Homework 2 - Sparse Generative Model

Lenord Melvix
A53092538

Short Description of Algorithm:

I built a Multinomial Bayesian classifier that classifies a document based on (1). Run time efficiency of the classifier was improved by using Term Frequency of the words in vocabulary to compute probability and each word was given a weight equal to its inverse document frequency

$$prediction = arg\ max\,(j)\ \ \pi j\ \ \Pi\, pji \qquad (1)$$

Term frequency of a word is the frequency of a word in given document category. I took the logarithmic value of this frequency to avoid burstiness.

$$\text{Term Frequency} = log(1 + f)\,. \qquad (2)$$

Further the probabilities of each word was weighted with a ratio of total number of documents to number of documents in which the word occurs. This would prevent common stop words that occur very frequently in text to influence the classifier.

$$IDF\,(word) = \#documents \ / \#documents\ where\ word\ occurs \qquad (3)$$

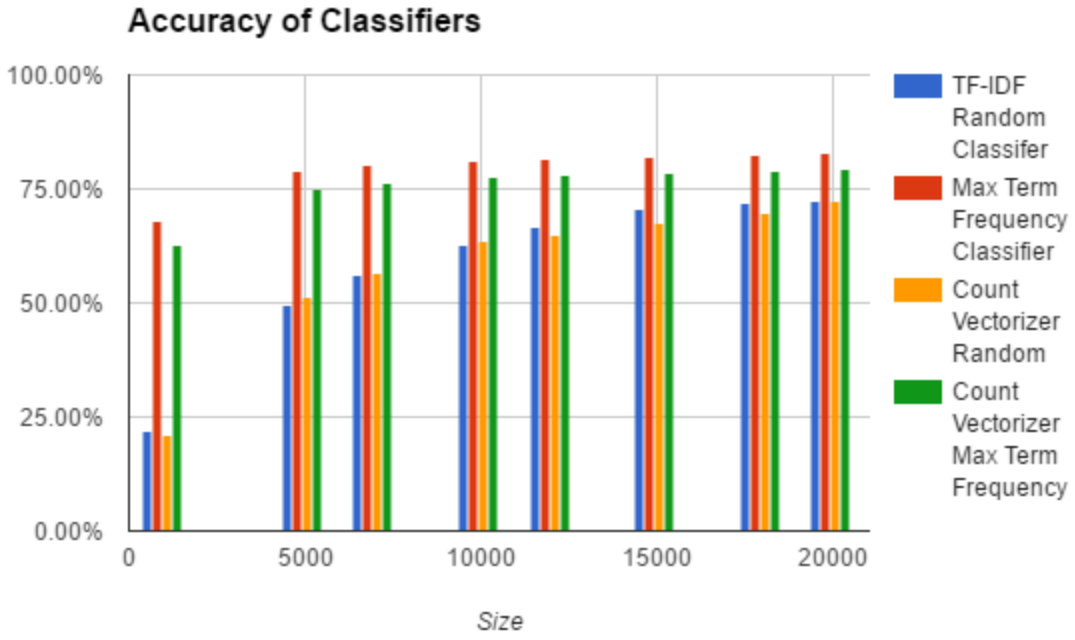
Pseudocode:

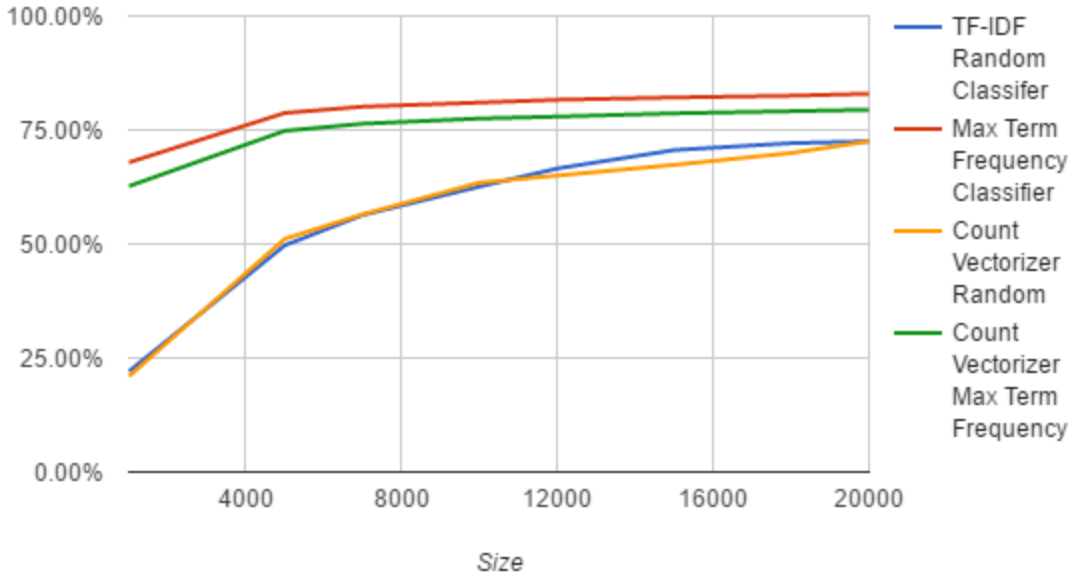
```
function extract_data():
       train_data = extract training data and labels
       test_data = extract test data and labels
       vocabulary = read from vocabulary.txt

function preprocess_features():
       for each word in vocabulary:
              compute term_frequency from equation (2)
              compute its inverse document frequency from (3)
              multiply weights to each term frequency and store in a dictionary
```

```
function vocabulary_builder(input size = M, type = random / tfidf):
        remove stop-words from vocabulary using a pre-determined text
        if type == random :
                pick M words from vocabulary randomly
        else :
                sort dictionary based on tf-idf value and pick top M words
        return picked up dictionary

function test_mnb_classifier():
        for each document in test_data:
                predict label using equation (1)
                 if (all categories have equal probability):   //No word present in test
                     arbitrarily predict a label
                keep track of error in prediction
```

Experimental Results:

Size	Accuracy					
	TF-IDF			Count Vectorizer		
	Random Classifier		Max Term Frequency Classifier	Random Classifier		Max Term Frequency Classifier
	Lower Bound	Upper Bound		Lower Bound	Upper Bound	
1000	22.06	25.24	67.88369623	20.99%	24.23%	62.68%
5000	49.71	53.22	78.73074881	51.13%	56.12%	74.81%
7000	56.35	58.5	80.11152416	56.49%	60.96%	76.37%
10000	62.59	63.5	81.01433882	63.45%	67.93%	77.51%
12000	66.56	67.45	81.62506638	65.00%	69.54%	77.93%
15000	70.59	71.8	82.12958046	67.35%	71.20%	78.65%
18000	72.11	73.12	82.51460435	69.94%	75.77%	79.12%
20000	72.5	72.88	82.93945831	72.52%	78.09%	79.43%
60000	82.83	83	83.29792884	80.07%	83.70%	80.16%


Accuracy of Classifiers
100.00%
75.00%
50.00%
25.00%
0.00%
0
5000
10000
15000
20000
Size
TF-IDF Random Classifer
Max Term Frequency Classifier
Count Vectorizer Random
Count Vectorizer Max Term Frequency


100.00%
75.00%
50.00%
25.00%
0.00%
4000
8000
12000
16000
20000
Size
TF-IDF Random Classifer
Max Term Frequency Classifier
Count Vectorizer Random
Count Vectorizer Max Term Frequency

Representative words for M=1000 :

Group	Max Features
alt.atheism	halat mark held crime couldn prove student sense au christian
comp.graphics	student vector al lee cpu zyeh top tdawson mark au
comp.os.ms-windows.misc	return upload mark fault lee character top communications results au
comp.sys.ibm.pc.hardware	sense couldn al appears top results communications mark au cpu
comp.sys.mac.hardware	couldn return results portable connected mark au top student cpu
comp.windows.x	lee communications character mark tgv appears top au sparc return
misc.forsale	udel al student maryland mark top portable communications camera genesis
rec.autos	prove al hate rare pickup couldn torque night mark top
rec.motorcycles	student curt fault return rtsg mark top night shop nick
rec.sport.baseball	lee udel astros night royals top al yankees mark fan
rec.sport.hockey	return deserve couldn al results mark capitals top night fan
sci.crypt	appears faire top signed trial au couldn crime sense communications
sci.electronics	rootstown student portable relay top communications al connected mark au
sci.med	reduce experimental gilbert couldn night sense communications mark risk results
sci.space	couldn nick maryland night atmosphere mark land astronomy au communications
soc.religion.christian	judged prove au held genesis student appears sense mark christian
talk.politics.guns	mark deserve couldn risk fault held trial al udel crime
talk.politics.mideast	hate return mark republic student beyer extermination sera land israeli
talk.politics.misc	prove reduce al neighbor sense trial romulus crime mark top
talk.religion.misc	tucson halat land held repentance prove mark lee sense christian

a) These subset of features for each category were selected by computing the sorting the (term frequency * inverse document frequency) of the words in vocabulary and top M/20 features for each label are chosen as representatives.

b) Yes, as we can see the word christian appears fairly often in all religion based documents while communications appear in technology related documents. Though these words do not significantly distinguish from one category to another. Such words easily demonstrate which subset of categories would the document belong to and the other specific words learned from the training further add on to the classification.

Critical Evaluation:

This approach clearly beats randomly chosen classifier with an even bigger margin when the words probabilities of word occurrences are considered instead of term frequency. So it is a clear benefit opposed to Naive MultiNomial Bayesian classification.

And can be further improved by taking into account the in-class variance of each feature and overall variance variance across all the categories to identify if those features that can uniquely classify to a category. We could have also implemented Support Vector Machine based classification that promises to be a better approach to this problem statement.