!maynard@ramsey.cs.laurentian.ca	0.005151	b!writes	9.67E-04
s!Keith	0.005045	b!hockey	8.87E-04
s!Gary	0.005043	b!25	8.72E-04
s!Keller	0.004426	b!play	8.03E-04
s!kkeller@mail.sas.upenn.edu	0.004426	b!games	7.52E-04
s!gld@cunibx.cc.columbia.edu	0.004299	b!article	7.18E-04
s!Dare	0.004299	b!year	6.39E-04

b!the	0.004038	b!NHL	6.11E-04
s!Scott	0.003874	b!season	5.99E-04
sci.crypt			
s!-LRB-	0.163125	b!n't	0.002229
s!-RRB-	0.163123	b!key	0.00171
s!David	0.008049	b!encryption	0.001242
s!John	0.006797	b!writes	0.001136
s!Steve	0.006367	b!government	0.001074
s!Graham	0.006345	b!article	8.95E-04
s!Toal	0.006345	b!people	8.84E-04
s!Sternlight	0.00634	b!system	8.18E-04
s!strnlght@netcom.com	0.00634	b!chip	8.04E-04
s!gtoal@gtoal.com	0.006034	b!keys	7.74E-04
b!-RRB-	0.005138	b!Clipper	7.43E-04
b!the	0.005098	b!security	6.26E-04
s!Marc	0.004408	b!information	6.07E-04
s!Walker	0.004286	b!law	5.69E-04
s!Amanda	0.004286	b!privacy	5.66E-04
s!amanda@intercon.com	0.004286	b!NSA	5.41E-04
s!Vesselin	0.004251	b!phone	5.35E-04
s!bontchev@fbihh.informatik.uni-hamburg.de	0.004251	b!data	5.10E-04
s!Bontchev	0.004251	b!DES	5.05E-04
s!Carl	0.004151	b!number	5.05E-04
sci.electronics			
s!-RRB-	0.168152	b!n't	0.00147
s!-LRB-	0.168151	b!writes	9.34E-04
s!John	0.012091	b!article	7.53E-04
s!David	0.007496	b!power	4.40E-04
s!Mark	0.007016	b!good	4.36E-04
s!Dave	0.004581	b!ve	3.97E-04
s!Bill	0.00458	b!time	3.97E-04
s!Robert	0.004444	b!work	3.95E-04
b!-RRB-	0.004077	b!circuit	3.69E-04
s!dtmedin@catbyte.b30.ingr.com	0.003908	b!ground	3.42E-04
s!Medin	0.003908	b!make	3.33E-04
s!Scott	0.003788	b!2	3.31E-04
s!Michael	0.003363	b!wire	3.24E-04
s!Chris	0.00327	b!copy	3.01E-04
s!Mike	0.003261	b!1	2.88E-04
s!Aaron	0.003167	b!re	2.84E-04
b!the	0.003052	b!battery	2.72E-04
s!Lung	0.002955	b!current	2.70E-04
s!alung@megatest.com	0.002955	b!problem	2.68E-04
s!wtm@uhura.neoucom.edu	0.002831	b!output	2.63E-04
sci.med			
s!-LRB-	0.165111	b!n't	0.001862
s!-RRB-	0.165106	b!writes	0.001037
s!Gordon	0.021341	b!article	0.001032
s!Banks	0.02121	b!people	7.25E-04
s!geb@cs.pitt.edu	0.02121	b!time	5.32E-04
s!David	0.008647	b!patients	5.08E-04
s!Steve	0.006119	b!disease	5.06E-04
s!Mark	0.00545	b!medical	4.51E-04
s!Jim	0.004766	b!MSG	4.49E-04
s!Robert	0.00475	b!ve	4.48E-04
b!-RRB-	0.004383	b!good	4.30E-04
s!Kenneth	0.00391	b!years	4.07E-04
s!Michael	0.003768	b!doctor	3.95E-04
b!the	0.003717	b!information	3.81E-04
s!dyer@spdcc.com	0.003625	b!food	3.69E-04
s!Dyer	0.003625	b!treatment	3.69E-04
s!Ken	0.003252	b!cancer	3.36E-04
s!Gilbert	0.003248	b!1993	3.23E-04
s!Stephen	0.002845	b!health	3.20E-04

s!Zisfein	0.002838	b!6	3.05E-04
sci.space			
s!-LRB-	0.162297	b!n't	0.00163
s!-RRB-	0.162134	b!space	0.001232
s!Pat	0.011404	b!writes	0.001178
s!prb@access.digex.com	0.010114	b!article	9.15E-04
s!Spencer	0.010073	b!Space	7.88E-04
s!Henry	0.010073	b!NASA	6.89E-04
s!henry@zoo.toronto.edu	0.010071	b!launch	5.26E-04
s!nsmca@aurora.alaska.edu	0.008021	b!orbit	5.20E-04
s!Jon	0.006224	b!Earth	4.92E-04
s!Leech	0.006224	b!time	4.77E-04
s!leech@cs.unc.edu	0.006224	b!1	4.66E-04
s!Ron	0.005813	b!people	4.59E-04
s!Baalke	0.005813	b!system	4.57E-04
s!baalke@kelvin.jpl.nasa.gov	0.005813	b!spacecraft	3.77E-04
s!David	0.005267	b!2	3.77E-04
s!mccall	0.004988	b!years	3.77E-04
s!mccall@mksol.dseg.ti.com	0.004988	b!data	3.77E-04
s!fred	0.004988	b!mission	3.67E-04
s!Doug	0.004906	b!satellite	3.57E-04
b!the	0.004762	b!make	3.45E-04
soc.religion.christian			
s!-LRB-	0.161067	b!God	0.003385
s!-RRB-	0.161064	b!n't	0.002453
s!Michael	0.008655	b!people	0.001441
b!the	0.006448	b!Jesus	0.001243
s!Paul	0.005583	b!writes	9.97E-04
s!Andrew	0.005322	b!Christ	9.88E-04
s!jodfishe@silver.ucs.indiana.edu	0.005097	b!article	7.94E-04
s!fisher	0.005097	b!time	7.91E-04
s!joseph	0.005097	b!Christians	7.89E-04
s!dale	0.005097	b!Christian	7.54E-04
s!Covington	0.004511	b!Bible	7.39E-04
s!Mark	0.004294	b!sin	7.23E-04
b!-RRB-	0.004274	b!faith	6.71E-04
s!Chuck	0.004246	b!church	6.24E-04
s!mcovingt@aisun3.ai.uga.edu	0.004205	b!Paul	6.22E-04
s!David	0.004164	b!life	6.00E-04
s!JEK@cu.nih.gov	0.004153	b!things	5.59E-04
s!Kulikauskas	0.003576	b!question	5.43E-04
s!Jayne	0.003245	b!Church	5.38E-04
s!jayne@mmalt.guild.org	0.003245	b!love	5.35E-04
talk.politics.guns			
s!-LRB-	0.157547	b!n't	0.002705
s!-RRB-	0.157547	b!people	0.001444
s!David	0.01023	b!writes	0.001326
s!Tavares	0.010146	b!gun	0.001259
s!cdt@sw.stratus.com	0.010146	b!article	0.001229
b!-RRB-	0.008199	b!guns	7.70E-04
s!Jim	0.007901	b!FBI	7.50E-04
s!Veal	0.006203	b!fire	7.04E-04
s!John	0.006032	b!government	6.55E-04
b!the	0.005911	b!weapons	5.40E-04
s!Frank	0.005449	b!time	5.40E-04
s!Arras	0.005276	b!BATF	4.79E-04
s!jmd@cube.handheld.com	0.005276	b!children	4.39E-04
s!Crary	0.00514	b!File	4.37E-04
s!fcrary@ucsu.Colorado.EDU	0.00514	b!make	4.32E-04
s!Jason	0.004801	b!re	4.07E-04
s!Steve	0.004671	b!law	4.07E-04
s!Dan	0.004562	b!Koresh	4.04E-04
s!Andy	0.004143	b!firearms	4.04E-04
s!Freeman	0.004142	b!compound	3.78E-04

talk.politics.mideast			
s!-RRB-	0.147544	b!n't	0.002535
s!-LRB-	0.147544	b!people	0.001886
s!Serdar	0.020641	b!Israel	0.001665
s!Argic	0.02064	b!Armenian	0.001379
s!sera@zuma.UUCP	0.020637	b!writes	0.001307
b!the	0.00872	b!Turkish	0.00121
s!David	0.008337	b!article	0.001194
s!dbd@urartu.sdpa.org	0.008034	b!Armenians	0.00118
s!Davidian	0.008034	b!Jews	0.001059
b!-RRB-	0.007762	b!Israeli	9.17E-04
s!Jake	0.007167	b!Arab	6.62E-04
s!jake@bony1.bony.com	0.007167	b!Armenia	6.62E-04
s!Livni	0.007167	b!Jewish	6.60E-04
s!Adam	0.006944	b!Turks	6.37E-04
s!clock@orion.oac.uci.edu	0.006372	b!time	6.35E-04
s!Tim	0.006372	b!Turkey	5.71E-04
s!Clock	0.006372	b!killed	5.57E-04
s!for	0.005553	b!years	5.43E-04
s!Center	0.005539	b!government	5.29E-04
s!Policy	0.005538	b!war	5.07E-04
talk.politics.misc			
s!-LRB-	0.148869	b!n't	0.002927
s!-RRB-	0.148848	b!people	0.00161
b!-RRB-	0.009953	b!writes	0.001416
s!Clayton	0.009316	b!article	0.001307
s!Cramer	0.009316	b!President	9.15E-04
s!cramer@optilink.COM	0.009312	b!government	8.97E-04
b!the	0.007975	b!re	7.74E-04
s!Clinton-HQ@Campaign92.Org	0.006156	b!STEPHANOPOULOS	7.39E-04
s!92	0.005765	b!make	6.22E-04
s!David	0.005574	b!time	5.70E-04
s!Mark	0.005509	b!'ve	5.07E-04
s!Steve	0.005079	b!MYERS	4.96E-04
s!Smith	0.00412	b!I	4.96E-04
s!steveh@thor.isc-br.com	0.004041	b!made	4.23E-04
s!Hendricks	0.004041	b!good	3.85E-04
s!Gary	0.00382	b!American	3.61E-04
s!Broward	0.003602	b!health	3.57E-04
s!Horne	0.003602	b!men	3.56E-04
s!Michael	0.003535	b!money	3.52E-04
s!William	0.003445	b!Clayton	3.44E-04
talk.religion.misc			
s!-RRB-	0.138914	b!n't	0.001823
s!-LRB-	0.138914	b!writes	0.001034
b!-RRB-	0.009428	b!God	9.36E-04
s!Brian	0.008971	b!Jesus	9.12E-04
s!Paul	0.008615	b!people	8.74E-04
b!the	0.00797	b!article	8.67E-04
s!David	0.007962	b!Christian	5.06E-04
s!sandvik@newton.apple.com	0.007828	b!Bible	4.43E-04
s!Sandvik	0.007828	b!good	4.22E-04
s!Kent	0.007828	b!life	3.88E-04
s!pharvey@quack.kfu.com	0.005888	b!time	3.61E-04
s!Harvey	0.005888	b!Christians	3.50E-04
s!Robert	0.005354	b!point	3.46E-04
s!Lee	0.004912	b!objective	3.26E-04
s!Malcolm	0.004912	b!'ve	3.19E-04
s!mlee@post.RoyalRoads.ca	0.004912	b!make	3.13E-04
s!Ceccarelli	0.004629	b!Christ	3.04E-04
s!brian@lpl.arizona.edu	0.004629	b!Koresh	2.99E-04
s!Weiss	0.004264	b!religion	2.93E-04
s!Kendig	0.004234	b!world	2.86E-04