CSE 250B: HW2

WEIWEI LI
PID: A53107958

1. In this assignment, I will develop my own ideas for selecting a small subset of the vocabulary for multinomial naive Bayes. The basic ideas for vocabulary selection are as follows:

TF-IDF: First, we discard a few dozen "stopwords", which leads to a new vocabulary with size N. Then we select a smaller subset with size M from the new vocabulary by using the term frequency-inverse document frequency (TF-IDF) algorithm. TF-IDF is a numerical statistic that attempts to quantify the importance of a particular word to a document class. We compute TF-IDF for each word in each class, and for each word, we choose the highest TF-IDF from the twenty class. Then we get N TF-IDF. Next, we choose words having the largest M TF-IDF from N TF-IDF. Therefore, we get M words, which forms the subset of the vocabulary.

2. The pseudocode of the proposed algorithm is as follows:

Input: X: full vocabulary, Trn: training set, Lrn: training label, M: subset size
Output: P: subset of the full vocabulary, Pjw: probability of each word in each class from new training set
TF-IDF
X = full vocabulary
Y= a subset of X by discarding stopwords with size N
tfidf-matrix= 20 by N matrix of TF-IDF using TF-IDF algorithm
tfidf-max= the max TF-IDF from each column of tfidf-matrix
tfidf-sorted = the largest M TF-IDF from tfidf-max
P = words having TF-IDF in tfidf-sorted
Trn-new = new training set T with words from P
Pjw= probability using Multinomial Naive Bayes model on Trn-new.

After choosing the subset of the full vocabulary by TF-IDF algorithm, we ignore the remaining words. Also, for the vocabulary of the test data set, we choose the same subset, and ignore the remaining words. Then we use the new train and new test data set to do Multinomial Naive Bayes classification.

Date: October 25, 2016.

3. We test the performance of this model on the 20 Newsgroups dataset. And we have the size of the full vocabulary 61188. To ensure that certain documents do not get misclassified, we employ Laplace smoothing with $\alpha = 1$. The experiment results are as shown in Table 1.

Classifier of The multinomial Naive Bayes model:
$h(x) = argmax_j log\pi_j + \sum_{i=1}^{|V|} x_i log p_{ji}$

$p_{jw} = \frac{the\,number\,of\,words\,w\,in\,j\,class + 1*\alpha}{the\,number\,of\,all\,words\,in\,j\,class + |V|*\alpha}$

Confidence Interval = $(\mu - z * \frac{\sigma}{\sqrt{n}}, \mu - z * \frac{\sigma}{\sqrt{n}})$

Test error = $\frac{\#\,of\,classified\,wrong}{\#\,of\,classified\,correctly + \#of\,classified\,wrong}$

I used the formula above to calculate the 95% confidence interval and the test error. The random selection of the vocabulary are done uniformly random, and is repeated 50 times for each M value.

From Table 1, we can see that the error rate reduces about 2% from full vocabulary after discarding "stopwords". Comparing the classification error on TF-IDF approach and Full vocabulary, TF-IDF improves performance when M is greater than or equal to 10000. When M equals 20000, the error rate of TF-IDF decreases about 2% from full vocabulary, and it is almost as small as the error rate of stopwords reduce method. And comparing with random selection, TF-IDF approach is doing much better.

M	5000	10000	20000
TF-IDF	0.2277	0.2123	0.1993
Random	0.5559 ± 0.0045	0.4401 ± 0.0042	0.3454 ± 0.0029
Full vocabulary	0.2189		
Stopwords reduce	0.1992		

TABLE 1. Error rate of the Multinomial Naive Bayes classification using M subset of the vocabulary with different selection scheme.

4. For each class, we choose the words which have the largest 20 TF-IDF. The selected words make sense to me. We find the words to be intuitively far more indicative of a document class. The "representatives words" from each class are shown below.

Class 1=['people', 'com', 'atheism', 'writes', 'edu', 'one', 'article', 'god', 'would', 'atheists', 'islam', 'think', 'religion', 'say', 'livesey', 'like', 'morality', 'know', 'atheist', 'jesus']

Class 2=['files', 'images', 'image', 'also', 'edu', 'writes', 'software', 'would', 'file', 'graphics', 'use', 'jpeg', 'ftp', 'gif', 'one', 'data', 'com', 'format', 'know', 'like']

Class 3=['use', 'files', 'com', 'card', 'know', 'problem', 'win', 'dos', 'ei', 'file', 'like', 'one', 'windows', 'article', 'mouse', 'using', 'edu', 'writes', 'get', 'would']

Class 4=['edu', 'one', 'bios', 'scsi', 'controller', 'ide', 'card', 'drive', 'com', 'mb', 'dos', 'disk', 'drives', 'system', 'bus', 'would', 'get', 'pc', 'use', 'floppy']

Class 5=['apple', 'edu', 'mac', 'scsi', 'one', 'quadra', 'mb', 'simms', 'would', 'drive', 'nubus', 'writes', 'know', 'fpu', 'mhz', 'problem', 'centris', 'duo', 'get', 'use']

Class 6=['edu', 'widget', 'com', 'window', 'motif', 'file', 'xterm', 'use', 'program', 'server', 'get', 'output', 'lib', 'contrib', 'oname', 'entry', 'xlib', 'eof', 'one', 'printf']

Class 7=['price', 'like', 'shipping', 'new', 'sale', 'edu', 'mail', 'interested', 'good', 'asking', 'one', 'offer', 'please', 'dos', 'used', 'condition', 'com', 'wolverine', 'email', 'obo']

Class 8=['com', 'car', 'cars', 'would', 'writes', 'article', 'edu', 'one', 'get', 'like', 'good', 'engine', 'callison', 'much', 'think', 'autos', 'dealer', 'know', 'also', 'new']

Class 9=['motorcycle', 'dod', 'bikes', 'like', 'writes', 'edu', 'bike', 'com', 'ride', 'article', 'one', 'helmet', 'rider', 'would', 'get', 'bmw', 'behanna', 'riding', 'know', 'apr']

Class 10=['pitching', 'year', 'writes', 'baseball', 'edu', 'article', 'would', 'team', 'one', 'alomar', 'cubs', 'hitter', 'last', 'game', 'good', 'season', 'braves', 'rbi', 'mets', 'think']

Class 11=['team', 'players', 'ca', 'leafs', 'hockey', 'writes', 'playoffs', 'edu', 'game', 'nhl', 'flyers', 'would', 'lemieux', 'pts', 'season', 'play', 'teams', 'go', 'puck', 'islanders']

Class 12=['escrow', 'db', 'key', 'encryption', 'clipper', 'chip', 'would', 'edu', 'privacy', 'rsa', 'crypto', 'ripem', 'encrypted', 'nsa', 'one', 'com', 'government', 'keys', 'use', 'algorithm']

Class 13=['one', 'like', 'would', 'wiring', 'get', 'writes', 'edu', 'use', 'com', 'circuit', 'article', 'ground', 'know', 'amp', 'good', 'power', 'voltage', 'anyone', 'wire', 'used']

Class 14=['msg', 'like', 'candida', 'disease', 'dyer', 'people', 'com', 'would', 'article', 'patients', 'edu', 'writes', 'health', 'one', 'geb', 'also', 'hiv', 'use', 'know', 'medical']

Class 15=['lunar', 'nasa', 'writes', 'shuttle', 'edu', 'article', 'spacecraft', 'one', 'satellite', 'space', 'moon', 'would', 'launch', 'orbit', 'like', 'mars', 'earth', 'com', 'also', 'satellites']

Class 16=['bible', 'church', 'christians', 'god', 'people', 'would', 'christ', 'athos', 'one', 'think', 'jesus', 'edu', 'faith', 'christian', 'christianity', 'clh', 'know', 'believe', 'writes', 'us']

Class 17=['gun', 'would', 'weapons', 'writes', 'edu', 'article', 'right', 'militia', 'one', 'people', 'fbi', 'guns', 'firearms', 'think', 'firearm', 'get', 'com', 'handgun', 'rkba', 'like']

Class 18=['turkish', 'israeli', 'israel', 'jews', 'armenian', 'people', 'armenians', 'armenia', 'would', 'turks', 'arabs', 'article', 'edu', 'writes', 'arab', 'jewish', 'azerbaijan', 'said', 'turkey', 'one']

Class 19=['writes', 'cramer', 'stephanopoulos', 'would', 'one', 'people', 'article', 'edu', 'com', 'think', 'president', 'mr', 'government', 'know', 'optilink', 'going', 'like', 'make', 'get', 'new']

Class 20=['one', 'like', 'know', 'people', 'sandvik', 'morality', 'com', 'christ', 'christians', 'article', 'god', 'think', 'even', 'would', 'edu', 'bible', 'say', 'writes', 'christian', 'jesus']

5. Clearly, comparing to normal selection and full vocabulary, TF-IDF approach is an improvement. The performance of these models can be promoted by tuning parameter α. The classification accuracy will be further improved by increasing the size of the subset of vocabulary. I would like to try calculating the TF-IDF for words in each document, and choose the words with largest TF-IDF or choose the words with median TF-IDF.