INDIAN INSTITUTE OF TECHNOLOGY ROPAR

CSL-603

Spam Email Filtering and Digit Classfication

Submitted To:
Narayanan C Krishnan
Computer Science
Department

Submitted By :
Siddharth Nahar
2016CSB1043
Group-G1
5th Semester

Contents

1 Naive Bayes Model and Basic Analysis

This is Report for Naive Bayes Classifier to classify Emails as Spam or Ham. For this classification we considered subset of 2005 TREC Public Spam Corpus with 5000 Training Examples and 1000 Test Examples.

From the Training Examples We have used m-estimate method to obtain conditional probabilities.Then computed log likelihood sum and output the max probablity label. Obtained 89.3% accuracy on Test Set.

1.1 Basic Analysis

x_j = Attribute word in Spam or Ham
n_j = Total frequency of word x_j in Spam or Ham
n = Sum of Total frequency of words in Spam or Ham
m = Parameter for m-estimate
p = Probability parameter for m-estimate
$P(spam)$ = Prior Probability of Spam emails in Training Example.
$P(ham)$ = Prior Probality of Non-Spam emails in Training Examples.

$$P(x_j/spam) = \frac{n_j + mp}{n_{spam} + m}, \quad P(x_j/ham) = \frac{n_j + mp}{n_{ham} + m}$$

$$P(spam/x) = (\prod_{j=1}^{N} P(x_j/spam)^{f_j})P(spam)$$

$$P(ham/x) = (\prod_{j=1}^{N} P(x_j/ham)^{f_j})P(ham)$$

For Numerical Stabiltiy :-

$$\log(P(c_k/x)) = \sum_{j=1}^{N} f_j log(P(x_j/c_k)) + log(P(c_k))$$

$$ClassPredicted = \max(\log(P(spam/x)), \log(P(ham/x)))$$

1.1.1 Results Obtained

- Vocabulary consists of 996 Words.
- m = len(Vocabulary) and p = 1/m
- P(spam) = 0.4804, P(ham) = 0.5196
- Most Indicative words in Spam = enron, a, the, corp, to
- Most Indicative words in Ham = aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa enron, the, to, a
- Accuracy = 89.3%

1.2 Difficulties faced while computing Posterior Probabilities and Solutions

- If Word is only present in Spam Vocabulary but absent in Ham Vocabulary.Conditional Probability in that case would be zero. $P(x_j/ham) = 0$.

  Solution :- Use m-estimate so that conditional probability would be non zero and very less suggesting that word would rarely occur in that class.

- Handling of Rare Words :- If we encounter a new word in Test Set that is not present in Vocabulary then we would result in 0 probability in both case. So confusion in prediction.

  Solution :- I have just ignore the word, As we have no prior information about that word So it wouldn't affect the model created.Or we could use Smooth Laplace formula and assign it 1/len(Vocabulary) as probability.So it's still classifies according to model.

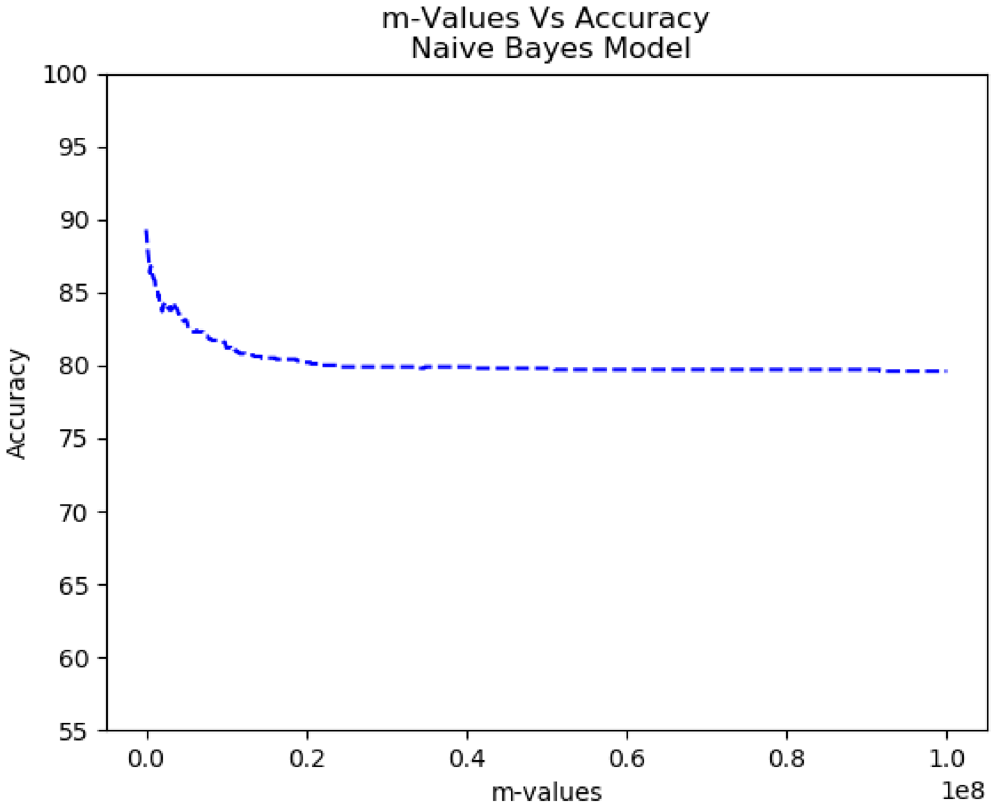
- We have to perform Product of probabilities raise to their frequency which may result in very small number.

  Solution :- We know log is monotonic so max(a,b) = max(log(a),log(b)) .So we perform summation of log's which prevents the situation.

1.3 Accuracy as function of m

In m-estimate m represents "User's Confidence in Current Probability of class".How much Prior Estimate can be trusted.

- High Value of m represents High Confidence in value .So it allows to update prior Estimate only if high evidence is available for current value i.e $x_j \geq m$ and $n \geq m$.This may result in Over-Prior and bad prior estimate.As we could clearly se in graph below.
- Low Value of m shows user is not so sure about it's prior.It changes probability with very few experimental trials. This shows that we have minimum information about prior and we totally believe in Experimental trials.



1.4 Methods to beat the Classifier

- Bayesian Poisoning :- Use more amount of legitimate words to cross threshold of Model. If we want to use any Spam word use it with more words which have high probabilities for non-spam case.
- Modification of Words :- We use Vocabulary for our prediction.If word is not present in Vocabulary we ignore it. So if we modify any spam word by adding different characters to beat the classifier.If SPAM word is modified as ¡S!PAM¿ ,SPPAM could be inserted in mail.
- Replace text with images which consists of this word.As we only consider body of mail,it would beat classifier.SPAM word is written image and send as attachment.

2 K-Means Clustering

For K-Means Clustering , We are using subset of MNIST handwritten digits dataset.We have 5000 samples of dataset.

For this Part I am using k-means algorithm of inbuilt scipy library.Results are based on that algorithm.

2.1 Results

- k = 10, Accuracy = 58.2%
- Centroids Formed are as follows :-

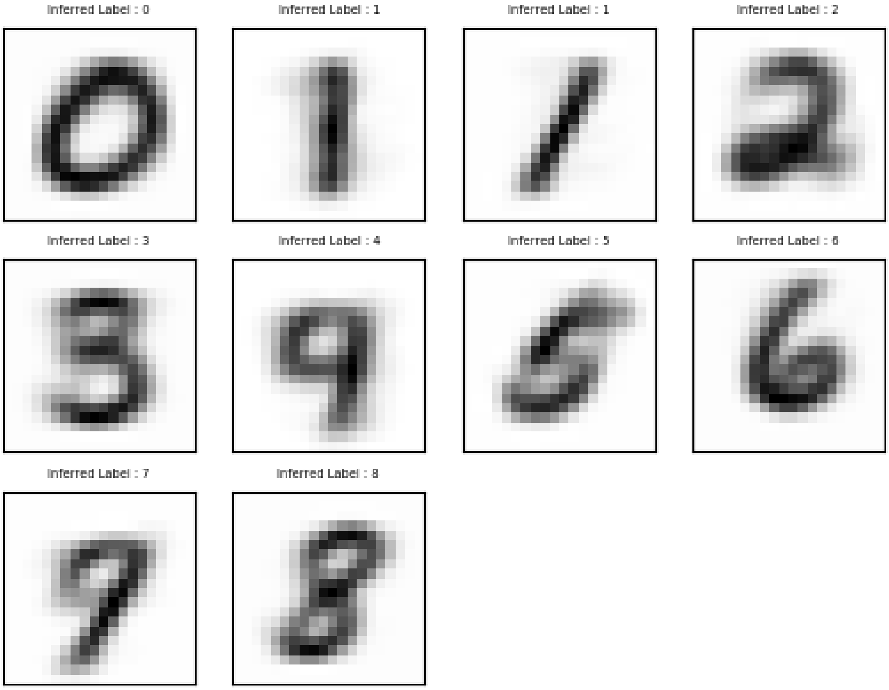


Figure 1: Centroids for k = 10

- Confusion Matrix :-

Table 1: Confusion Matrix for k = 10

		Predicted Labels									
		0	1	2	3	4	5	6	7	8	9
Actual Labels	0	395	3	0	26	7	0	19	3	47	0
	1	0	497	0	1	1	0	0	0	1	0
	2	3	81	333	31	14	0	16	5	17	0
	3	1	55	16	310	10	0	2	10	96	0
	4	0	36	2	0	177	0	10	272	3	0
	5	5	73	1	157	23	0	11	33	197	0
	6	6	55	5	2	43	0	371	0	18	0
	7	1	55	1	0	47	0	0	396	0	0
	8	0	69	3	145	23	0	2	30	228	0
	9	2	23	0	8	125	0	2	337	3	0

- We could see that Label 9 is not found as centroid . From Confusion matrix we could see 4,5 labels are being confused by 9.
- For label 4 , 7 is predicted more. similarly for 3, 8 is predicted more.
- (4,7,9),(3,8,5) is being confused most.

2.2 Results for k = 15

- Accuracy = 66.58%
- We can see as we increase k accuracy increases.
- As we increase k we see more splits occur from k = 10.Some digits which were merged gets split and this increases accuracy.We could see results for k = 15 below.
- Digits which got split are 7 to 4. 8 to 0,3 and 5 .
- 4,7 differs very closely for only standing line in 4 increasing k increases that difference. 8,3,5 have similar curve shape So there is more confusion for prediction
- Confusion Matrix :-

Table 2: Confusion Matrix for k = 15

		Predicted Labels									
		0	1	2	3	4	5	6	7	8	9
Actual Labels	0	443	0	0	20	3	13	19	1	1	0
	1	0	495	0	1	1	1	0	0	2	0
	2	6	81	322	28	13	9	20	8	13	0
	3	1	42	7	392	15	9	2	5	27	0
	4	0	33	5	0	401	7	11	42	1	0
	5	6	19	1	220	42	194	10	4	4	0
	6	6	30	1	4	6	3	450	0	0	0
	7	1	50	0	0	30	2	0	416	1	0
	8	3	52	3	130	28	14	3	1	266	0
	9	2	32	1	8	310	2	2	143	0	0

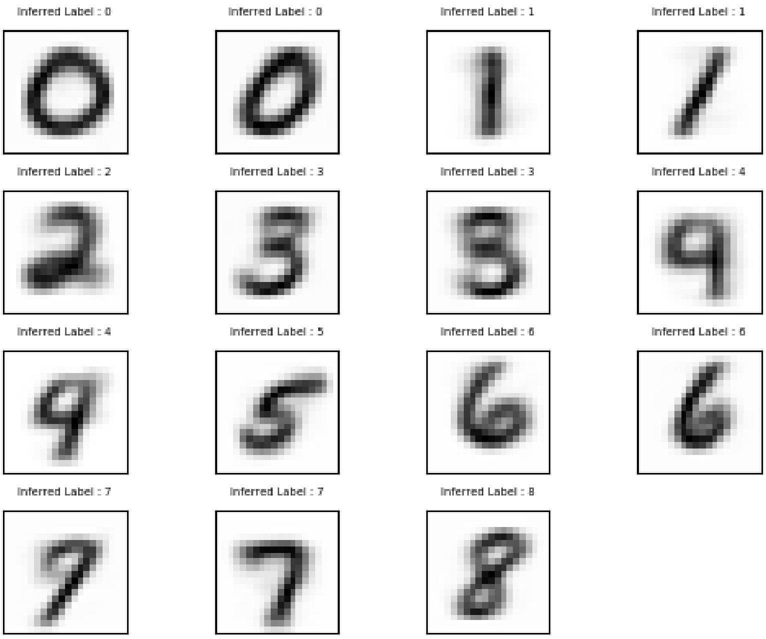


Figure 2: Centroids for k = 15

2.3 Results for k = 5

- Accuracy = 43.48%
- We can see as we decrease k accuracy decreases.
- As we decrease k we see more merge occur from k = 10.Some digits which were split gets and this decreases accuracy.We could see results for k = 5 below.
- Digits which got merged are 8 and 5 are merged with 3,2 and 6 are merged 4,7 are merged.Some 7 are merged with 1 unexpectedly 7 and 1 are rarely merged. 7 is more like to 4 .
- 4,7 differes very closely for only standing line in 4 decreasing k decreases that difference. 8,3,5 have similar curve shape So there is more confusion for prediction . 2,6 also have similar nature.
- Confusion Matrix :-

Table 3: Confusion Matrix for k = 5

		Predicted Labels									
		0	1	2	3	4	5	6	7	8	9
Actual Labels	0	429	3	0	36	0	0	27	5	0	0
	1	0	495	0	3	0	0	0	2	0	0
	2	6	91	0	56	0	0	336	11	0	0
	3	3	56	0	412	0	0	7	22	0	0
	4	0	41	0	0	0	0	33	426	0	0
	5	10	160	0	248	0	0	14	68	0	0
	6	7	69	0	9	0	0	410	5	0	0
	7	4	65	0	0	0	0	3	428	0	0
	8	2	165	0	277	0	0	19	37	0	0
	9	2	55	0	11	0	0	7	425	0	0

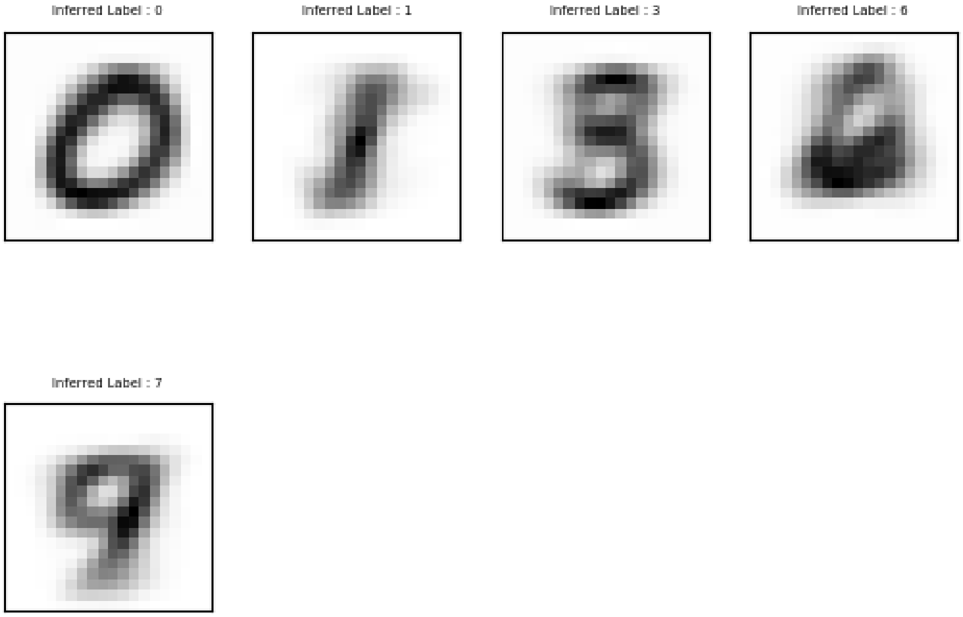


Figure 3: Centroids for k = 5