Naïve Bayes Classifier with Various Smoothing Techniques for Text Documents

Shruti Aggarwal

[1]Assistant Professor, Dept. of CSE, S.G.G.S.W.U., Fatehgarh Sahib (Punjab), India

Devinder Kaur

[1]Research Scholar, Dept. of CSE, S.G.G.S.W.U., Fatehgarh Sahib (Punjab), India,

Abstract: **Due to huge amount of increase in text data, its classification has become an important issue, now days. There are many good classification techniques discussed in this paper. Each classification method has its own assumptions, advantages and limitations. One of the most widely used classifier is Naïve Bayes which performs well with different data sets. Various Smoothing techniques are applied on Naïve Bayes. The idea behind them is to improve the classification accuracy and performance.**

Keywords**: Text classification, Naïve Bayes, Jelinek-Mercer, Smoothing, Dirichlet, Two-Stage, Absolute Discounting.**

I. INTRODUCTION

With large increase in the amount of text documents, manual handling is not a feasible solution and it has become necessary to categorize them in different classes. There is various classification methods developed, but the choice of using these techniques mainly depend upon the type of data collections. In the next section, Some Classifiers are discussed. Few methods perform well on numerical and text data like Naive Bayes but neural networks handle both discrete and continuous data. KNN is a time consuming method and finding the optimal value is always an issue. Decision tree reduces the complexity but fails to handle continuous data. Naïve Bayes along with its simplicity is computationally cheap also. In the third section of the paper, Naïve Bayes classifier is discussed in detail. One of the major drawback of Naïve Bayes is of unseen words, which can be eliminated by applying smoothing techniques. In the IV section, various smoothing methods when applied on Naïve Byes are discussed and their performances are compared.

II CLASSIFICATION TECHNIQUES

There are number of techniques available for classifying text such as:

A. Naïve Bayes Classifier: [1]

A Naïve Bayes is a simple classifier based on the probabilistic model which is implemented using Bayes theorem with strong independent assumption. The output of the classifier is the probability, according to which the document is classified. The probability is calculated using the formula:

$$C_{NB} = \text{argmax}_{cj \in c} P(c) * \prod_{1<k<d} P(wk|c) \ldots\ldots\ldots(1)$$

B. Nearest Neighbor Classifier:[2]

This is the non-parametric method. In this classifier, TF-IDF weighing scheme is used with Cosine similarity and Euclidean distance to find the similarity of documents. Then, k the most similar documents are selected. The similarity between d1 and d2 is defined as:

$$S(d1,d2) = \frac{T1T2}{|T1||T2|} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)[2]$$

where T1 and T2 are the feature vectors of the d1 and d2 documents.

C. Centroid Based Classifier:[9]

It is the most popular supervised approach with relatively low computation. Given a set S of documents and their representation, we need to compute the summed centroid C^s and normalized centroid C^N of class C_j. then we have to calculate similarity between class C_j and document d. then document d is assigned to the most similar centroid.

D. Decision Trees: [1]

The Decision Tree classifier is a tree in which the internal nodes are the terms and branches are labeled by weights and leaf nodes are the classes. This classifier classifies a text document d by repeatedly testing for the weights of the terms until a leaf node is reached. The label of the reached leaf node is then assigned to the tested document d.

E. Support Vector Machines:[1][9]

It is the most widely used algorithm in text classification. A document d_j is represented by vector t_{d1} of its words counts. Here, a document is classified into two classes-positive class and negative class. A hyper plane is defined by setting y=0 in the following class:

Sr .No	Name of Classifier	Type of Algorithm	Complexity	Criteria
1.	Nearest Neighbor	M-Way	Training-> (NL_d) Testing-> $o\left(\frac{N}{V}{L_v}^2\right) + O(N)$ Where, L_d = The Average word count L_v = The Average unique words in D	$S(d1,d2) = \frac{T1T2}{\|T1\|\|T2\|}$ Calculate the similarity between documents d1 and d2. T1 and T2 are the Feature Vectors.
2.	Naïve Bayes	Binary	$\theta(\|C\|\|V\|)$	$C_{NB} = argmax_{c_i \epsilon c}$ P(c) $* \prod_{1<k<d} P(w\text{k}\|\text{c})$ Where P(c) is prior Probability and $P(w\text{k}\|\text{c})$ is posterior Probability of word w in Class c.
3.	Support Vector Machine	2-class or M-class	Training time on M documents -> $O(MN^c)$	$Y= f(t_d) = b_0+\sum_{j=1} b_j t_{dj}$ A new document is classified to positive class if $f(t_d)>0$ otherwise negative class. Where T_d is a document vector.
4.	Centroid Based	M-Way	O(TKW) Where, T = test documents K = Number of classes W= Words in total	Summed Centroid $C_i^s = \sum_{d \epsilon ci} d$ Normalized Centroid $C_i^N = {c^s}_i \,/\, \|{c^s}_i\|_2$
5.	Decision Tree	Generate a decision tree from the training tuples of the data partition D	Training set D -> O(n X |D| X log |D|) Where, N = Number of attributes |D| = Number of training tuples in D	*Data Partition, D, which is set of training tuples and their associated class labels. * attribute list, set of candidate attributes. *Attribute selection methods, a procedure to determine the splitting criterion.
6.	Neural Networks	M-way	Depends on the selection of learning rate If the learning rate is too small, then learning will occur at very slow pace. If the learning rate is too large, then oscillations between inadequate solution may occur. Thumb Rule: Set learning rate to 1/t, where t is the number of iterations through training set.	$I_j = \sum_i w_{ij} O_j + \theta_j$ Which compute the net input of unit j with respect to th previous layer i. $O_j = \frac{1}{1+ e^{-Ij}}$

Table1. Comparison of various Classifiers [1][[2]

$Y= f(t_d) = b_0+\sum_{j=1} b_j t_{dj}$…….(3) [1]
A new document is classified to positive class if $f(t_d)>0$ otherwise negative class.

E. *Neural Network Classifier*: [1] [2]

For classifying a new document by using NN, its term weights are loaded into the input units. Then these units propagate through the network and the resulting output units generate the categories. Back propagation is a one way to train the classifier.

From the all above the mentioned Classifiers, no single classifier can be recommended as a general model. Each algorithm performs differently depending on data collections. None of them can be said globally superior over the other.[1]
However, to the certain extent SVM with Naïve Bayes Classifier is said to perform well. [2][1] In spite of the design of the Naïve Bayes and its simplified assumptions, Naïve Bayes Classifier performs well for different type of the data collections as compare to the other techniques. Naive Bayes algorithm is proved to be the best for numerical and text data. It is also easy and computationally cheap when compared with other techniques of classificationas shown in Table 1.

III NAÏVE BAYES CLASSIFIER

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem (from Bayesian statistics) with strong (naive) independence assumptions.[3] An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters necessary for classification.
Assumption: A Naive Bayes Classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature.

A.Models of Naïve Bayes Classifier:[4] [5]
1)Multivariate Bernoulli model: A document is represented by a binary feature vector, whose elements (1/0) indicate presence or absence of a particular word in a given document. In this case the document is considered to be the event and the presence and absence of words are considered as attributes of the event. [4]

2)Multinomial model: A document is represented by an integer feature vector, whose elements indicate frequency of corresponding word in the given document. Thus the individual word occurrence is considered to be events and document is considered to be collection of word events. Multinomial model is more accurate than the multivariate Bernoulli model for many classification tasks because it considers the frequency of the words too. [4][5]

B. Probabilistic Model:[3]
Consider D be the set of documents and C be the set of classes. The probability of assigning a document d to a class c is given by:

$C_{NB} = \text{argmax}_{cj \in c} P(c|d) = \text{argmax}_{cj \in c} \frac{P(c)P(d|c)}{P(d)}$.....(4)

As P(d) is independent of the class, it can be ignored.

$C_{NB} = \text{argmax}_{cj \in c} P(c)\ P(d|c)$……(5)

According to Naïve Bayes assumption,

$P(d|c) = P(w_1|c)\ P(w_2|c)..\ P(w_d|c) = \prod_{1<k<d} P(wk|c)$.....(6)

Replacing (5) by

$C_{NB} = \text{argmax}_{cj \in c} P(c)\ \prod_{1<k<d} P(wk|c)$…….(7)

Where P(c) is the prior probability of the class c_j, which is calculated as $\frac{N}{n}$, where N is the total number of training documents in class c, n is the total number of training documents. P(c|d) is the posterior probability.

$P(w_k|c) = \frac{T}{\sum_{t' \in V} T'}$ ……..……(8)[5]

Where T is the number of occurrences of w in d from classs c, $\sum_{t' \in V} T'$ is the total number of words in d from class c. [5][4]

IV SMOOTHING METHODS

It refers to the adjustment of maximum likelihood estimator for the language model so that it will be more accurate. At the very first, it is not required to assign the zero value to the unseen word. It plays two important roles: 1) Improves the accuracy of the language model. 2) Accommodate the generation of common and non informative words.
General Model:
The maximum likelihood generator generally under estimate the probability of unseen words. So the main purpose of the smoothing is to provide a non-zero probability to unseen words and improve the accuracy of probability estimator. The general form of smoothed model is of the form:

$$P(w|d) = \begin{cases} Ps(w|d) & \text{if w is seen} \\ \alpha d\, P(w|c) & \text{otherwise} \end{cases}$$

Where $P_s(w \mid d)$ is the smoothed probability word seen in the document and $P(w \mid d)$ is the collection language model and α_d is the coefficient controlling the probability assigned to unseen words so that probabilities sum to one.
Generally, Smoothing methods differ in choice of $P_s(w \mid d)$. A Smoothing method can be as simple as adding extra count or more complex where words of different count are treated differently.
1.) Jelinek-Mercer method: This method involves a linear interpolation of the maximum likelihood model with the collection model using a coefficient ʎ.[6] [7] [8]

$P_\lambda(w|d) = (1-\lambda)\ P_{ml}(w|d) + \lambda\ P(w|c)$……(9)

2.) Using Dirichlet Priors: A language model is a multi-nominal distribution, for which the conjugate prior for the Bayesian analysis is the Dirichlet distribution with parameters

(µp(w1|c), µp(w2|c), µp(w3|c),….., µp(w1|c))

Thus, model is given by: [6] [7] [8]

$P_\mu(w|d) = \frac{count(w,d) + P(w|c)}{\sum_w count(w.d) + \mu}$……...……...……..(10)

3.) Absolute Discounting: It lowers the probability of seen words by subtracting a constant from their counts. It is similar to JK method but differs in that it discounts the probability by subtracting instead of multiplying.

$$P_\delta(w|d) = \frac{\max(count(w,d)-\delta,0)}{\sum_w count(w,d)} + \sigma\, P(w|c) \text{ …….....(11)}$$

Where δ is a discount constant and $\sigma = \delta\, |d|_u / |d|$, so that it equals to one. Here, $|d|_u$ is the number of unique terms in d and |d| are the total number of terms. [6] [7] [8]

4.)Two-Stage Smoothing: It combines the Dirichlet Smoothing with the Interpolation method as; [6][7][8]

$$P_{TS}(w|c_i) = (1 - \lambda)\, \frac{count(w,c)+\mu P(w|c)}{|ci|+\mu} + \lambda\, P(w|c) \text{……...(12)}$$

Name	***Method***	***Para meter***
JM Smoothing	$P_\lambda(w|d) = (1-\lambda)P_{ml}(w|d) + \lambda\, P(w|c)$	λ
Dirichlet Smoothing	$P_\mu(w|d) = \frac{count(w,d)+P(w|c)}{\sum_w count(w.d)+\mu}$	μ
Absolute Discounting	$P_\delta(w|d)=\frac{\max(count(w,d)-\delta,0)}{\sum_w count(w,d)}+\sigma P(w|c)$	δ
Two-Stage Smoothing	$P_{TS}(w|c_i) = (1 - \lambda)\, \frac{count(w,c)+\mu P(w|c)}{|ci|+\mu} + \lambda\, P(w|c)$	λ and μ

Table 2. Summary of Smoothing Techniques [6]

Laplace Smoothing is replaced by various sophisticated smoothing methods like JK Smoothing, Dirichlet Smoothing, Two-Stage Smoothing, and Absolute Discounting. By applying the various Smoothing techniques, the performance of the Naïve Bayes has been increased. Dirichlet Smoothing method performed well than other methods. JM performs well mostly in case long verbose Queries instead of precise ones. Dirichlet is the most efficient type of smoothing. Absolute discounting performs well in case of short term documents. [6]

CONCLUSION

Text classification has become a major issue, now a days and one reason of it is the lack of single technique, which is able to produce good classification for different da sets. There are various classification methods such as Decision Trees, Neural Networks, Naïve Bayes and Centroid Based, but Naïve Bayes performs better for different data collections and is easy and computationally cheap. Along with its simplicity, Naïve Bayes also suffers from the some issues like unseen words. So, we use various smoothing techniques like JK method, Absolute Discounting method, Dirichlet Smoothing and Two-stage Smoothing to enhance the performance and accuracy of Naïve Bayes. We conclude that two-stage smoothing performs well with NB.

References

[1] B S Harish, D S Guru and S Manjunath," Representation and Classification of Text Documents: A Brief Review", *IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition, 2010.*

[2] Y. H. LI and A. K. JAIN, "Classification of Text Documents", *The Computer Journal*,1998.

[3] Kevin P. Murphy, " Naïve Bayes classifier", Department of Computer Science, University of British Columbia,2006.

[4] Hetal Doshi and Maruti Zalte, "Performance of Naïve Bayes Classifier-Multinomial model on different categories of documents" National Conference on Emerging Trends in Computer Science and Infortiom Technology,IJCA,2011.

[5] Ajay S. Patil and B.V. Pawar, "Automated Classification of Naïve Bayesian Algorithm" ,Proceedings of International Multi-Conference of Engineers and Computer Scientists, March 14-16,2012.

[6] C. Zhai and J. Lafferty, "A Study of Smoothing Methods for language Models Applied to Information Retrieval" TOIS, 22:179 – 214, 2004.

[7] Jing Bai and Jian-Yun Nie. "Using Language Models for Text Classification", InAIRS, 2004.

[8] Quan Yuan , Gao Cong and Nadia M. Thalmann,"Enhancing Naïve Bayes with Various Smoothing Method for Short text Classification", Proceedings of 21st International Conference on World Wide Web, pages 645-646,2012.

[9] Colas, Fabrice, and Pavel Brazdil. "Comparison of SVM and some older classification algorithms in text classification tasks." In *Artificial Intelligence in Theory and Practice*, pp. 169-178. Springer US, 2006.