

Sentiment Analysis in Movie Reviews using Naive Bayes Algorithm

Final Project Report

Submitted by

Madhavi Polu

**For Big Data Applications and Analytics Class
Indiana State University Fall 2015**

Table of Contents

1. INTRODUCTION.....	2
2. TECHNOLOGY DESCRIPTION	3
2.1 MOVIE REVIEW	3
2.2 TEXT MINING AND SENTIMENT ANALYSIS.....	3
2.3 MACHINE LEARNING ALGORITHMS	3
2.3.1 <i>Naive Bayes Classifier</i>	3
2.3.2 <i>Support Vector Machines</i>	4
2.4 PYTHON PROGRAMMING LANGUAGE	4
3. PROJECT SOLUTION	5
3.1 DATA EXTRACTION	6
3.2 DATA PRE-PROCESSING.....	6
3.3 EXECUTING THE CLASSIFICATION	6
4. RESULTS.....	7
5. FUTURE IMPROVEMENTS.....	7
6. CONCLUSIONS	8
7. REFERENCES	8

ABSTRACT

Sentiment Analysis is a field that analyzes opinions or sentiments of people to detect polarity and recognize emotion from the text towards entities such as products and services. It has become an integral part of the product marketing and customer relationship management as both business world and consumer turn to online web resources for capturing opinion on products and services. Mining sentiments from natural language is challenging, because opinions in natural language can be expressed in complex ways containing ambiguity, idiom, sarcasm, and slang.

This project applies the Naive Bayes Classifier machine learning algorithm, to predict sentiment of movie reviews as positive or negative by understanding the meaning and relationship between the words. It is shown how Naive Bayes machine learning algorithms facilitates the sentiment analysis of movie reviews taken from a movie review website such as Rotten Tomatoes. This project also measures the performance of Naive Bayes classifier and compares performance of Naive Bayes classifier algorithm with Support Vector Machine (SVM) algorithm. Finally, this project recommends suitable approaches to take optimal advantage of Naive Bayes classifier.

Keywords—Sentiment Analysis, Movie Reviews, Naive Bayes classifier

1. Introduction

Opinion from others can be essential for better decision making on situations that involves valuable resources. With the advent of computer and communication technology and availability of blogs, forums, online review websites, and social networks enables to communicate and share ideas with everyone connected to the web [10]. Capturing such reviews on weblogs provides great deal of information that is valuable to companies and people for better decision making on the basis of the reviews and comments of their customers. An important factor of such review analysis is to characterize the opinion expressed in weblogs about specific brands and products [10].

Sentiment analysis refers to a group of tasks that use statistics and natural language processing to mine opinions to identify and extract subjective information from texts [1]. Sentiment analysis of movie reviews face the challenge of addressing the real facts which is generally mixed with actual review data [10]. People generally express their opinions in language that is often obscured by sarcasm, and discuss about the general traits of actors, plot of movie and relate the movie to their normal life in blogs that do not assign numerical ratings to movies. Because of this, analyzing the sentiment of movie reviews is difficult as the people discusses the artist of characteristics and in the end dislikes the movie. Thus, the movie review classification must be derived from the unstructured text of the review itself. Furthermore, another major challenges of movie review analysis is to analyze the negated opinion from the text of the review [10].

Based upon the above reasons, this project applies the Naive Bayes classifier machine learning techniques to predict polarity of movie reviews as positive or negative by understanding meaning

and relationship between the words. Mining the movie reviews and generating valuable meta-data provides an opportunity to understand the general sentiment around movies in an independent way.

The project is implemented using Python Programming Language and machine learning libraries of Python to predict sentiment of movie reviews as positive or negative using Naive Bayes classifier machine learning algorithm.

2. Technology Description

2.1 Movie Review

Movie review is the analysis and evaluation of movies and the film medium [8]. A number of websites allow web users to share movie reviews and scores in online that allow a broad consensus review of a movie. Websites, such as Rotten Tomatoes seek to improve the usefulness of film reviews by combining critic reviews and audience ratings and assigning a score to provide an overview of movie's quality. Blogs has also introduced opportunities for amateur film critics to provide their opinions [9]. Rotten Tomatoes website gathers together all of the reviews from top critics and averages out their scores. An entry on the website typically consists of a short quote, a link to the full review, and a 'fresh' rating of over 50% and a 'rotten' one of under 50% which summarizes whether the critic liked or disliked the movie [7].

2.2 Text Mining and Sentiment Analysis

Text mining and Sentiment analysis are commonly used interchangeably refers to tasks that uses data mining and natural language processing techniques to mine opinions to identify and extract subjective information from texts. Sentiment Analysis actually focuses on determining the polarity of a text, the polarity could be positive or negative and emotion recognition [1, 5].

Mining sentiments from natural language is challenging because opinions in natural language can be expressed in subtle and complex ways which requires better understanding of the syntactical and semantic language rules [1, 5].

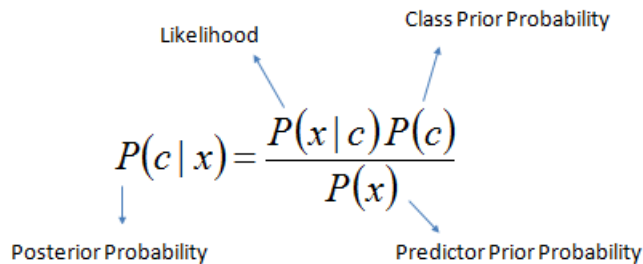
2.3 Machine Learning Algorithms

Machine learning methods are widely applied for sentiment classification problem. Mainly Support Vector Machine (SVM), Naive Bayes (NB), Maximum Entropy (MaxEnt) methods has been adopted by most of the researchers for sentiment classification. Pang et al. (2002) used different machine learning algorithms like NB, SVM, and MaxEnt for sentiment analysis of movie review dataset [1].

2.3.1 Naive Bayes Classifier

The Naive Bayes classifier is a standard probabilistic classifier based on Bayes theorem with strong independence assumptions between the events [11, 12].

Bayes theorem describes the relationship between probabilities of two events, based on conditions that might be related to the events. Bayes theorem is expressed mathematically as [12].

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$


Where,

- c is class for event occurring and x is the predictor event that has already occurred
- $P(c)$ is the prior probability of class and $P(x)$ is the prior probability of predictor.
- $P(c|x)$ is posterior probability, is the probability of observing class (target) given that predictor (attribute) is true.
- $P(x|c)$ is the likelihood which is the probability of observing predictor given that class is true.

Naive Bayes classifier assumes that the presence or absence of a particular event of a class is independent to the presence or absence of other events [11]. That is, on a given class (c), the effect a predictor (x) is conditionally independent of each other. This assumption is referred as class conditional independence [11]. Also, Naive Bayes classifier algorithm ignores the prior probability of predictor by assuming it has no impact on the relative probability [11].

Naive Bayes classifier is stated mathematically as the following equation based on the conditional independence assumption [11, 12].

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

2.3.2 Support Vector Machines

Support Vector Machines (SVM) is a statistical supervised learning algorithm used for text classification. SVM algorithm classify the data into two labels by learning a hyperplane from the training set that separates the data into two classes [25]. That is, on given labeled training data (supervised learning), the algorithm gives an optimum hyperplane which categorizes new data [25]. In this project, SVM classifier with a linear kernel of scikit-learn is used to predict the sentiment of movie reviews.

2.4 Python Programming Language

Python is an interpreted interactive object-oriented high-level programming language that is available on multiple hardware platforms and multiple software operating systems [4].

Enthought Canopy [20] is a comprehensive Python package for scientific and analysis environment that provides analytic Python computing distribution plus integrated tools for iterative data analysis, data visualization and application development. This project uses machine learning libraries of Enthought Canopy such as numpy[15] for arrays, scikit-learn[16] for machine learning as Naive Bayes Classifier, json[21] for parsing JSON data from the web, pandas[22] for data frames, matplotlib[23] for plotting, and requests[24] for downloading web content.

3. Project Solution

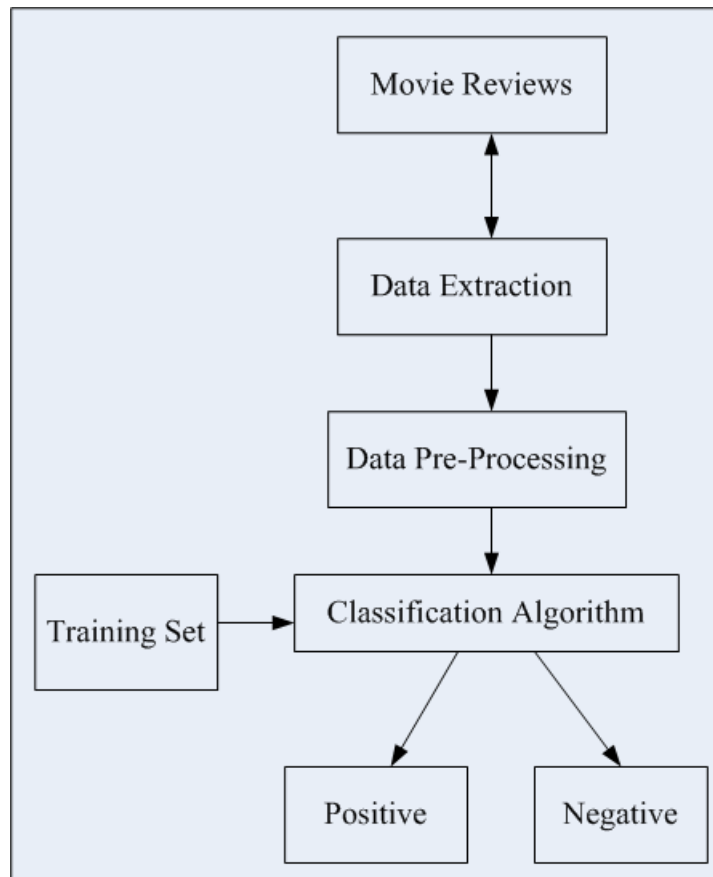


Figure 1. Solution flow

The overall solution of the project is illustrated in the **Figure 1** to classify the movie reviews in positive or negative sentiments based on the text of the review. The solution is implemented using the following machine learning phases [10].

1. Data Extraction
2. Data Pre-Processing
3. Classification Algorithm

In data collection phase, movie reviews are extracted from movie reviews source and parsed in to data frame. In pre-processing phase, movie reviews text is decomposed into numerical data and split the data randomly into training and testing sets. Finally, in the classification algorithm phase, which is implemented using a Naive Bayes classifier takes the training set as input during its initialization. For each review in the training set, the classifier learns how each feature impacts the outcome sentiment. Next, the classifier is given the testing set. For each review in the set, it predicts what the corresponding sentiment should be, given the features in the current review [26].

3.1 Data Extraction

MovieLens dataset [2] is used to collect the database of movies that includes information for about 10,000 movies, including the IMDB id for each movie. Rotten Tomatoes data API [3] (Application Programming Interface) is used to download first 20 top critic reviews from 3000 of these movies. Using JSON and Panda data Analysis tool, over 12,000 movie reviews are parsed from the Rotten Tomatoes web APIs into the data frame.

3.2 Data Pre-Processing

Once the movie review data is collected into the data frame, it is passed to pre-processing phase. Data pre-processing phase consists of breaking up the movie reviews text into vectors of features that serve as input for the classifier. Using a vectorizer object (CountVectorizer) of scikit-learn package translates the textual collection of reviews into a “bag of words” vector, that takes individual words in a sentence as features. In this approach, the movie reviews are represented by collection of words, disregarding grammar and word order where each word is conditionally independent from the other word.

In the next step of pre-processing, the data is randomly split into training and testing sets using “train_test_split” helper function of scikit-learn. With random split option, 3/4th of the data is taken as Training set to train the classifier and 1/4th of the data considered as testing set to validate the classifier.

3.3 Executing the Classification

Naive Bayes classifier “MultinomialNB” [13] function of scikit-learn is used for performing classification. The classifier is first trained over the training set to learn the characteristics or patterns residing in the data. After the training, the classifier is tested over the testing set to infer the sentiment of reviews. Finally, the result is compared against the original sentiment of Rotten Tomatoes movie reviews to evaluate the performance of the Naive Bayes classifier. Furthermore, 5-fold cross validation function of scikit-learn is performed to optimize the accuracy of the Naive Bayes classifier model. Finally, SVM classifier with a linear kernel of scikit-learn is used to predict the reviews and these results are compared with Naive Bayes classifier results.

4. Results

The project is implemented in Python Programming Language using scikit-learn library and demonstrated how to perform sentiment analysis of movie reviews using the Naive Bayes classifier. Python program output shows that the Naive Bayes classifier is applied over 12,699 reviews and achieved an accuracy of 93% on training set and an accuracy of 77% on test set. The gap between training and testing set is over 16% shows that classifier performance on training data is better than testing data and it did not extrapolate and perform well on test data. This phenomenon is referred as overfitting [17]. Using 5-fold cross-validation [14], the classifier achieved an ‘accuracy’ of 79% on training set and achieves 73% on test set, which is slightly less accurate compared to original Naive Bayes classifier results. However, it is less over-fit than before as the gap between train and test accuracy is narrowed down to 6% which gives accurate prediction probabilities.

Finally, Naive Bayes classifier results are compared with SVM classifier shows that both Naive Bayes and SVM prove to be good methods to learn sentiments from a review, but SVM is slight better than Naive Bayes in achieving considerable accuracy. Also, the results indicated that top features for positive reviews include words such as *powerful*, *rare*, and *touching* and top features for negative reviews include the words such as *unfunny*, *unfortunately*, and *dull*.

Table 1 shows the summary of overall performance of different classifiers explored in this project to predict the sentiment of reviews [26].

Model	Accuracy	Precision	Recall
Naive Bayes Algorithm	77%	78%	84%
Naive Bayes with 5-fold cross validation	73%	76%	80%
Support vector Machines (SVM)	78%	78%	85%

Table 1: Summary of Results

5. Future Improvements

Using Naive Bayes classifier machine learning technique on sentimental analysis of movie reviews is able to predict positive and negative movie reviews with 75% accuracy. Also, Naive Bayes learning classifier is faster than support vector machine (SVM) due to its simple classification algorithm. Improving Naive Bayes classifier has many advantages in terms of developing text classification models as it has been used in many machine-learning related classification projects [18]. Combining Naive Bayes classifier machine learning algorithm with other meta-learning approaches such as EM (Expectation Maximization) and Boosting could improve the classification performance [18].

Naive Bayes classifier assumes the existence of good quality data for training which is not practical all the time in real operational environments. Naive Bayes classifier in combination with EM algorithm improves overall accuracy of text classification by selecting optimal training data [18]. Boosting is a meta-learning technique for improving the accuracy of machine learning based classifiers, which combines a series of classifiers to produce a single powerful classifier [18, 19].

Furthermore, Naive Bayes classifier can be enhanced by choosing the right type of features and removing noise by appropriate feature selection such as Negation Handling, and n-grams [10].

6. Conclusions

Sentiment analysis is an evolving field with a variety of user applications. Although sentiment analysis tasks in natural language is challenging due to syntactical and semantic language rules, Naive Bayes Machine learning algorithm helped to better understanding of natural language opinions and reported high accuracy for predicting sentiment of movie reviews. Due to the conditional independence assumptions, Naive Bayes classifiers are extremely fast to train and can scale over large datasets. Both Naive Bayes and SVM classifiers are prove to be good methods to learn sentiments from movie reviews. Furthermore, Naive Bayes classifier can be enhanced by selecting the right type of features and combing with other meta-learning approaches.

7. References

1. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques.
2. https://github.com/cs109/cs109_data
3. <http://developer.rottentomatoes.com/>
4. Andy Bromberg(2013) Second Try. Sentiment Analysis in Python. Retrieved from <http://andybromberg.com/sentiment-analysis-python/>
5. https://en.wikipedia.org/wiki/Sentiment_analysis
6. http://nbviewer.ipython.org/github/cs109/content/blob/master/HW3_solutions.ipynb
7. <http://developer.rottentomatoes.com/f>
8. <https://en.wikipedia.org/wiki/Review>
9. https://en.wikipedia.org/wiki/Film_criticism
10. Data Mining and Analysis in the Engineering Field- Chapter 11: Machine Learning Approaches for Sentiment Analysis- Vishal Bhatnagar
11. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data-John Wiley & Sons- Chapter 7: Advanced Analytical Theory and Methods: Classification
12. http://www.saedsayad.com/naive_bayesian.htm
13. http://scikit-learn.org/stable/modules/naive_bayes.html
14. http://scikit-learn.org/stable/modules/cross_validation.html
15. <http://docs.scipy.org/doc/numpy-dev/user/index.html>

16. <http://scikit-learn.org/stable/>
17. <https://en.wikipedia.org/wiki/Overfitting>
18. Handbook of Research on Text and Web Mining Technologies by Min Song and Yi-fang Brook Wu (eds)- Chapter VII - Improving Techniques for Naïve Bayes Text Classifiers
19. Y. & Freund, R.E Schapire, (1996). Experiments with a new boosting algorithm, Proceedings from ICML '96: The 13th International Conference on Machine Learning, Bari, Italy: Morgan Kaufmann, 148–156.
20. <https://www.enthought.com/products/canopy/>
21. <https://docs.python.org/2/library/json.html>
22. <http://pandas.pydata.org/>
23. <http://matplotlib.org/>
24. <http://docs.python-requests.org/en/latest/>
25. https://en.wikipedia.org/wiki/Support_vector_machine
26. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data-John Wiley & Sons- Chapter 9- Advanced Analytical Theory and Methods: Text Analysis