A Comparison of Text Classifiers For Sentimental Analysis

Mrs. Padmini C[1], Dr. Vani Priya C H[2]
[1, 2] Sir M. Visvesvaraya Institute of Technology, Bangalore, Karnataka, India.

Abstract- *Now a days sentimental analysis is active field of research, to extract people's opinion about a particular product or service. The task of sentiment classification is to classify reviews of users as positive or negative from textual information. In recent years we are provided with many data mining classification techniques such as Naïve Bayes classifier with strong independence assumption. But they lack in accuracy for many complex real world situations where there exists dependency among features. So we try to use the probabilistic classifiers Gaussian, Bernoulli and Multinomial which makes the Naïve bases assumption to improve the accuracy. The other classifier Support Vector Machine is a non-probabilistic binary linear classifier which is also experimented with sentimental classification to obtain a higher accuracy than the Naïve Bayes origin. The paper aims to determine the efficiency of classifiers to sort the list of data set into its appropriate sentiment. This is done by comparing the featured data sets classified by each of these data mining classifiers.*

Keywords- sentimental analysis, sentimental classification, Naïve Bayes classifier, Gaussian, Bernoulli, Multinomial, Support Vector Machine.

I. INTRODUCTION

Language is a powerful tool to communicate and convey information. It is a means to express emotion and sentiment. Sentiment analysis is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes and emotions towards entities such as products, individuals, issues, events, movies and topic. It uses natural language processing and data mining to extract opinions from text. In the recent years, the exponential increase in the Internet usage and exchange of public opinion is the driving force behind sentiment analysis [1]. The most useful application of sentimental analysis is the sentimental classification of product reviews and movie reviews. The sentimental classification can be categorized into positive and negative. Positive returns are good and negative returns are bad review [2].

II. BACKGROUND AND LITERATURE REVIEW

Sentimental analysis can be performed at three different levels: document, sentence and aspect level. The document level sentiment analysis aims at classifying the entire document as positive or negative [3], [4].The sentence level sentiment analysis is closely related to subjectivity analysis. At this level each sentence is analyzed and its opinion is determined as positive, negative or neutral [5-8]. The aspect level sentiment analysis aims at identifying the target of the opinion. The basis of this approach is that every opinion has a target and an opinion without a target is of limited use [9].

Our research is based on aspect level sentiment analysis where we review a summarized list of 3000 words from movie review and categorize them into positive or negative words using the machine learning classifiers. To do this we train the 1900 words and test it on the remaining 1100 words. Sentiment categorization is essentially a classification, where features that contain opinions or sentiment information should be identified before the classification [10].

Let's consider an illustration of how text classifiers are implemented. Consider a set of objects from the data set. The objects can classified as either GREEN or RED. Our task is to classify new cases when they arrive and decide as to which group they belong.

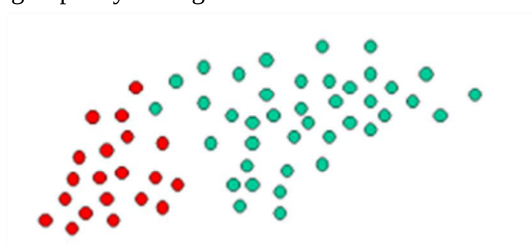

Figure 1. Objects classified to GREEN or RED

We can calculate the prior probabilities of the objects among all objects. Hence

$$\text{probability for GREEN} \propto \frac{\text{Number of GREEN objects}}{\text{Total number of objects}} \quad (1)$$

$$\text{probability for RED} \propto \frac{\text{Number of RED objects}}{\text{Total number of objects}} \quad (2)$$

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our probabilities for class membership are:

$$\text{probability for GREEN} \propto \frac{40}{60} \quad (3)$$

$$\text{probability for RED} \propto \frac{20}{60} \quad (4)$$

Since we have calculated the probability, we are ready to classify a new object (WHITE circle in Figure 2).

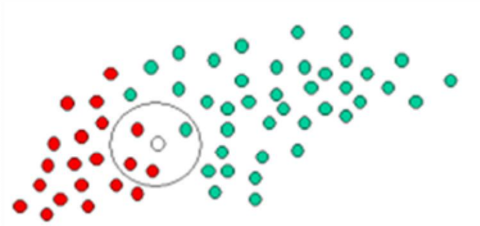

Figure 2. Classify the WHITE circle

Let's consider a circle X encompassing a number of points including the new object irrespective of their class label. Then we calculate the number of points in the circle belonging to each label.

$$\text{Likelihood of X if GREEN} \propto \frac{\text{Number of GREEN inside X}}{\text{Total number of GREEN cases}} \quad (5)$$

$$\text{Likelihood of X if RED} \propto \frac{\text{Number of RED inside X}}{\text{Total number of RED cases}} \quad (6)$$

In Figure 2, it is clear that Likelihood of X if RED is larger than Likelihood of X if GREEN, since the circle encloses 1 GREEN object and 3 RED ones. Hence:

$$\text{Likelihood of X if GREEN} \propto \frac{1}{40} \quad (7)$$

$$\text{Likelihood of X if RED} \propto \frac{3}{20} \quad (8)$$

Although the prior probabilities indicate that X may belong to GREEN, the likelihood indicates otherwise. The final classification is produced by combining both sources of information (i.e. the prior probability and the likelihood) to form a posterior probability using Bayes Rule.

$$\text{Posterior probability of X being GREEN} \propto \text{probaility of GREEN} \times \text{Likelihood of X if GREEN} \quad (9)$$

$$= \frac{4}{6} \times \frac{1}{40} = \frac{1}{60}$$

$$\text{Posterior probability of X being RED} \propto \text{probaility of RED} \times \text{Likelihood of X if RED} \quad (10)$$

$$= \frac{2}{6} \times \frac{3}{20} = \frac{1}{20}$$

Finally, we classify X as RED since its class membership achieves the largest posterior probability.

III. METHODOLOGY

A modern approach towards sentiment classification is to use machine learning techniques which trains several list of words. Popular data mining methods include Naïve Bayes,Gaussian, Bernoulli, Multinomial of Naïve Bayes assumption and Support Vector Machine which are used for classification.

A. Naïve Bayes Classifier

In machine learning, naïve Bayes classifiers are a family of simple probabilistic classifiers. They are based on applying Bayes' theorem. Naïve Bayes has been studied since the 1950s. It was gained importance in the 1960 for the document classification as spam or legitimate [11]. It is a popular method for text categorization; it determines the problem of judging documents as belonging to one category or the other with word frequencies as the features. Apart from this it also has a numerous application in the field of mathematics, computer science etc [12].As a classifier it is easy and fast to implement. The naïve Bayes classifier is the simplest of other models, in that it assumes that all attributes of the examples are independent of each other and due to this it performs classifications very well [13].

This technique assumes that the presence or absence of any feature in the document is independent of the presence or absence of any other feature. Naïve Bayes classifier considers a document as a bag of words and assumes that the probability of a word in the document is independent of its position in the document and the presence of other word.

The Bayes rule is

$$P(c|t) = P(c)^* P(t|c) / P(t) \quad (11)$$

Where, c is a specific class and t is text to classify. P(c) and P(t) is the prior probabilities of this class and this text. P(t|c) is the probability the text appears given this class. The value of class c might be POSITIVE or NEGATIVE, and t is just a text[14].

B. Gaussian Naïve Bayesian

In this technique, the continuous values associated with each class are distributed according to a Gaussian distribution. The actual problem with the Naïve Bayes Classifier is that it assumes all attributes are independent of each other which in general cannot be applied. Gaussian PDF can plug-in here to estimate the attribute probability density function (PDF) [15]. Because of the well developed Gaussian PDF theories, we can easily classify new object through the same Bayesian Classifier Model. Normally this gives more accurate classification result.

C. Multinomial Naïve Bayesian

Multinomial Naïve Bayes (MNB) has been widely used in text classification. Given a set of labeled data, it often uses a parameter learning method called Frequency Estimate (FE), which estimates word probabilities by computing appropriate frequencies from data. The major advantage of FE is that it is simple to implement, often provides reasonable prediction, and is efficient [16].

It models the distribution of words in a document as a multinomial. A document is treated as a sequence of words and it is assumed that each word position is generated independently of every other. For classification, weassume that there are a fixed number of classes, $C \in \{1, 2, . . . , m\}$ each with a fixedset of multinomial parameters. The parameter vector for a class C is $\vec{\theta}_c = \{\theta_{c1}, \theta_{c2}, . . . , \theta_{cn}\}$ where n is the size of the vocabulary, $\sum_i \theta_{ci} = 1$ and θ_{ci} is the probability is the probability that word i occurs in that class. The likelihood of a document is a product of the parameters of the words that appear in the document [17],

$$p(d|\vec{\theta}_c) = \frac{(\Sigma_i f_i)!}{\Pi_i f_i!} \prod_i (\theta_{ci})^{f_i} \quad (12)$$

where f_i is the frequency count of word i in document d. By assigning a prior distribution over the set of classes, $p(\vec{\theta}_c)$, we can arrive at the minimum error classification rule which selects the class with the largest posterior probability,

$$l(d) = argmax_c \left[\log p(\vec{\theta}_c) + \textstyle\sum_i f_i \log \theta_{ci}\right] \quad (13)$$

$$= argmax_c [b_c + \textstyle\sum_i f_i w_{ci}] \quad (14)$$

Where bc is the threshold term and w_{ci} is the class c weight for word i. These values are natural parameters for the decision boundary [18]. This is especially easy to see for binary classification, where the boundary is defined by setting the differences between the positive and negative class parameters equal to zero,

$$(b_+ - b_-) + \textstyle\sum_i f_i (w_{+i} - w_{-i}) = 0 \quad (15)$$

D. Multi-variant 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 cast the document is considered to be the event and presence and absence of words are considered as attributes of the event [19] [20].

Given a vocabulary V, each dimension of the space t, $t \in \{1, . . . , |V|\}$, corresponds to word wt from the vocabulary. Dimension t of the vector for document di is written Bit ,and it is either 0 or 1, indicating whether word wt occurs at least once in the document. Withsuch a document representation, we make the naïve Bayes assumption : that probability of each word occurring in a document is independent of the occurrence of other words in a document [21]. Thus, the probability of a document given its class is product of the attribute values over all word attributes:

$$P(d_i|c_j;\theta) = \prod_{t=1}^{|V|} (B_{it} P(w_t|c_j;\theta) + (1 - B_{it})(1 - P(w_t|c_j;\theta)) \quad (16)$$

Thus given a generating component, a document can be seen as a collection of multiple independent Bernoulli experiments, one for each word in the vocabulary, with the probabilities for each of these word events defined by each component, $P(w_t|c_j;\theta)$. This is equivalent to viewing the distribution over documents as being described by a Bayesian network, where the absence or presence of each word is dependent only on the class of the document.

E. Support Vector Machine model

Another algorithm for solving the text classification problem is Support Vector Machine (SVM) introduced by [22]. The idea of this algorithm is to consider each document as a point in the document space and to find the appropriate hyper plane which separates them. The x and y are the co-ordinates of two dimensional spaces [23].

However, text classification problem involves with not only two classes but also multiples classes. So the algorithm needs to be extended. There are several works done with extension of SVM [24]. Two simple approaches are:

- One against all: assume that there are only two classes, one class v/s other classes
- Pair wise classification: one class against one other class and aggregate the results.
-

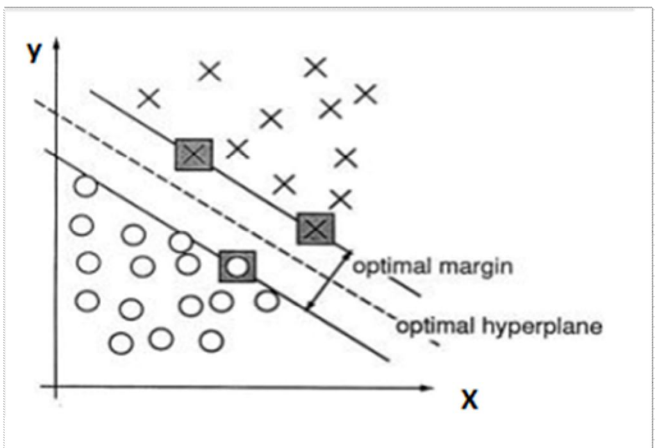

Figure 3. Support Vector Machine Illustration

Implementation

The comparison of the classifiers can be implemented using the following steps:

Initially we need to get retrieve the data set. These data sets are actually the words from the many movie reviews.

Out of these reviews frequently repeated words are obtained and stored them as featured words. The featured words can be up to 3000 words.

These feature words are shuffled using a randomize function on to the data set.

These featured words are divided into two parts. The first 1900 words are trained using all the mentioned classifiers.

The second 1100 words can be used to test the trained data sets of each classifier and to determine the accuracy of the classifier.

- The algorithm is designed in such a way that the data is trained and tested for a finite no of loop and in each loop maximum accuracy of each classifier is found out.
- At the end of the final loop the most accurate classifier for sentimental text classification can be found out.

IV. RESULTS AND ANALYSIS

After the training of the classifiers it is ready for the test.

We run different algorithms on the same data set for hundred times and get the following result (taking 10 random samples). We find the comparison between probabilistic models and represent these comparisons in ten 10 loops.

Number of scans	Original Naïve Bayes (in %)	Multinomial naïve Bayes (in %)	Bernoulli classifier (in %)
1	59	67	61
2	69	72	76
3	72	72	75
4	60	69	59
5	64	67	64
6	73	72	71
7	72	78	76
8	64	63	63
9	63	64	62
10	71	75	72

Figure 4. Comparison of Probabilistic Models

Using these compared values from the table we find the percentage of efficiency of these probabilistic models and also find out the average efficiency with the following ten samples.

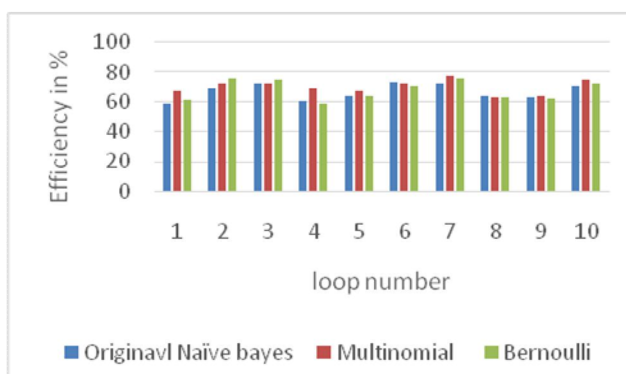

Figure 5. Graphical representation of the comparison

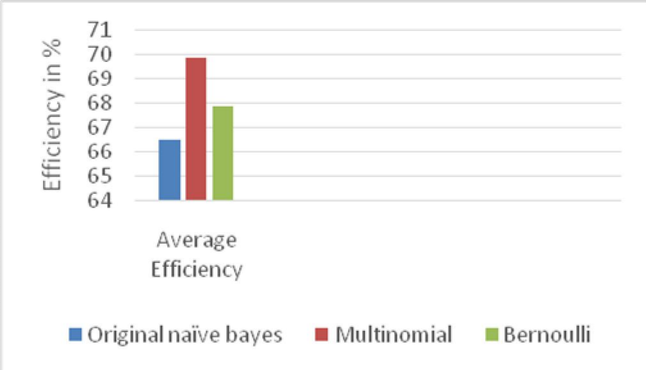

Figure 6.Graphical Representation of efficiencies of Probabilistic Models

Taking those ten samples we also find the comparison between Probabilistic and Non-probabilistic models and similarly the average efficiencies.

Number of scans	Original Naïve Bayes (in %)	Multinomial naïve Bayes (in %)	Non Probabilistic model (in %)
1	69	72	76
2	72	72	75
3	70	68	71
4	64	67	64
5	71	74	78
6	75	74	76
7	72	70	71
8	70	70	73
9	76	75	77
10	70	73	73

Figure 7. Comparison between Probabilistic and Non-Probabilistic Models

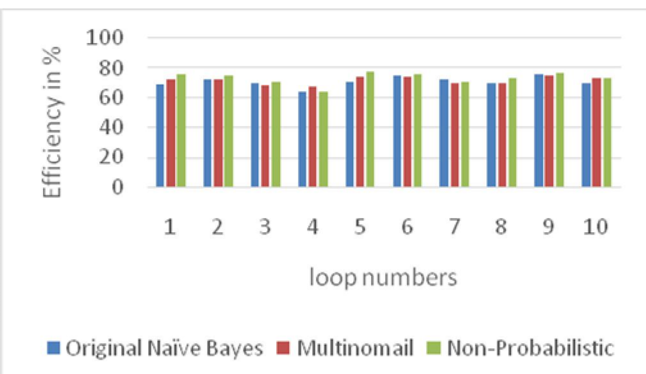

Figure 8. Graphical representation of the comparison

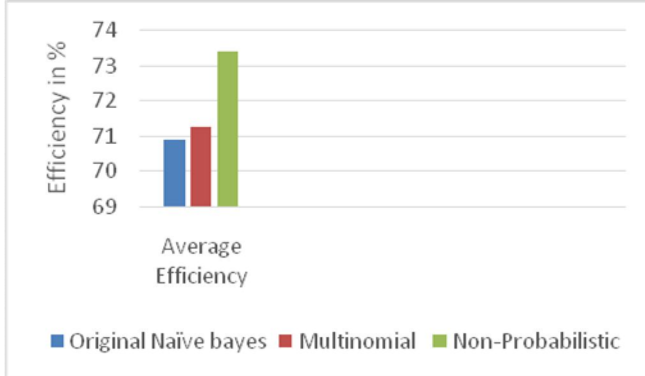

Figure 6. Graphical Representation of Efficiencies between Probabilistic & Non-Probabilistic Models

Thus we can see that we get the SVC and the Multinomial algorithm as the most accurate once.

Conclusion and Future Work

Thus we would like to conclude that only pure naïve Bayes algorithm is not as effective to classify text. We either need to use the Multinomial variant or the non-probabilistic algorithm for better classification.The multinomial model uses the frequency estimation model and eliminates the unwanted word and comes up with the most used words.

But the more accurate and commonly used industrial classifier is the support vector machine. They can be used to classify the text which are unlabeled as well.The non-probabilistic approach is more compact and doesn't take any random input. This is much closer to the real time applications used.

Thus as the area of the text classifiers are expanding the non-probabilistic algorithms are growing more popular and accurate.

In future we are going to use the following techniques we can determine the most efficient classifier and use it for classifying real time online reviews.

REFERENCES

[1] Lina L. Dhande and Dr. Prof. Girish K.Patnaik," Analyzing Sentiment of Movie Review Data using Naïve Bayes Neural Classifier", International Journal of Emerging Trends &Technology in Computer Science, vol.3, no.4, pp.313-320, July-August 2014.

[2] N. M. Shelke, S. Deshpande, V. Thakre, "Survey of techniques for opinion mining," International Journal of Computer Applications, vol.57, no.13, pp. 30-35, November 2012.

[3] B. Seerat and F. Azam, “Opinion mining: Issues and challenges (a survey),” International Journal of Computer Applications, vol. 49, no. 9, pp. 42-51, July 2012.

[4] P. Routray, C. K. Swain, S. P. Mishra, “A survey on sentiment analysis,” International Journal of Computer Applications, vol. 76,no. 10, pp. 1-8, August 2013.

[5] Chesley P, Vincent B, Xu L, Srihari RK (2006) Using verbs and adjectives to automatically classify blog sentiment. Training580(263): 233.

[6] Choi Y, Cardie C (2009) Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, EMNLP ’09, 590–598.. Association for Computational Linguistics, Stroudsburg, PA, USA.

[7] Jiang L, Yu M, Zhou M, Liu X, Zhao T (2011) Target-dependent twitter sentiment classification In: Proceedings of the 49th, Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, 151–160.. Association for Computational Linguistics, Stroudsburg, PA, USA.

[8] Tan LK-W, Na J-C, Theng Y-L, Chang K (2011) Sentence-level sentiment polarity classification using a linguistic approach In: Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation, 77–87.. , Springer, Heidelberg, Germany.

[9] Liu B (2012) Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

[10] Sunny Dahiya,” sentiment analysis using svm and naïve bayes algorithm”,vol.4,no.9,pp.212-219,September 2015.

[11] Russell Stuart, Norvig, Peter (2003) [1995]. Artificial Intelligence: A modern Approach (2nd ed.). Prentice Hall. ISBN 978-0137903955.

[12] Hand, D.J,Yu, K. (2001). “Idiot’s Bayes – not so stupid after all?”.International Statistical Review 69(3): 385-399.

[13] Xing Fangand Justin Zhan,”Sentiment analysis using product review data”,Journal of Big Data 2015,2:5,DOI: 10.1186/s40537-015-0015-2

[14] Lina L. Dhande , Dr. Girish K. Patnaik, ,” Review of Sentiment Analysis using Naive Bayes and Neural Network Classifier”, International Journey of Scientific Engineering and Technology Research,vol.3,no.7,pp.1110-13,May 2014

[15] Ho, T. J.: “Data Mining and Data Warehousing”, Prentice Hall, 2005.

[16] Jiang Su, Jelber Sayyad-Shirabad and Stan Matwin,” Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes,” The 28th International Conference on Machine Learning,2011.

[17] R. O., & Hart, P. E. (1973).Pattern classification and scene analysis. Wiley and Sons, Inc.

[18] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger, “Tackling the Poor Assumptions of Naive Bayes Text Classifiers,” In: Proceedings of the Twentieth International Conference on Machine Learning, AAAI Press (2003) 616-623.

[19] 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 Information Technology,IJCA,2011.

[20] 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.

[21] Shruti Aggarwal, Devinder Kaur,” Naïve Bayes Classifier with Various Smoothing Techniques for Text Documents,” International Journal of Computer Trends and Technology,vol.4, no.4,pp.873-876.

[22] Cortes, C., & Vapnik, V. “Support-Vector Networks”. Machine Learning, 1995, pp. 273-297.

[23] Hsu, C.-W., & Lin, C.-J. A comparison of methods for multiclass support vector machines. IEEE TRANS.NEURAL NETWORKS, 2002, pp. 415-425.

[24] Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). Target-dependent Twitter Sentiment Classification. Computational Linguistics,151-160.

[25] AsthaChharia, R.K.Gupta, Enhancing Naïve Bayes Performance with Modified Absolute Discount Smoothing Method in Spam Classification. International Journal of Advanced Research in Computer Science and Software Engineering, vol.3, no 3, pp. 424-29, March-2013.

[26] D. Vilar, H. Ney, A. Juan, and E. Vidal, “Effect of Feature Smoothing Methods in Text Classification Tasks”. In International Workshop on Pattern Recognition in Information Systems, pages 108-117. Porto, Portugal, 2004.

[27] Y. Yang, and J.O. Pedersen, “A comparative study on feature selection in text categorization”. In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US, MorganKaufmann Publishers, SanFrancisco, 412–420, 1997.