PERFORMANCE EVALUATION AND IMPROVING PREDICTIONS OF CLASSIFIERS USING ENSEMBLE TECHNIQUES ON SENTIMENT LABELLED TEXT DATASET

ABSTRACT

Text mining is the discovery and extraction of interesting, non-trivial knowledge from free or unstructured text. This encompasses everything from information

retrieval (i.e., document or web site retrieval) to text classification

and clustering, to (somewhat more recently) entity, relation, and event extraction. [1]

INTRODUCTION

DATASETS

To conduct the research, three datasets have been obtained from UCI machine learning repository (<http://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences>) as text files.

The datasets contain sentences labelled with positive or negative sentiment, extracted from reviews of products, movies, and restaurants and are in the given format: Sentence /t score

The sentences were referred from the following websites:

* IMDB
* Amazon
* Yelp

Score is either 1 (for positive) or 0 (for negative) for each sentence.  
  
For each website, there exist 500 positive and 500 negative sentences. Those were selected randomly for larger datasets of reviews.   
The sentences have a clearly positive or negative connotation, there were no neutral sentences to be selected. [2]

METHODOLOGY

1. Collecting Dataset

The dataset used in the experiment are text files of specified with specific format of: Sentence \t Score.

Thus, the character ‘\t’ is used as the limiter for the dataset.

1. Pre-processing
   1. Cleaning the data – filtering all symbols but alphabets and white spaces

Using Regex, i.e. Regular Expression to filter everything but alphabets (for e.g. punctuations and various other symbols) and white spaces from the data.

* 1. Stopwords removal - The text is filtered using a stopword list. This filtering removes words that are very common in a language; for instance, in English the list includes all closed-class words such as “the,” “a,”, “in,” “he,” etc. [1]
  2. Stemming the words - A stem is a natural group of words with equal (or very similar) meaning. This method describes the base of particular word. [3] By stemming, we assume that the detailed morphological description of words is irrelevant for the purpose of classification. [1]
  3. Tokenizing - The process of splitting each document into the words that appear in it (called *tokens*), for example by splitting them on whitespace and punctuation is called tokenising. [4]

The Tokenizers used to obtain the results of this paper are:

* Count Vectorizer: Converts a collection of text documents into a sparse matrix of token counts. [5]
* TF-IDF (term frequency–inverse document frequency): The term frequency is typically defined as the number of times a given term t (i.e., word or token) appears in a document d (this approach is sometimes also called raw frequency). The TF-IDF can be understood as a weighted term frequency, approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents. [6]

1. Classification Algorithms
   1. SVM

Support Vector Machine algorithm classifies the data by finding a hyperplane, called the optimal hyperplane, between two classes. The SVM finds this hyperplane using support vectors and margins.

Although, the training time of SVM is slow, they are highly accurate, owing to their ability to model complex nonlinear decision boundaries. [7]

* 1. K – Nearest Neighbours classification

KNN classification model finds the distance of each point of testing data with every other point in the training data. Then it finds the nearest K members for that point.

The performance of a KNN classifier is primarily determined by the choice of K as well as the distance metric applied. [8] In this paper, we have used Euclidean distance among the data points to classify the data.

* 1. Logistic Regression
  2. Naïve Bayes

Naïve Bayes’ is a conditional probability model, based on the Bayes’ theorem considering Naïve (strong) independence assumption. [9] Naïve Bayes Classifier calculates the conditional probability of every feature then it selects the outcome with highest probability to predict the class of the feature.

In this paper we have explored the following Naïve Bayes Classifiers:

* MultinomialNB
* GaussianNB

MultinomialNB takes into account the average value of each feature for each class of outcomes, while GaussianNB stores the average value as well as the standard deviation of each feature for each class. [4]

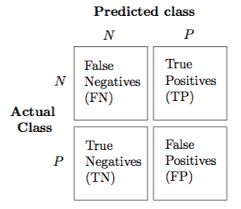
* 1. Decision Tree

Decision trees are widely used models for classification and regression tasks. Essentially, they learn a hierarchy of if/else questions, leading to a decision. [4] In other words, it repetitively divides the working area into the sub part of identifying lines, using the top down approach.

The degree of randomness of the decision tree used in this paper is entropy.

1. Evaluation of Performance

Confusion Matrix:



* 1. Accuracy

* 1. Precision

Precision measures the exactness of a classifier. [10] The value of precision is indirectly proportional to the number of false positives.

* 1. Recall

Recall measures the completeness or sensitivity of a classifier. [10] Unlike precision, recall is indirectly proportional to number of false positives.

* 1. F – measure

F – measure is defined as the weighted harmonic mean of precision and recall. [10]

* 1. Graphs

1. Ensemble learning techniques

Ensemble learning is when we take multiple machine learning algorithms and put them together to create one bigger algorithm. The algorithm created this way, thus leverages many other algorithms to predict the results.

* 1. Random Forest Classifier

Random Forest Classifier combines a lot of decision tree methods and creates a set of randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

* 1. Boosting

1. Evaluation of Performance ensembled algorithm vs Normal algorithm
2. Conclusion
3. References

# References

|  |  |
| --- | --- |
| [1] | A. Kao and S. R. Poteet, "Natural Language Processing and Text Mining," USA, Springer Science+Business Media, LLC, 2007, pp. 1 - 3. |
| [2] | P. Smyth, D. Kotzias, M. Denil and N. D. Freitaz, "From Group to Individual Labels using Deep Features," KDD, Sydney, NSW, Australia, 2015. |
| [3] | J. Sharma and P. M.Inzalkar, "A Survey on Text Mining- techniques and application," *International Journal of Research In Science & Engineering (IJRISE),* no. Techno-Xtreme 16, pp. 1-9, 2016. |
| [4] | S. Guido and A. C. Müller, in *Introduction to Machine Learning with Python*, United States of America, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2016, pp. 323-356. |
| [5] | "Scikit Learn," Scikit-learn Developers, 2017. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.CountVectorizer.html. [Accessed 2 August 2018]. |
| [6] | S. Raschka, "Naive Bayes and Text Classification," 2014. |
| [7] | J. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques," Waltham, USA, Elsevier Inc., 2012, pp. 408 - 410. |
| [8] | S. B. Imandoust and M. Bolandraftar, "Application of K-Nearest Neighbour (KNN) Approach for Predicting Economic Events: Theoretical Background," *Journal of Engineering Research and Applications,* vol. 3, no. 5, pp. 605-610, 2013. |
| [9] | L. Dey, S. Chakraborty, A. Biswas, B. Bose and S. Tiwari, "Sentiment Analysis of Review Datasets using Naïve Bayes’ and K-NN Classifier," Kolkata, India, 2016. |
| [10] | Jacaob, "Stream Hacker," 17 May 2010. [Online]. Available: streamhacker.com/2010/05/17/text-classification-sentiment-analysis-precision-recall/. [Accessed 30 July 2018]. |