Supervised and Unsupervised Algorithms for Text Analytics

Ganapathy Seshadri C Aiyer

Computer Science, NYU Tandon

New York, USA

Gsa277@nyu.edu

Jay Dharmendra Solanki

Computer Science, NYU Tandon

New York, USA

jds797@nyu.edu

***Abstract*-** Text Analytics is one of the major fields in Machine learning. With the increase in amount of data it is very important to group/classify data which has same features. This helps in information retrieval of relevant topics that a user wants. In this project we try to study different algorithms and find which algorithm is more efficient and accurate.

Keywords—machine learning

# Introduction

The aim of the project is to learn Supervised and Unsupervised learning algorithms on text classification using the 20 Newsgroups data. For unsupervised learning, the aim is to cluster the documents using unsupervised learning algorithms and then comparing it with labelled data.

For Supervised learning we have created a Web Interface in which a user can select various parameters for different classification algorithms and see its accuracy, error, ROC curve etc. The results are in a form of chart which makes it easier to understand. We have also compared the algorithms and based on their accuracy.

# Motivation

Text Analytics is one of the major fields in Machine learning. With the increase in amount of data it is very important to group/classify data which has same features. This helps in information retrieval of relevant topics that a user wants. In this

project we try to study different algorithms and find which algorithm is more efficient and accurate. Classification of news is considered as a relevant topic. Every user wants to view news only of some particular categories. These categories can be very specific from user to user. Therefore, our motivation was to find algorithm which classifies efficiently the news documents based on the 20 newsgroup data set. The supervised algorithms implemented here, will work for any kind of data set, if and only if we have a good set of labelled data to train the model.

# Related Work

[1] http://www.kamalnigam.com/papers/emcat- mlj99.pdf : Text Classification of labeled and unlabelled documents. We studied about Multinomial Naive Bayes algorithm and how it can be used effectively in our project.

[2] http://people.csail.mit.edu/jrennie/papers/ai memo2001.pdf : Improving Text Classification using Support Vector machine.

[3] Notes provided by Professor Raman Kannan on Text Analytics.

# data

20 Newsgroups dataset:

http://qwone.com/~jason/20Newsgroups/

This dataset consists of 18828 labelled news items. The news articles is divided into 20 groups and each news article consists of ‘from’, ‘subject’ and ‘news content’. All the articles belonging to a particular class are kept inside a folder, where the folder’s name is class name. Scikit provides us with functions that gets training and testing data for us. It parses the content of each news article and also provides us with the labels for each article.

# Algorithm(s) used

We have used the above UI to run algorithms by changing parameters and to analyze results.

Supervised Learning

1. Multinomial Naive Bayes Algorithm with Word Count and TF-IDF

In multinomial naive bayes we use a countvectorizer to find the count of each words occurring, except for stop words. We then either use this count vector to fit the multinomial Naive Bayes model or we can use this vector to find Tfidf score. We then pass this tfidf score to fit the model. In the model we pass the alpha which is the k-smoothing parameter. We then use the test set to find the performance measures. We have also used n-grams in countvectorizer, to see if there is a combination of words that is occurring more frequently. We have provided a user interface wherein the user can vary different parameters and check the performance measures, ROC curve and confusion matrix. We have used probabilities assigned to predicted classes as the confidence scores which we have used to obtain false positive rate and true positive rates and we have used fpr and tpr to finally obtain the ROC curve.

# Result

(In this section, can describe: Your experiment performance, accuracy/ROC/goodness etc.)

# Code

*(Include a link to your github repo with the code)*

# Video link

*(Include a link to your presentation video. You will upload this video to youtube and paste the link here.)*

# Evaluation

(Please comment on what went well and what didn’t go well in your experiment. If you had more time what could have been improved?)

# Conclusion