SENTIMENT CLASSFICATION ON PRODUCT REVIEWS

Arun Kumar Pasupathi (29638917)
Ranganath Srinivasan Kalaimani (29360714)
Vishalakshi Baskar (29424437)

June 2, 2019

Abstract

Table of Contents

1 Introduction

This project is about developing automatic sentiment classification that relies on machine learning techniques to learn from a large data set of product reviews provided by yelp. There are 5 different levels of sentiment that classifies a review: Strongly Positive, Positive, Neutral, Negative, Strongly Negative. Each of the sentiment is considered as a numeric value and assigned to the reviews.

The training data, train_data.csv contains about 650,000 product reviews and the data train_label.csv contains its corresponding polarity. The test data, "test_data.csv" contains the reviews that needs to be classified. An efficient machine learning model is to be built to train the product reviews based on their sentiments and give prediction to the test data.

In our group, 3 features were developed from the given product review and each of us build 3 models on one of these 3 features and the model with highest accuracy is submitted for review.

2 Pre-processing and feature generation

2.1 Pre-processing

The various steps that were implemented on pre-processing the reviews are:

- All the reviews are converted into lower case
- The numbers in the reviews are removed
- The URLS doesn't contribute to the sentiments and hence removed
- Hashtags and contexts pointing to someone are removed
- Stop words hardly contribute to sentiments, hence removed
- Words are lemmatised and stemmed so that all the words are in their root form
- Punctuation in the reviews are removed
- Terms occurring more than 99% of the time are removed

2.2 Feature generation

There are 3 feature sets that are used in our group:

- Unigram
- Bigrams
- Sentiment Tokens

2.2a Sentiment Tokens:

Since this is a project about sentiment classification, we started by using sentiment tokens as the feature.The train data and the test data are merged together. There is a package in R (sentimentR), which extracts tokens that contribute to sentiment from the review (extract_sentiment_terms).

The above function classifies the terms as positive, negative and neutral. We take all these terms as a single feature. Therefore, these 3 columns are merged together to one column which is “sentitokens”. This column is again cleaned for punctuations.

Now the sentiment tokens is used to create a corpus. A Document term Matrix is created from the corpus which contains the documents as the rows and the terms as the features. Terms having document frequency more than 95% are removed and documents that do not contain the selected features are also removed automatically.

The term document matrix is then split into train data and the test data. The training data is used to train the classifier and the test data is used for prediction.

FEATURE STATISTICS:

DocumentTermMatrix (documents: 700000, terms: 199)>>
Non-/sparse entries: 4220627/135079373
Sparsity : 97%
Maximal term length: 23
Weighting : term frequency - inverse document frequency (normalized) (tf-idf)

2.2 b UNIGRAMS:

Unigrams are extracted from text after removing the stopwords, punctuations, numbers and the URLS from the Original product review. Also words with length less than 3 are removed from the list. A corpus is built with the remaining tokens and the corpusis split into train data and test data where the train data is used to fit the data to a model and the test data is used for prediction.

FEATURE STATISTICS:

DocumentTermMatrix (documents: 700000, terms:8451003)>>
Sparsity : 99%
Weighting : term frequency - inverse document frequency (normalized) (tf-idf)

2.2c BIGRAMS

Bigrams are extracted from the original text. The reviews are converted into lower case and with punctuations removed, bigrams are generated using unnest_tokens function with ngrams = 2. If both the words in the bigrams are stop words then they are removed. Eg words like “is_the” . A corpus is built on the remaining bigrams and the corpus is split into train and test data , where the train data is used to fit the data to a model and the test data is used for prediction.

FEATURE STATISTICS:

DocumentTermMatrix (documents:640063, terms:2406733)>>
Non-/sparse entries: 7104966/1540453639213
Sparsity : 100%
Weighting : term frequency - inverse document frequency (normalized) (tf-idf)

3. Models

There are 3 models used in this project.

- SVM Classifier
- Logistic Regression
- Random Forest

3.1 SUPPORT VECTOR MACHINE

SVM is a non-linear boundary classifier which is a combination of a support vector classifier and a non-linear kernel. We started with this classifer because it allows the user to enlarge the feature space and results in efficient computation.

ASSUMPTION: the classification problem to be a linear one i.e, the predictor variable and the response variable have linear relationship between them. Eg: A good review will have a positive sentiment and a bad review will have a negative sentiment.

SVM creates a hyperplane (boundary), where each observation can be classified based on which an observation is located with respect to the hyperplane i.e, observation with good reviews will lie on one side of the hyper plane and the bad reviews will lie on the other side of the hyper plane making the classification easy.

There are support vectors which lies on the margin of the hyperplane. The position of the hyperplane is directly dependent on the support vectors. **PROBLEM:** Sometimes, there are sarcastic reviews which are mis-classified even if we consider this to be slack variables there is a even bigger issue with SVM.

SVM is one of the more flexible model which may result in overfitting the data. Thereby leading to less training error high test error.

3.2 RANDOM FOREST

We considered Random Forest as our second model because it reduces the variance over a single tree. Also, Random Forest works by decorrelating the trees thereby making the resulting trees have less variance.

ADVANTAGES:

High Accuracy and are flexible.

DISADVANTAGES:

Prediction using Random forest is extremely time consuming and they are very complex, Also they require more computational resources.

Moreover the given data set is extremely large and this will result in extremely large number of trees and finding the relationship between the features will become highly problematic.

3.3 LOGISTIC REGRESSION

We use Multinomial Logistic regression to classify the response variable with more than 2 classes. **Assumption**: The chances of choosing one class over another class does not depend on irrelevant alternatives.

ADVANTAGES: Corelation between the features is not considered in Logistic regression. Also Logistic Regression easily handles huge volume of data i.e, as the training set increases,

we need not worry about the computational resources or the efficiency. Also Logistic Regression provides us a good probabilistic interpretation.
DISADVANTAGES: Handling sarcastic reviews becomes difficult i.e, logistic regression can handle only linear problems because its output is discrete. Non-Linear problems are not handled by logistic regression.

3.4 Discussion of model difference(s)

SVM and Random Forest uses more computational Resources for huge amount of data whereas Logistic Regression is computationally efficient. Random forest uses ENSEMBLE learning where multiple trees are trained to solve the same problem whereas SVM uses hyperplane between each classes and decides the class of the test set based on the distance between the test observation and each of the training observation.

SVM works by maximising the marginal difference between the observation and the hyperplane where as logistic regression works by maximising the posterior probability in the data. We can also say that, all of the discussed models works well when there is a linear relationship between the predictor variable and the response variable.

SVM and logistic Regression can handle some noise in the data where as Random Forest cant. Also SVM and Random forest may train the model with train data and hence have low training error but has high test error which is not preferable.

4 Experiment setups

This Experiment Setup result values is discussed only for the final model we have used.

TRAINING VALIDATION:
Training set validation approach is implemented for estimating the model accuracies with the given training data set. 80 percentage of data are treated as training data and the remaining are considered as test data. Model's accuracy is calculated by tabulating a confusion matrix of actual and predicted values and the accuracy of the training data with the **feature UNIGRAM AND BIGRAM on the model Multinomial logistic Regression is 62.007.**

5 Experimental results

Below are the accuracy of each feature with respect to its Model.

Feature Set	Model	Accuracy
Feature 1 - Unigrams	SVM Classifier	57.046
	Multinomial Logistic Regression	62.829
	Random Forest	53.728
Feature 2 - Bigrams	SVM Classifier	57.921
	Multinomial Logistic Regression	58.473
	Random Forest	50.015
Feature 3 – Sentiment Tokens	SVM Classifier	47.854
	Multinomial Logistic Regression	40.004
	Random Forest	38.221

It can be seen that Multinomial Logistic Regression perform better for the features Unigrams and Bigrams whereas in case of sentiment tokens, SVM classifier has performed well.
Thus for large feature set Multinomial Logistic Regression has performed well.

6. Conclusion

From this project it can be seen that Multinomial Logistic Regression perform better with large number of features. The computation efficiency of Logistic regression is high and it does not use much of computational resources. It is evident that text pre-processing plays a important role in selecting the features. For each document, the features are sent to the classifier where a sentiment score is calculated. Based on the sentiment score and the label of the train data the observations are placed in a p-dimensional space. When the new data or the test data comes, a sentiment score is calculated based on the features and is placed in a class to its train observation. Sarcastic Reviews will not be classified properly in the above models. Advanced Algorithms such as neural networks can be used to train such reviews.

References

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*(Jan), 993-1022.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionally. *Advances in Neural Information Processing Systems 26*, 3111-3119.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM, 38*(11), 39-41.

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. *the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532-1543.