Homework 2: Text Classification

Chen Huang
chhuang@umich.edu

March 15, 2009

1 Description of the Classifier

I implemented a Navie Bayes Classifier, with Chi-Square feature selection. Unlikely linear percepton model, NB is suitable for multi-class classification as we don't have to consider the case when multiple hyperplane don't divided the dataset into disjoint regions. All the details could be found from the IR book. When extract the top m features specified by the user, I first computed the top k features for each class and then combine the k features into m features.

While implementing NB, I used TFIDF as document vector's value, and used Laplace smoothing.

One thing needs to notice is that when compute the Chi-square for all the terms, there are cases when a word only appear excludsively in one class of documents, thus leads to computation exceptions. I did a laplace smoothing also on the Chi-square computation.

Also, when computing accuracy using NB, documents with no features found are excluded from the computation.

2 Comparsion between SVM and NB

2.1 Accuracy Comparsion

Please see the table below for a comparsion.

Training Size / Features#	10	100	1000	10000	100000	All
20	39.914	57.2473	69.0323	71.957	71.6559	71.6559
100	40.5591	65.7204	79.7419	83.957	83.4839	83.4839
200	41.5054	67.3548	82.9247	87.5699	88.043	88.043
300	41.8065	66.0215	84.8172	89.2473	89.4624	89.5054
500	41.5484	69.1183	85.2043	89.8925	90.6667	90.6667

Naive Bayes Performance

Training Size / Features#	10	100	1000	10000	100000	All
20	44.05	61.12	67.96	70.75	71.35	71.35
100	42.88	54.22	78.11	82.45	82.32	82.32
200	43.44	63.57	79.60	84.56	85.51	85.51
300	43.44	63.87	82.24	85.29	86.62	86.62
500	43.18	67.1	82.45	87.44	88.3	88.3

SVM Performance

Observations:
Using a parameter c=1 for SVM, NB and SVM performs very close to each other. SVM works better when the number of feaures and the training set size is small, while NB outperform SVM when the training set is large and features are abundant. I think this might because I have not tune SVM parameters and used all default settings, and this could also because the training set I used favor more to NB, A multi-fold cross validation

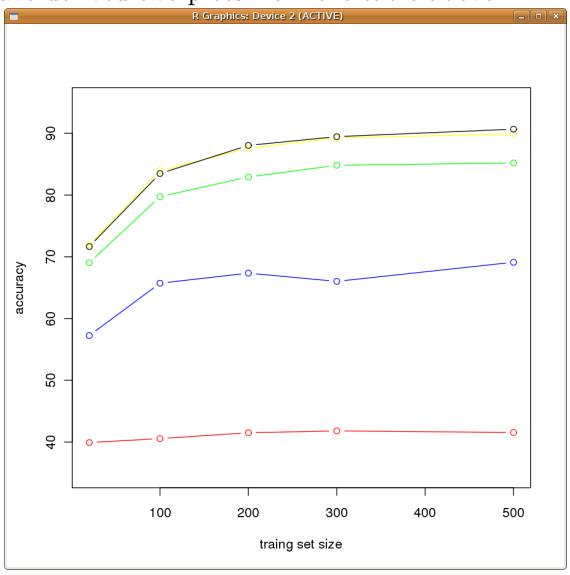
might be able to give a more accurate comparsion. But NB's performance is really good, I guess simply sometimes really is the best.

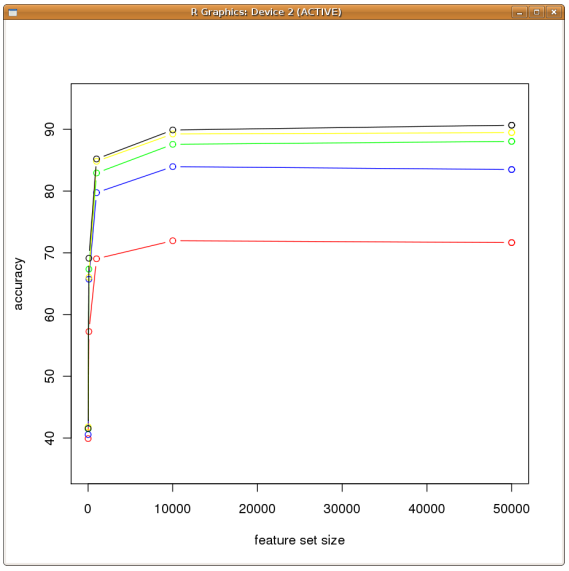
2.2 Performance comparsion

Theoratically, Multi-class SVM's time complexity should be much higher than that of Naive Bayes. Based on the comparsion table on Page 329 of the IR book, NB runs much faster in training phase, and the two are almost run the same in testing phase. This also reflected in this assignment. In order to do a fair comparsion, I used C++ in implementing NB. And NB's learning code runs obviously faster than SVM light, training on 4000+ documents with all words included as features takes less than 2 seconds.

3 Effects of the number of features and size of training samples

I have derived two plots from the table above:





The above plots describes the table. The left plot describes change of accuracy with traning set size changing but the number of features fixed. The right plot describes the chagne of accuracy with feature set size changing but training set fixed. From the above plot, we can see that:
When the feature set's size is as small as 10, change of traning set size won't improve the accuracy singanificantly. (red line of the left plot). The reason might be because 10 features are not representative enough in distinguishing certain document classes that are more similar than other documents, 10 features's likelihood are almost the same in similar classes. Accuracy increases to a significant 70%+ when the features set size increase to 1000. Also increasing feature set size from 1000 to 10000 only increases accuracy by 5%. So 1000 features should be able to nicely cover essential information in the corpus, that's an average of 100 keywords for each class. Also we can see that increasing training set size will increase accuracy by 10% when feature set size is 1000, and 20% when the feature set size is 10000. So the conclusion is that training set size do effect the accuracy, only when the feature set size is larger than 1000. Also needs to notices that change of feature size from 300 to 500 only increase accuracy by less than 5%. So 300 documents are enough to extract all necessary features.

From the right plot we can observe that change feature set size from 10000 to all almost won't increase accuracy and will even do harm when training set size is small. This probably is because of include all words in training will overfit the model. For best NB model, I would choose feature set size to be at around 20000, and the training set size to be as large as possible.

4 Smoothing methods

In this assignment, I used laplace smoothing method. The reason of using Laplace smoothing method is that it's easy to implement and proved to be effective. Other possible smoothing methods are :Good-Turing estimate, Jelinek-Mercer smoothing, Katz smoothing, Witten-Bell smoothing, Absolute discounting, Kneser-Ney smoothing, they are introduced in Stanley F. Chen and Joshua Goodman's paper "An Empirical Study of Smoothing Techniques for Language Modeling".

5 References

Introduction to Information Retrieval, Chris Manning and etc.