STAT S626 Bayesian Theory and Data Analysis

CLASSIFICATION OF TEXT DOCUMENTS BASED ON A BAYESIAN CLASSIFIER

April 29, 2016

Jayasankar, Siddharth

Kamalapuram Muralidhar, Anirudh

Madhavan, Sarvothaman

Sanakaranarayanan, Arun Ram

Contents

Chapter 1

An Exective Summary

1.1 Objective and problem statement:

Document classification is a problem in library science, information science and computer science The task is to assign a document to one or more categories. This task of classifying documents takes into consideration the content of the documents, that is, by the words of which they are comprised. In this project we try to classify news articles to different news categories. We intend to use the Naive Bayes approach to tackle this problem

1.2 Naive Bayes approach:

This approach uses a simple naive assumption of words present in documents as being independent of each other, that is, the occurrence of one word does not affect the occurrence of other words appearing in the document. Then using the Bayes rule posterior probability of each word occurring in each news category is calculated and this probability is used to classify the documents to different categories.
In this project we have considered two different data models to represent each document.

1. **Bernoulli document model:** a document is represented by a feature vector with binary elements taking value 1 if the corresponding word is present in the document and 0 if the word is not present

2. **Multinomial document model:** a document is represented by a feature vector with integer elements whose value is the frequency of that word in the document

1.3 Why is this task important?:

Classification of documents in this present world is a tedious task even for people who have specialized in linguistics. The two main reasons are

1. A large volume of articles/documents are begin generated in different fields on a daily basis.
2. Lack of knowledge of an individual in different fields. Example: A person who does not have knowledge about phones could end up classifying a document on Apple Corporation under the category fruits rather than classifying it as corporate article

This naive reasoning clearly highlights and expands the variety of applications in which this can be used.

1.4 Applications of text document classification:

1. Spam filtering, differentiate E-mail spam messages from legitimate emails.
2. Email routing, sending an email sent to a general address to a specific address or mailbox depending on topic
3. Language identification, automatically determining the language of a text
4. Genre classification, automatically determining the genre of a text
5. Readability assessment, automatically determining the degree of readability of a text, either to find suitable materials for different age groups or reader types or as part of a larger text simplification system
6. Sentiment analysis, determining the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document.
7. Article triage, selecting articles that are relevant for manual literature curation, for example as is being done as the first step to generate manually curated annotation databases in biology

Chapter 2

Data Description

We have used the famous 20 newsgroup dataset[1] for this analysis. This data set is a collection of 18,846 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. This data was collected by Ken Lang, probably for his Newsweeder: Learning to filter netnews paper. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other while others are highly unrelated. The 20 categories present in this data set are given in the table next page.

Category	Number of Documents
comp.graphics	973
comp.os.ms-windows.misc	985
comp.sys.ibm.pc.hardware	982
comp.sys.mac.hardware	963
comp.windows.x	988
rec.autos	990
rec.motorcycles	996
rec.sport.baseball	994
rec.sport.hockey	999

Category	Number of Documents
sci.crypt	991
sci.electronics	984
sci.med	990
sci.space	987
misc.forsale	975
talk.politics.misc	775
talk.politics.guns	910
talk.politics.mideast	940
talk.religion.misc	628
alt.atheism	799
soc.religion.christian	997

Of these 18846 documents we have used 11314 documents to train the model and get the prior probabilities of words for each category. Then the rest 7532 documents were used to test the performance of the model, that is, the model will then use the prior information which it observed from the train data and use it to predict the categories of these 7532 documents.

Chapter 3

Bayesian Analysis of Text Classification

3.1 PRIOR REPRESENTATION

In our classifier, we use a bag of words representation of the documents. Simply put, this representation ignores the order of words in the document and keeps track of count of each word in each document.

Consider a document D, whose class is given by c. That is this document belongs to a category c among all possible classes C

$c_d \epsilon C$

We know that,

$P(C|D) = \frac{P(D|C)P(C)}{P(D)} \alpha P(D|C)P(C)$

For prior information, the documents are to be classified into the following classes:

comp.graphics	comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware
comp.sys.mac.hardware	comp.windows.x	rec.autos
rec.motorcycles	rec.sport.baseball	rec.sport.hockey
sci.crypt	sci.electronics	sci.med
sci.space	misc.forsale	talk.politics.misc
talk.politics.guns	talk.politics.mideast	talk.religion.misc
alt.atheism	soc.religion.christian	

As noted in the data description, these are the topics the newsgroup data can contain.

Since each topic is equally likely in a newsgroup conversation without looking into the data

we can assume a uniform probability for each topic.
An alternative machine learning approach would be to use prior formed by simple counting of documents that are classified as each class from a set existing documents with known classes.

3.2 Sampling Model

For the sampling model, we consider and evaluate two alternatives. Both models represent documents using feature vectors whose components correspond to word types. If we have a vocabulary V, containing |V| word types
Bernoulli document model: a document is represented by a feature vector with binary elements taking value 1 if the corresponding word is present in the document and 0 if the word is not present.
Multinomial document model: a document is represented by a feature vector with integer elements whose value is the frequency of that word in the document.
By using this approach from the dataset used we obtain the following prior
,

3.3 Posterior Formulation:

3.3.1 Bernoulli Document Model:

In the Bernoulli model a document is represented by a binary vector, which represents a point in the space of words. If we have a vocabulary V containing a set of $|V|$ words, then the t^{th} dimension of a document vector corresponds to word w_t in the vocabulary. Let b_i be the feature vector for the i^{th} document D_i ; then the t^{th} element of b_i , written b_{it} , is either 0 or 1 representing the absence or presence of word w_t in the i^{th} document.

Let P(w_t |C) be the probability of word w_t occurring in a document of class C; the probability of w_t not occurring in a document of this class is given by (1 - P(w_t |C)). If we make the i.i.d. assumption, that the probability of each word occurring in the document is independent of the occurrences of the other words, then we can write the document likelihood P(D_i| C) in terms of the individual word likelihoods

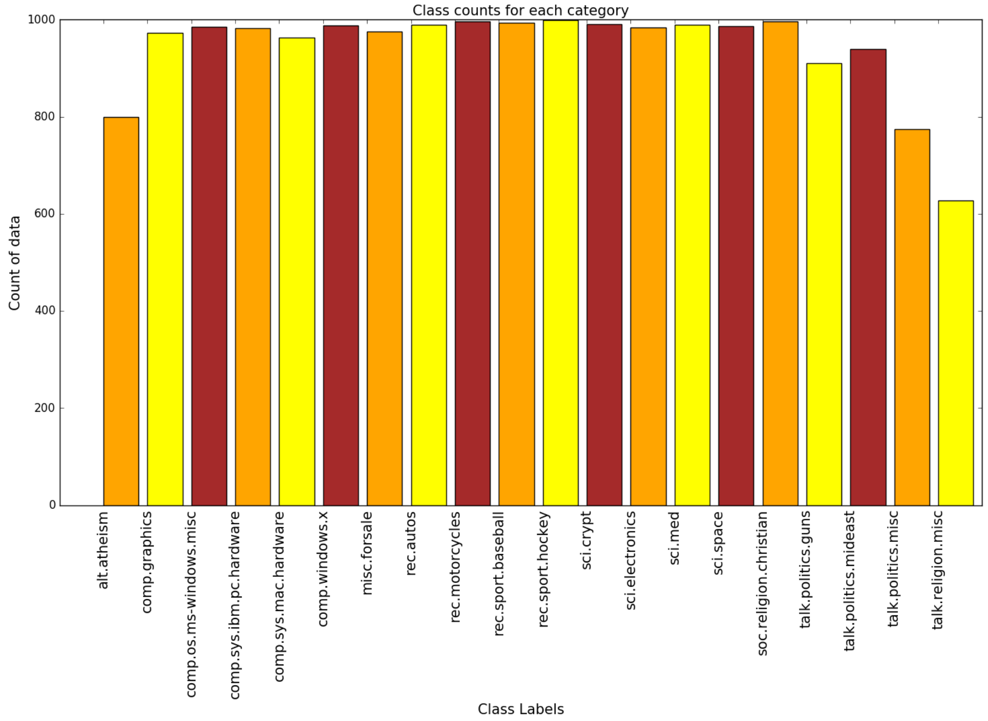

Figure 3.1: Accuracy

$$\mathrm{P}(w_t|\mathrm{C})\text{: } \mathrm{P}(D_i\,|\mathrm{C}) \sim \mathrm{P}(b_{it}|\mathrm{C}) = {}_{t=1}\prod^{|V|} [b_{it}\,\mathrm{P}(w_t\,|\mathrm{C}) + (1 - b_{it}\,)(1 - \mathrm{P}(w_t\,|\mathrm{C}))]$$

3.3.2 Posterior Distribution using Bernoulli sampling:

Here we show the posterior probability applied to our dataset. Since the number of documents is large we show 10 documents and thier posterior probability across 5 words

	$word1$	$word2$	$word3$	$word4$	$word5$
$Doc1$	1	$8.93e-07$	$1.63e-36$	$5.61e-04$	$7.94e-03$
$Doc2$	$1.45e-42$	$4.48e-01$	$9.66e-30$	$5.26e-01$	$2.48e-02$
$Doc3$	$5.29e-21$	$8.56e-02$	$5.18e-32$	$1.01e-03$	$2.15e-01$
$Doc4$	$1.70e-040$	$1.19e-148$	$3.50e-183$	$2.46e-136$	$4.51e-142$
$Doc5$	$3.15e-09$	$9.07e-04$	1	$9.44e-01$	$2.38e-02$
$Doc6$	$8.14e-055$	$4.34e-103$	$1.48e-138$	$7.36e-101$	$5.00e-099$
$Doc7$	$1.82e-54$	1	$5.94e-41$	$1.01e-11$	$4.21e-10$
$Doc8$	$5.07e-47$	$2.87e-04$	$3.11e-29$	$5.92e-12$	$9.94e-01$
$Doc9$	$1.17e-60$	$8.40e-12$	$9.62e-42$	$1.73e-13$	$3.64e-10$
$Doc10$	$3.10e-51$	$9.99e-01$	$6.23e-40$	$1.47e-07$	$9.40e-09$

As we can see, the posterior probability is 0 for most cases and 1 other wise. This is consistent with bernoulli model

3.3.3 Multinomial Document Model:

In the multinomial document model, the document feature vectors capture the frequency of words, not just their presence or absence. Let x_i be the multinomial model feature vector for the i^{th} document D_i. The t^{th} element of x_i , written x_{it} , is the count of the number of times word w_t occurs in document D_i. Let $n_i = \sum_t x_{it}$ be the total number of words in document D_i

Let P(w_t|C) again be the probability of word w_t occurring in class C, this time estimated using the word frequency information from the document feature vectors. We again make the naive Bayes assumption, that the probability of each word occurring in the document is independent of the occurrences of the other words. We can then write the document likelihood P(D_i| C) as a multinomial distribution where the number of draws corresponds to the length of the document, and the proportion of drawing item t is the probability of word type t occurring in a document of class C, P(w_t |C).

$$\mathrm{P}(D_i \mid \mathrm{C}) \sim \mathrm{P}(x_i \mid \mathrm{C}) = \frac{n_i}{{}_{t=1}\prod^{|V|} x_{it}} {}_{t=1}\prod^{|V|} P(w_t|C)$$

$$\sim {}_{t=1}\prod^{|V|} P(w_t|C)$$

3.3.4 Posterior Probability

The posterior probability applied to our sample dataset using multinomial model is below. Since the number of documents is large we show 10 documents and thier posterior probability across 10 words

	$word1$	$word2$	$word3$	$word4$	$word5$	$word6$	$word7$	$word8$	$word9$	$word10$
$Doc1$	0.028	0.037	0.029	0.053	0.043	0.02	0.029	0.176	0.055	0.061
$Doc2$	0.021	0.066	0.042	0.042	0.047	0.03	0.016	0.073	0.068	0.054
$Doc3$	0.679	0.019	0.006	0.009	0.016	0.01	0.006	0.018	0.013	0.015
$Doc4$	0.843	0.008	0.003	0.003	0.002	0.00	0.002	0.008	0.001	0.002
$Doc5$	0.024	0.003	0.004	0.005	0.006	0.00	0.007	0.011	0.001	0.002
$Doc6$	0.044	0.017	0.013	0.011	0.018	0.01	0.012	0.036	0.044	0.036
$Doc7$	0.034	0.089	0.121	0.055	0.038	0.05	0.029	0.035	0.034	0.018
$Doc8$	0.019	0.026	0.039	0.026	0.032	0.58	0.003	0.021	0.019	0.012
$Doc9$	0.029	0.269	0.058	0.058	0.058	0.03	0.019	0.043	0.059	0.022

Once we have the posterior probability we use a naive bayes method to classify the documents. That is, whichever class probability is highest we assign that class to a given document

Chapter 4

Results and Interpretation

For the evaluation of this project we have considered the 4 statistical measures which are apt for the task (classification) at hand.

1. **Precision:** In a classification task, the precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).
   $Precision = \frac{TruePositive}{TruePositive+FalsePositive}$

2. **Recall:**Recall in this context is defined as the number of true positives divided by the total number of elements that actually belong to the positive class(i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been)
   $Recall = \frac{TruePositive}{TruePositive+FalseNegative}$

3. **F1 Score:** F1 Score is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. In simple terms, F1 score is the harmonic mean of Precision and Recall

F1 Score = $\frac{Precision*Recall}{Precision+Recall}$

4. **Accuracy:** Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined.
   Accuracy = $\frac{TruePositive+TrueNeagtive}{TruePositive+FalsePositive+TrueNegative+FalseNegative}$

Confusion Matrix:
We have also provided the confusion matrix of for each model. The confusion matrix, also known as an error matrix is a specific table layout that allows visualization of the performance of an algorithm. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.
The diagonals of the confusion matrix provides the True positive values of each category. The False Negative values for each category are obtained by summing the values of the corresponding row leaving the diagonal element. The False positive values for each category are obtained by summing the values of the corresponding column, leaving the diagonal elements. True Negative values for each category are obtained by summing all the elements of the matrix which are not part of the corresponding row and column
Below is the key for confusion matrix. We have represented each category with an alphabet in the confusion matrix.

4.1 Multinomial Results

We use a machine learning approach to evaluate our modelling. We form a prior from a section of data itself called training data. Using our multinomial model, posterior probabilities are calculated. Using this posterior probability a model is formed which is then applied to a separate section of data called testing data. The results of this evaluation on test data are summarized in the tables.

Category	Number of Documents
alt.atheism	A
comp.graphics	B
comp.os.ms-windows.misc	C
comp.sys.ibm.pc.hardware	D
comp.sys.mac.hardware	E
comp.windows.x	F
misc.forsale	G
rec.autos	H
rec.motorcycles	I
rec.sport.baseball	J
rec.sport.hockey	K
sci.crypt	L
sci.electronics	M
sci.med	N
sci.space	O
soc.religion.christian	P
talk.politics.guns	Q
talk.politics.mideast	R
talk.politics.misc	S
talk.religion.misc	T

4.1.1 Confusion Matrix

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*
A	166	0	0	1	0	1	0	0	1	1	1	3	0	6	3	12	34	8	0	1
B	1	252	15	12	9	18	1	2	1	5	2	41	4	0	6	15	4	1	0	0
C	0	14	258	45	3	9	0	2	1	3	2	25	1	0	6	23	2	0	0	0
D	0	5	11	305	17	1	3	6	1	0	2	19	13	0	5	3	1	0	0	0
E	0	3	8	23	298	0	3	8	1	3	1	16	8	0	2	8	3	0	0	0
F	1	21	17	13	2	298	1	0	1	1	0	23	0	1	4	10	2	0	0	0
G	0	1	3	31	12	1	271	19	4	4	6	5	12	6	3	9	3	0	0	0
H	0	1	0	3	0	0	4	364	3	2	2	4	1	1	3	3	4	0	1	0
I	0	0	0	1	0	0	2	10	371	0	0	4	0	0	0	8	2	0	0	0
J	0	0	0	0	1	0	0	4	0	357	22	0	0	0	2	9	1	1	0	0
K	0	0	0	0	0	0	0	1	0	4	387	1	0	0	1	5	0	0	0	0
L	0	2	1	0	0	1	1	3	0	0	0	383	1	0	0	3	1	0	0	0
M	0	4	2	17	5	0	2	8	7	1	2	78	235	3	11	15	2	1	0	0
N	2	3	0	1	1	3	1	0	2	3	4	11	5	292	6	52	6	4	0	0
O	0	2	0	1	0	3	0	2	1	0	1	6	1	2	351	19	4	0	1	0
P	2	0	0	0	0	0	0	0	1	0	0	0	0	1	2	392	0	0	0	0
Q	0	0	0	1	0	0	2	0	1	1	0	10	0	0	1	6	341	1	0	0
R	0	1	0	0	0	0	0	0	0	1	0	2	0	0	0	24	3	344	1	0
S	2	0	0	0	0	0	0	1	0	0	1	11	0	1	7	35	118	5	129	0
T	33	2	0	0	0	0	0	0	0	1	1	3	0	4	4	131	29	5	3	35

category	precision	recall	f1-score	support
alt.atheism	0.80	0.52	0.63	319
comp.graphics	0.81	0.65	0.72	389
comp.os.ms-windows.misc	0.82	0.65	0.73	394
comp.sys.ibm.pc.hardware	0.67	0.78	0.72	392
comp.sys.mac.hardware	0.86	0.77	0.81	385
comp.windows.x	0.89	0.75	0.82	395
misc.forsale	0.93	0.69	0.80	390
rec.autos	0.85	0.92	0.88	396
rec.motorcycles	0.94	0.93	0.93	398
rec.sport.baseball	0.92	0.90	0.91	397
rec.sport.hockey	0.89	0.97	0.93	399
sci.crypt	0.59	0.97	0.74	396
sci.electronics	0.84	0.60	0.70	393
sci.med	0.92	0.74	0.82	396
sci.space	0.84	0.89	0.87	394
soc.religion.christian	0.44	0.98	0.61	398
talk.politics.guns	0.64	0.94	0.76	364
talk.politics.mideast	0.93	0.91	0.92	376
talk.politics.misc	0.96	0.42	0.58	310
talk.religion.misc	0.97	0.14	0.24	251
avg/total	0.82	0.77	0.77	7532

Table 4.1: Result of using Multinomial Sampling Model
Accuracy: 77.389803505

From the tables we can see that the classification accuracy is around 77%. Precision = 82%, Recall = 82%, F1 score = 82%.

4.2 Bernoulli Results

For evaluating Bernoulli model, we use similar approach as used in the multinomial model. The results are available in the tables.

category	precision	recall	f1-score	support
alt.atheism	0.92	0.32	0.47	319
comp.graphics	0.58	0.63	0.61	389
comp.os.ms-windows.misc	0.33	0.01	0.01	394
comp.sys.ibm.pc.hardware	0.43	0.81	0.56	392
comp.sys.mac.hardware	0.64	0.76	0.70	385
comp.windows.x	0.84	0.61	0.70	395
misc.forsale	0.30	0.93	0.45	390
rec.autos	0.67	0.78	0.72	396
rec.motorcycles	0.74	0.91	0.82	398
rec.sport.baseball	0.77	0.87	0.82	397
rec.sport.hockey	0.99	0.83	0.90	399
sci.crypt	0.82	0.69	0.75	396
sci.electronics	0.57	0.67	0.62	393
sci.med	0.84	0.52	0.64	396
sci.space	0.88	0.68	0.77	394
soc.religion.christian	0.53	0.80	0.64	398
talk.politics.guns	0.74	0.57	0.64	364
talk.politics.mideast	0.96	0.65	0.78	376
talk.politics.misc	0.89	0.19	0.31	310
talk.religion.misc	1.00	0.00	0.01	251
avg / total	0.71	0.63	0.61	7532

Table 4.2: Result of using Bernoulli Sampling Model
Accuracy: 63.0775358471

4.2.1 Confusion Matrix

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*
A	101	2	0	17	8	0	34	14	22	12	0	1	8	9	3	84	0	4	0	0
B	0	245	0	28	12	14	63	1	0	1	0	10	8	1	4	2	0	0	0	0
C	0	50	2	205	24	24	55	4	4	2	0	12	3	0	4	4	0	0	1	0
D	0	2	1	319	8	2	36	1	0	0	0	2	21	0	0	0	0	0	0	0
E	0	3	1	25	291	0	50	1	0	2	0	1	7	0	3	1	0	0	0	0
F	0	60	2	27	6	240	51	0	1	0	0	4	3	1	0	0	0	0	0	0
G	0	0	0	14	2	0	362	3	1	0	0	1	6	1	0	0	0	0	0	0
H	0	1	0	5	2	0	59	308	5	0	0	0	15	0	1	0	0	0	0	0
I	0	0	0	1	1	0	22	7	364	0	0	0	2	0	0	0	0	1	0	0
J	0	1	0	2	1	0	40	1	1	346	0	0	0	0	0	4	0	0	1	0
K	0	0	0	1	0	0	39	1	2	19	330	0	3	0	0	4	0	0	0	0
L	0	12	0	18	14	1	44	2	4	3	0	273	22	0	1	1	0	0	1	0
M	0	11	0	30	7	0	60	3	1	0	0	15	262	3	1	0	0	0	0	0
N	0	11	0	12	22	0	88	11	11	1	0	0	25	206	1	4	1	1	2	0
O	0	11	0	2	8	4	51	7	3	0	0	2	27	4	269	5	0	0	1	0
P	0	3	0	10	8	1	42	0	3	4	0	0	7	0	0	320	0	0	0	0
Q	0	0	0	7	7	0	37	45	20	10	0	6	13	4	2	6	207	0	0	0
R	0	3	0	4	15	0	37	7	14	15	1	2	4	1	0	27	0	246	0	0
S	0	1	0	6	7	0	29	34	21	19	1	5	16	10	12	31	58	1	59	0
T	9	3	0	14	9	1	26	11	15	14	0	0	5	6	4	116	14	2	1	1

4.3 Impact of study and Future Work

Implications:

- Accuracy of multinomial model = 77.39 %
  Accuracy of Bernoulli model = 63.08 %

- It is observed from the results (Accuracy) that Navies Bayes approach can be effectively used to classify text documents.

- Despite the naive assumption that all the words of the document are independent of each other, the accuracy of the models are pretty decent.

- The precision, recall, F1score and Accuracy proves that the Multinomial model out performs the Bernoulli Model.

- The results prove that , representing words in terms of frequency (Multinomial model) is a better than representing them in to of just presence and absence (Bernoulli model)

The method and results obtained lead to some interesting future works such as

- Identifying why some classes are misclassified
- Topic formation based on significant terms
- Implementing semantic search based on bayesian statistics

References

1. http://qwone.com/~jason/20Newsgroups/ [Accessed 26 April 2016]

2. https://en.wikipedia.org/wiki/F1_score [Accessed 26 April 2016]

3. https://en.wikipedia.org/wiki/Accuracy_and_precision#In_binary_classification [Accessed 26 April 2016]

4. https://en.wikipedia.org/wiki/Naive_Bayes_classifier [Accessed 26 April 2016]

5. https://en.wikipedia.org/wiki/Document_classification#Applications [Accessed 26 April 2016]