ISSN:0975-9646

Comparative Study of Classification Algorithms used in Sentiment Analysis

Amit Gupte, Sourabh Joshi, Pratik Gadgul, Akshay Kadam

Department of Computer Engineering, P.E.S Modern College of Engineering
Shivajinagar, Pune

***Abstract*—The field of information extraction and retrieval has grown exponentially in the last decade. Sentiment analysis is a task in which you identify the polarity of given text using text processing and classification. There are various approaches in the task of classification of text into various classes. Use of particular algorithms depends on the kind of input provided. Analyzing and understanding when to use which algorithm is an important aspect and can help in improving accuracy of results.**

***Keywords*— Sentiment Analysis, Classification Algorithms, Naïve Bayes, Max Entropy, Boosted Trees, Random Forest.**

I. Introduction

In this paper we have presented a comparative study of most commonly used algorithms for sentimental analysis. The task of classification is a very vital task in any system that performs sentiment analysis. We present a study of algorithms viz. 1. Naïve Bayes 2.Max Entropy 3.Boosted Trees and 4. Random Forest Algorithms. We showcase the basic theory behind the algorithms, when they are generally used and their pros and cons etc. The reason behind selecting only the above mentioned algorithms is the extensive use in various tasks of sentiment analysis. Sentiment analysis of reviews is very common application, the method of taking reviews has evolved over a period of time. The scope of expressing a person's thoughts is often restricted when people have to give reviews about a product in form of score / star ratings. But when a person is allowed to express reviews in form of open text he can be very precise about what aspects about the product are good and what are not. Sentiment analysis engines parse through this textual reviews and generate output in form of polarities i.e. – Positive, Negative or Neutral. This helps in finding the reasons behind crucial fluctuations in sales of products and they can be rectified accordingly.

The algorithms in classifications influence the accuracy of the outcome in polarity and hence a fault in classification can result in an ultimate outcome of a flawed business monitoring strategy [1].

A. Sentiment Analysis

Sentiment analysis is a task that involves information extraction from customer feedback and other authentic sources like survey agencies. As the word suggests it includes detecting sentiments of any individual from the text that is writes in digital format. There are a wide array of applications of this concept. This concept became centre of attention since industry got revolutionized with the change in paradigm of "Sellers' Market" to "Buyers' Market" in order to capture market share.

Major steps in Sentiment analysis are

- Text Extraction – This step involves extracting words from text that influence the outcome of the result.
- Text Refinement – This step involves refining text in form of relevant phrases, words etc.
- Text Classification – This step includes classification of text into its class (positive/negative)
- Score Aggregation – This step collects total scores from classifier and then aggregates it further to produce the total sentiment score [2].

II. Classification

Classification is a stage in sentiment analysis that can described as a process in which we predict qualitative response, or in this case we classify the document into its polarity. Predicting a qualitative response of a document can be referred to as classifying the document since it involves since it involves assigning an observation to a category or class. There are many possible classification techniques, or classifiers that one might use for to predict the qualitative response or class of a document. In sentiment analysis some widely used classification techniques are as follows:

- Naïve Bayes Classifier
- Max Entropy Classifier
- Boosted Trees Classifier
- Random Forest Classifier

In this paper we have showcased a comparative study of the above stated classification algorithms that are widely used in sentiment classification [3].

III. Naïve Bayes Classifier

Naïve Bayes classifier is based on Bayes theorem. It's a baseline classification algorithm. Naïve Bayes classifier assumes that the classes for classification are independent. Though this is rarely true Bayesian classification has shown that there are some theoretical reasons for this apparent unreasonable efficiency. There are various proofs that show that even though the probability estimates of Naïve Bayes classification are low it delivers quite good results in real life examples. Naïve Bayes just over estimates the class that certain object belongs too. Assuming that we are using it only for making decisions (which is true in case of sentiment analysis problem) the decision making is correct and the model is useful [4].

In Text classification we tokenize the document in order to classify it in its appropriate class. By using the "Max

Posterior Probability" Decision rule we get the following classifier:

$$c_{map} = \arg\max_{c\in C}\left(P(c|d)\right) = \arg\max_{c\in C}\left(P(c)\prod_{1\le k\le n_d} P(t_k|c)\right)$$

In the above equation t_k are the tokens / words in the document, C is the set of classes used in classification, P(c|d) is the conditional probability of class c given the document d, P(c) is the prior probability of class C and P(t_k |C) is the conditional probability of token t_k given class C. This means that in order to find in which class we should classify a new document, we must estimate the product of the probability of each word of the document given a particular class (likelihood), multiplied by the probability of the particular class (prior). After calculating the above for all the classes of set C, we select the one with the highest probability.

Naïve Bayes is used as a classifier in various real world problems like Sentiment analysis, email Spam Detection, email Auto Grouping, email sorting by priority, Document Categorization and Sexually explicit content detection. The major advantage of Naïve Bayes is it requires low processing memory and less time for execution. It's advised that this classifier should be used when Training time is a crucial factor in the system. Naïve Bayes is the baseline algorithm for researches in decision level classification problem. In presence of limited resources in terms of CPU and Memory Naïve Bayes is recommended classifier. This algorithm is tweaked a lot by researchers to match the system requirement.

There are several variations in Naïve Bayes classifier:

- Multinomial Naïve Bayes - Used when Multiple Occurrences of Word Matter a lot in Text Classification problems. Such an example is when we try topic classification.
- Binarized Multinomial Naïve Bayes - Used when frequencies of the words don't pay a key role in our classification. Such an example is Sentiment analysis where it doesn't matter how many times someone enters the word 'bad' or 'good' but rather only the fact that he does
- Bernoulli Naïve Bayes - This is used when in our problem the absence of a particular word matters For example Bernoulli is commonly used in Spam or Adult Content Detection with very good results.

Even though that this fact is well accepted this algorithm is outperformed by other classifiers like Max Entropy , Boosted Tress , Random Forest etc. the simplicity of Naïve Bayes and the efficiency that it provides in less amount of resources makes it a classifier that must be considered in Sentiment Analysis[5,6].

IV. Max Entropy Classifier

Another well-known classifier is the Max Entropy Classifier or MaxEnt as some people prefer to call it. The idea behind MaxEnt classifiers is that we should prefer the most uniform models that satisfy any given constraint. MaxEnt models are feature based models. We use these features to find a distribution over the different classes using logistic regression. The probability of a particular data point belonging to a particular class is calculated as follows:

$$p(c \mid d, \vec{\lambda}) = \frac{\exp\left[\sum_i \lambda_i f_i(c,d)\right]}{\sum_{c'} \exp\left[\sum_i \lambda_i f_i(c',d)\right]}$$

Where, c is the class, d is the data point we are looking at, and λ is a weight vector. MaxEnt makes no independence assumptions for its features, unlike Naïve Bayes. This means we can add features like bigrams and phrases to MaxEnt without worrying about feature overlapping. The principle of maximum entropy is useful explicitly only when applied to *testable information*. A piece of information is testable if it can be determined whether a given distribution is consistent with it. For example, the statements - The expectation of the variable *x* is 2.87 and $p_2 + p_3 > 0.6$ are statements of testable information. Given testable information, the maximum entropy procedure consists of seeking the probability distribution which maximizes information entropy (This is the average amount of data that one data set will contain.), subject to the constraints of the information. Entropy maximization with no testable information takes place under a single constraint: the sum of the probabilities must be one. Under this constraint, the maximum entropy discrete probability distribution is the uniform distribution. [7]

$$p_i = \frac{1}{n} \text{ for all } i \in \{1, \ldots, n\}.$$

Various results that we came across in our study exclusively mentioned that Naïve Bayes theorem has very less efficiency then a simple Max Entropy Algorithm. Our research revealed a variation in Max Entropy knows as Max Entropy using Priors which enhances the efficiency of MaxEnt classifier by using Prior Results as a part of training Data set. MaxEnt using Priors is more effective in Natural language processing applications. The Major Advantages of using MaxEnt or its variations can be listed out as follows:

- Accuracy
- Consistency – This algorithm shows consistency in results and if priors are used results also improve over a period of time.
- Performance / Efficiency - Can handle huge amounts of data
- Flexibility - The algorithm is flexible of having many different typed of data in a unified platform and classify it accordingly [8].

V. Boosted Trees Classifier

Boosted trees is a classifier that is basically a combination of Boosting and Decision Trees. Boosting is a machine Meta learning algorithm for reducing preconception in supervised learning. In Boosting predictive classifiers are used to develop weighted trees which are further combined into single prediction models. Boosted trees combine the strengths of two algorithms:

regression trees (models that relate a response to their predictors by recursive binary splits) and boosting (an adaptive method for combining many simple models to give improved predictive performance). Most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learner's accuracy. After a weak learner is added, the data is re–rated and new weights are produced and examples that are incorrectly classified gain weight and examples that are classified correctly lose weight (some boosting algorithms actually decrease the weight of repeatedly misclassified examples, e.g., boost by majority and Brown Boost) [10]. There are many variants of Boosting algorithms some of them are – Ada Boost, LP Boost, Total Boost, Logit Boost, Gradient Boosted Regression Trees etc. Boosting algorithms such as AdaBoost are known to perform well for classification and are very resistant to over fitting with respect to misclassification error, even though conditional class probability estimates eventually diverge to zero and one, implying complete over fit in terms of CCPF (Conditional Class Probabilities) estimation but not classification. Gradient boosting is a machine learning technique for regression problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function [11]. The gradient boosting method can also be used for classification problems by reducing them to regression with a suitable loss function. Certain advantages of Boosted trees classifier are - Fast Training without sacrificing accuracy, Can handle different types of predictor variables and accommodate missing data. On the contrary a major disadvantage is inability to compute conditional class probabilities [12].

VI. Random Forest Classifier

Random forests are an ensemble learning method for classification that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. It produces multi-altitude decision trees at inputting phase and output is generated in the form of multiple decision trees. The correlation between trees is reduced by randomly selecting trees and thus the prediction power increases and leads to increase in efficiency. The predictions are made by aggregating the predictions of various ensemble data sets. Studies show that the performance is seen always rising. There is no downtrend for the performance of this algorithm in any available data sets. Applications and real life of examples Random Forests are widespread. There is no single type for RF data sets. They can vary from any kind of applications like medical as well as general data sets. Decrease is less relevant data set to 50% affects the RF classifications in lowering the result accuracy. RF is a parallelized and multi-core friendly algorithm. So simultaneous running of different trees is also a support feature. The popularity of this machine increased with practical machine learning research and their related algorithms. We came across experimental results in our study in which people had used Random Forest for Opinion mining and have found impressive accuracy in classification of their data sets. The Major advantages of this algorithm can be listed out as follows:

- Easy to interpret and understand
- Non-parametric so you don't have to worry about the linearity of the input data set.
- If parameters are there they can be easily entered thus eliminating the need for pruning the trees.
- The classification model is fast and scalable
- Importance and relevance of text/tokens in a class is automatically generated
- Robust to irrelevant text present in document [13].

A disadvantage that we came across in our study is that the random forest classifier easily over fits its class. (This over fit can be solved. Since data sets have more number of trees and vague links or the data sets that you have provided are too small then the model over fits. To reduce the over fits reduce the number of trees in a random forest classifier as well as decrease the vague links that are present.)

VII. Differential Analysis

Table I

Features	**Naïve Bayes**	**Max Entropy**	**Boosted Trees**	**Random Forest**
Based On	Bayes Theorem	Feature Based Classifier	Decision Tree Learning	Decision Tree Aggregation
Simplicity	Very Simple	Hard	Moderate	Simple
Performance	Better	Good	Good	Excellent
Accuracy	Good	High	Poor	Excellent
Memory Requirement	Low	High	Low	High
Other Applications	Spam Detection, Document Classification, Sexually Explicit Content Detection	Diagnosis Tests in Pathology Labs	Classifying Cardiovascular Outcomes	Biomedical Applications, Sexually Explicit Content Detection
Result Accuracy Over a Period of Time	Variable	Consistent	Incremental	Incremental
Time Required For Training Classifier	Less	Moderate	High	Recurrent Learning with every novel dataset

Table I shows the net outcome of our comparative study based on various features. The results evidently state what algorithm should be selected depending upon what kind of classification scenario is under argument.

VIII. Conclusion

Study makes it pretty evident that every kind of classification model has its own benefits and drawbacks. Selection of classification models can be decided on the basis of resources, accuracy requirement, training time available etc.

Considering sentiment analysis the Random Forest classifier clearly has an upper hand with high accuracy and performance, simplicity in understanding, and improvement in results over a period of time. This makes the classifier best fit for situations like sentiment analysis. Though it requires high training time and processing power the improved accuracy due to aggregation of decision trees, more than makes up for other shortcomings. Random Forest is also very well supported in terms of implementation. Many libraries are available in programming languages like java, python and R which makes it easy to use as well.

We can conclude that if accuracy is at our highest priority then we must prefer a classifier model like Random Forest that consumes high learning time but has best accuracy. If processing power and memory is an issue then the Naïve Bayes classifier should be selected due to its low memory & processing power requirements. If less training time is available but you have powerful processing system and memory then Max Entropy proves to be a worthy alternative. If you need to select a classifier that is average on all aspects then Boosted Trees might be the right choice. Selection of a classification model should be dome wisely in sentiment analysis systems because this decision will influence the precision of your system and your end product.

ACKNOWLEDGMENT

We are using this opportunity to express our gratitude to Mrs. Manisha Petare & Mrs. Deipali Gore who have guided us regarding matters where we needed clarity about the subject. We are thankful for their aspiring guidance, invaluably constructive criticism and advice during the course of this study.

REFERENCES

[1] Bo Pang and Lillian Lee “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts” in *ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 2004, Article No. 271

[2] Sasha Blair- Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, Jeff Reynar. *Building a Sentiment Summarizer for local service Reviews, 2008*.

[3] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning: with Applications in R, Springer Texts in Statistics,* 2013.

[4] (2013) Machine Learning with Naïve Bayes Classifier [Online]. Available:http://blog.datumbox.com/machine-learning-tutorial-the-naive-bayes-text-classifier/

[5] Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis. Computational linguistics - Association for Computational Linguistics, 399-433.

[6] Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. Found. Trends Inf. Retr. 2, 1-2 (January 2008), 1-135.

[7] Y. M. C. S. Kostas Fragos, “A Weighted Maximum Entropy Language Model for Text Classification,” *Natural Language Understanding and Cognitive Science,* no. NCLUS May 2005.

[8] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[9] Kamal Nigam, John Lafferty ,Andrew McCallum,“ Using Maximum Entropy for Text Classification” *IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999, Pages 61-67 – Max Entropy*

[10] Jerome Friedman, Trevor Hastie, Robert Tibshirani, “Additive Logistic Regression: A statistical view of Boosting”, The Annals of Statistics 2000, Vol 28, No.2, 337-407.

[11] (2000)Gradient boosting. On Wikipedia the free encyclopedia. Available: http://en.wikipedia.org/wiki/Gradient_boosting

[12] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura “Sentiment Classification Using Word Sub-sequences and Dependency Sub-trees” Advances in Knowledge Discovery and Data Mining ,301-311, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings

[13] Rui Xia, Chengqing Zong ,Shoushan Li ,“ Ensemble of feature sets and classification algorithms for sentiment classification” *Information Sciences, Volume 181, Issue 6, 15 March 2011, Pages 1138–1152*