Implementation Assignment 2: Naive Bayes

CS534 - Machine Learning

Amir Azarbakht
Mandana Hamidi

May 2, 2014

Abstract

The main objective of this programming assignment is to implement the Naive Bayes classifier for document classification with both the Bernoulli model and the Multinomial model. The results of these methods on the document classifiers shows that The accuracy of these methods are almost equal, however, the Naive Bayes with Bernoulli has a higher accuracy comparing to the Naive Bayes with Multinomial method.

1 Naive Bayes

In this assignment we are asked to implement the Naive Bayes classifier for document classification using two following methods, Binomal and Multinomial models:
Bernoulli model: for this model a document is described by a set of binary variables, and each variable corresponds to a word in the vocabulary V and represents its presence/absence. The probability of observing a document doc given its class label y is then defined as:

$$p(doc|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}(1 - p_{i|y})^{(1-x_i)}$$

where $p_{i|y}$ denotes the probability that the word i will be present for a document of class y. If $x_i = 1$, the contribution of this word to the product will be $p_{i|y}$, otherwise it will be $1 - p_{i|y}$.
Multinomial model, for this model a document doc is represented by a set of continuous variables, and each variable x_i also corresponds to the i^{th} word in the vocabulary and represents the number of times it appeared in the document. The probability of observing a document x given its class label y is defined as:

$$p(doc|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}$$

where doc contains a set of feature vectors $doc = \left[x_1, x_2, ..., x_{|V|}\right]$

1.1 Data set

The data set is the classic 20-newsgroup data set. There are six files.
vocabulary.txt is a list of the words that may appear in documents. The line number is word's id in other files.

newsgrouplabels.txt is a list of newsgroups from which a document may have come. Again, the line number corresponds to the label's id, which is used in the .label files.
train.label contains one line for each training document specifying its label. The document's id (docId) is the line number.
test.label specifies the labels for the testing documents.
train.data describes the counts for each of the words used in each of the documents. It uses a sparse format that contains a collection of tuples "docId wordId count". The first number in the tuple species the document ID, and the second number is the word ID, and the final number species the number of times the word with id wordId in the training document with id docId. For example "5 3 10" species that the 3rd word in the vocabulary appeared 10 times in document 5.
• **test.data** is exactly the same as train.data, but contains the counts for the test documents. Note that you don't have to use the vocabulary.txt file in your learning and testing. It is only provided to help possibly interpret the models. For example, if you find that removing the first feature can help the performance, you might want to know what word that first feature actually corresponds to in the vocabulary.

1.2 Basic implementation:

We implemented the Naive Bayes algorithm with Bernoulli and the Multinomial method that we explained in the previous section. The only thing that we did was in computing the likelihoods,$p(doc|y)$, instead of computing the following equations :

$$p(doc|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}(1 - p_{i|y})^{(1-x_i)}$$
$$\text{or } p(doc|y) = \prod_{i=1}^{|V|} p_{i|y}^{x_i}$$

we computed:

$$log\left(p(doc|y)\right) = \sum_{i=1}^{|V|} x_i log(p_{i|y}) + (1 - x_i) log(1 - p_{i|y})$$
$$\text{or } log\left(p(doc|y)\right) = \sum_{i=1}^{|V|} x_i log(p_{i|y})$$

The overall testing accuracy (number of correctly classified documents over the total number of documents) for both models are shown in $Table\ 2$.
According to the results, the classifier with Multinomial performs better than the classifier with Bernoulli

Table 1: Compare the accuracy of Naive Bayes with Bernoulli and Multinomial methods

	Naive Bayes (Bernoulli Method)	Naive Bayes (Multinomial Method)
Accuracy	0.7702	0.7785

method. The reason is that in Multinomal method we consider the number of frequency of each word into account.
In order to show the find whether there are any news groups that are confused more often than others, we computed the K by K confusion matrix, where $K = 20$ is the number of classes, and the i, j^{th} entry of the matrix shows the number of class i documents being predicted to belong to class j. $Figure\ 1$ and $Figure\ 2$ show the confusion matrices that we computed for both Bernoulli and Multinomial methods:

From the confusion matrices, it can be conclude that the value of the elements on the diagonal of these matrices are higher than the other elements value. However there are some high element values, which are not on diagonal, with high values including:

1.3 Priors and over-fitting:

The main objective of this part is to select different priors for the Multinomial model and test this model with these priors. These priors are used in Laplace smoothing for the MAP estimation. Actually, we use $Dirichlet(1+\alpha, ..., 1+\alpha)$ distribution as a prior and each time we change the parameter of this distribution from 0.00001 and 1.
Thus using this prior the Laplace smoothing that we with get is as follows:

$$p(z = k) = \frac{n_k + \alpha_k - 1}{N + \sum_{i=1}^{K} \alpha_i - K} \tag{1}$$

We computed the the accuracy on the test set for different α values. $Figure\ 3$ shows a plot with value of α on the X-axis and test set accuracy on the Y-axis. Use a logarithmic scale for the X-axis. Comment on how the test set accuracy change as α changes and provide a short explanation for your observation.

1.4 Identifying important features:

In this part, the objective is to design and test a heuristic to reduce the vocabulary size and improve the classification performance. The method that we used is a follows:
1) Remove the features(vocabularies) which has not been used in the whole training data sets. Actually, these features are not useful and have not been seen in any document.

2) Remove the common features(vocabularies), it means that if a feature has been used in all documents of all classes, so this word is not the one that distinguish a class from other classes. In order to remove the common features, we removed the vocabularies, whose normalized variances is smaller than a threshold equal to 0.6.

$$\frac{var(Number\ of\ feature\ in\ categeroies)}{mean(Number\ of\ feature\ in\ categeroies)} < 0.6 \tag{2}$$

The number of features that were removed is 7632 , Some features like of, usa, $from$, and, $other$, are, and we , has seen in all the documents, so it makes sense that they cannot be useful distinguishing between classes. Some features like $etrbom$, $gosple$, $deist$ and $spreadeth$ has not been used in any document.

Table 2: Compare the accuracy of Naive Bayes with Multinomial with and without feature reduction

Naive Bayes (Multinomial) without feature reduction	0.7785
Naive Bayes (Multinomial) with feature reduction	0.8099

The accuracy of the algorithm increased when we reduced the feature vector size, the main reason is that the in Multiomial the $\sum_i P_i = 1$ and the probability of the irrelevant features affected the probability of the other relevant features. When we remove the irrelevant features, the probability of the good features increased so the classifies performs better.

2 Submitted codes

We implemented the code of this assignment in Matlab language. The submitted codes contains the following files:

1) $NaiveBayseBernoulliMultinomial.m$: This file contains the code of the Naive Bayes algorithm with Bernoulli and Multinomial methods. There is a variable, named $testMethod$, whose value correspond to the method that we are testing. If $testMethod = 0$, the algorithm works with Bernoulli method, otherwise if $testMethod = 1$, the algorithm works with Multinomial method

2) $NaiveBayseBernoulliMultinomialDrichletTest.m$: This file contains the code for computing different Dirichlet distribution as the prior of the algorithm

3) $NaiveBayseBernoulliMultinomialFeaturesReduxtion.m$: This file contains the code for identifying important features

1	36902	0	0	0	0	40	0	0	0	0	5	149	53	305	977	9059	307	1519	577	1062
2	283	37801	302	1259	832	2046	0	310	76	0	0	3064	324	443	1003	383	0	0	145	0
3	277	3664	23456	5746	630	2751	0	18	106	57	28	2381	129	326	369	707	0	0	1202	78
4	0	581	1269	28886	1597	179	304	492	0	0	26	945	2967	0	34	0	142	0	0	0
5	0	587	756	2639	24483	1703	45	495	70	78	0	538	1170	664	196	67	167	0	985	0
6	0	2894	1438	921	143	34607	10	0	155	37	0	897	0	0	90	192	18	54	118	0
7	0	493	202	4304	970	48	15817	2170	521	0	42	121	1130	71	244	181	126	295	610	0
8	60	66	0	82	0	25	147	40450	213	57	0	218	218	0	115	9	510	100	908	0
9	0	45	0	0	0	0	0	2918	36977	90	0	65	0	35	0	55	452	434	1017	0
10	1183	0	0	9	13	137	90	94	63	43119	979	99	131	0	0	339	118	302	1555	13
11	166	0	0	0	0	0	174	0	0	277	45814	265	0	106	59	210	64	247	312	0
12	0	166	71	27	26	286	31	63	0	0	0	48507	142	167	0	242	786	0	975	56
13	167	1032	0	2940	833	308	14	1274	88	0	38	6635	27020	440	559	437	0	229	0	0
14	1018	282	41	106	0	0	0	112	0	56	0	3	170	47675	238	1834	219	941	1584	0
15	380	874	0	0	0	341	0	0	30	0	19	329	381	590	45456	608	71	211	3596	69
16	520	129	0	0	22	183	0	0	0	0	0	63	0	67	0	69224	182	157	82	22
17	17	0	0	0	60	0	12	145	83	0	8	677	0	153	139	194	52080	388	2023	322
18	1602	46	0	0	0	0	0	55	72	18	41	332	0	0	0	1284	99	75534	2152	0
19	574	42	0	0	0	109	0	169	0	0	0	303	0	167	1546	299	16638	456	35346	84
20	6004	311	0	0	0	0	0	0	37	0	0	128	0	267	1225	16091	3268	702	1336	10453

Figure 1: Confusion Matrix of Naive Bayes with Multinomail Method

1	40109	40	0	0	186	0	0	0	214	0	5	149	131	173	837	484	385	0	442	7800
2	240	38183	1843	1757	3051	827	1142	0	0	54	0	93	443	0	17	0	0	0	0	621
3	0	1547	30244	5376	2569	463	523	0	0	191	46	0	0	0	264	0	0	0	101	601
4	0	657	1912	28874	3258	0	728	141	0	0	26	0	1792	0	34	0	0	0	0	0
5	0	164	1193	1243	30247	0	924	193	0	78	0	0	601	0	0	0	0	0	0	0
6	0	3477	4165	944	628	31562	301	88	0	31	0	72	0	0	20	0	18	0	122	146
7	0	43	112	1646	533	0	24299	501	73	0	0	0	138	0	0	0	0	0	0	0
8	129	180	105	400	72	0	724	39402	663	25	0	0	860	0	56	0	0	0	351	211
9	0	114	0	0	269	206	328	1935	38815	80	0	0	148	0	0	0	0	0	0	193
10	567	87	112	52	51	0	379	208	0	46427	196	0	165	0	0	0	0	0	0	0
11	310	0	23	0	189	0	413	0	0	1610	44482	0	148	30	0	0	0	0	0	489
12	155	1093	502	264	1343	323	174	0	108	135	0	43836	1320	39	81	0	1296	0	236	640
13	80	1288	2188	4420	3878	0	620	512	251	46	0	17	28642	72	0	0	0	0	0	0
14	1589	944	251	477	1390	56	1912	1301	816	56	0	0	1441	40957	54	158	32	325	1208	1312
15	935	1540	281	0	633	0	368	537	249	0	19	15	1566	446	43644	0	0	98	1781	843
16	1463	267	177	97	251	0	246	18	17	0	0	0	0	0	82	47274	166	0	0	20593
17	37	0	70	92	330	0	271	299	314	49	0	1272	368	153	73	59	48140	65	1007	3702
18	7572	46	119	0	117	12	78	135	0	234	59	59	0	0	214	278	229	62647	4852	4584
19	1298	151	128	0	191	0	58	0	0	144	0	290	0	80	1546	3	14261	13	31970	5600
20	4996	360	0	0	37	0	0	53	0	182	0	0	0	116	1171	1542	1358	538	666	28803

Figure 2: Confusion Matrix of Naive Bayes with Bernoulli Method

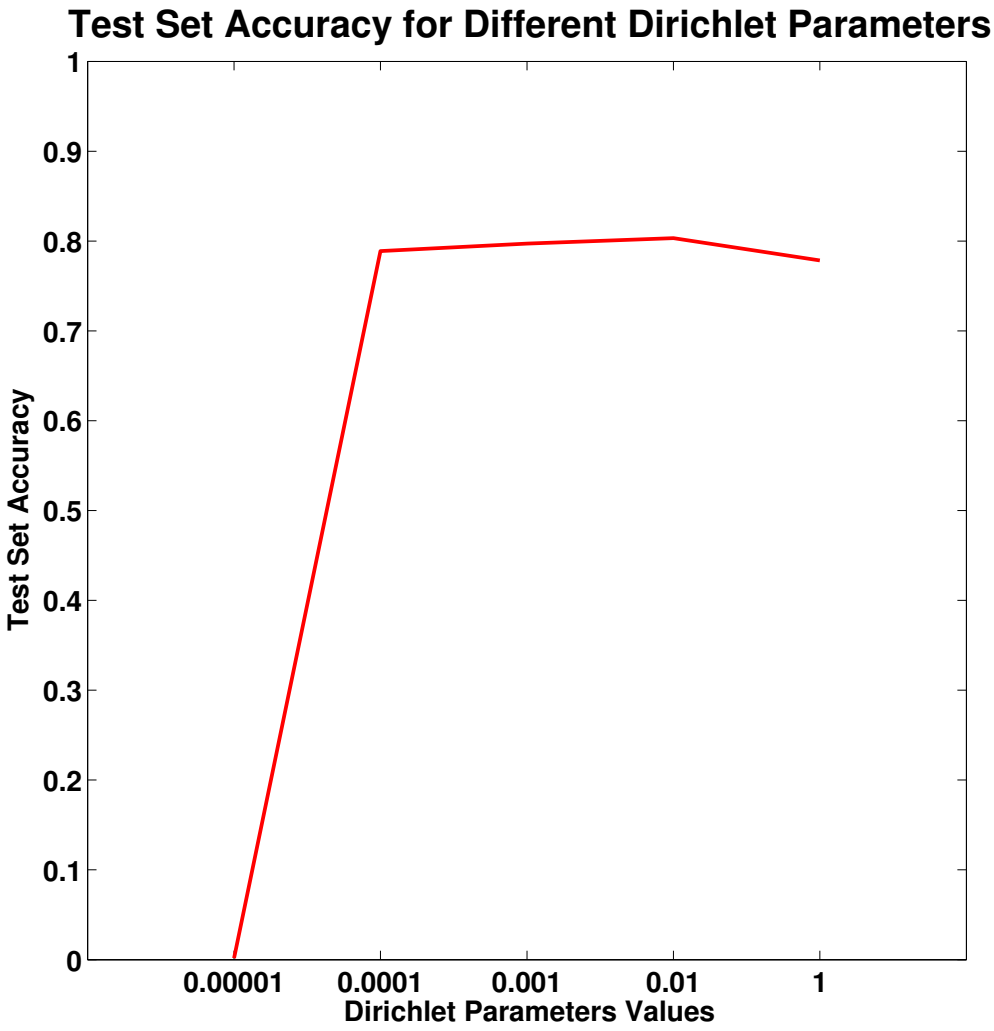

Figure 3: Accuracy on the test set for different Dirichlet α values