MOVIE REVIEW SENTIMENT ANALYSIS USING SUPPORT VECTOR MACHINES

BY

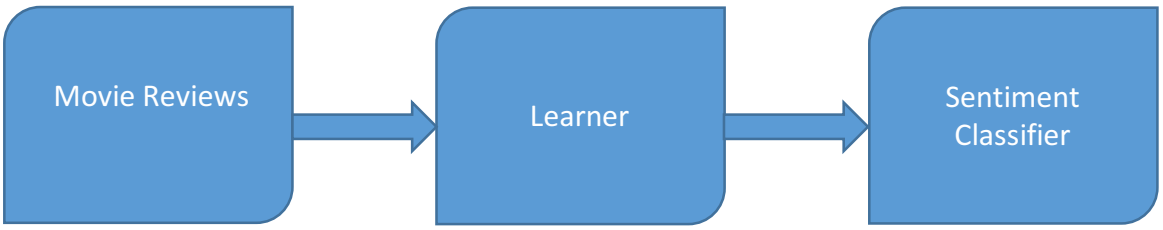
SAMEERA MOGULLA, AARTHY MOHAN, PARIDHI SAXENA

GOAL

The goal of this project is to classify a movie review, written by the viewer, as positive review or negative review.

INTRODUCTION

Sentiment Analysis has become a popular field with growing popularity of social media like facebook, tweeter; online shopping sites like amazon, ebay for reviews for product. People rely on reviews given online before purchasing anything or even before going to watch a movie. Hence sentiment analysis is also known as “opinion mining”. It contributes a lot towards the decision of the customer. So it has become an important issue for companies as well to see if their products are getting positive reviews. This whole cycle leads to automated categorization of sentiments. It has many applications. Firstly, it would help customers to classify if the product is good to buy or not, or services are good or not. Secondly, this would also proof beneficial for business man as it will provide a feedback. Say, automated classifier can tell the percentage of happy customers without having actually to read any review. It is also applied to emails also to classify is the mail is spam or not. Or google calendar pick up the meeting or appointment date. It’s all possible because of the automated classifier. In this report, we would focus on movie review classification. We categorize the movie review as either positive or negative. Positive is denoted as 1 and negative as 0.



Firstly, we trained classifier using examples. Movie Review is given as input and producing the output as positive or negative. Once we have trained classifier, then we use to test the classifier with the new reviews (test data). So we calculate the performance of the classifier by its percentage of accuracy during test data.

BACKGROUND

Automated classification history includes a document level investigation by Turney[1] and Pang[2]. They discussed different methods for detecting the polarity of product reviews and movie reviews. Pang and Lee [3] extended the classification of movie review to predict the star scale on the basis of positive or negative review. Prediction of star based on the reviews is a way of classification on a multi way scale. Initially classification includes only 2 categories – positive and negative. Later on, researcher believe that third category need to be included as well i.e. neutral. Third category is aimed to improve the over all accuracy of the classification especially in the algorithm like SVM or Max Entropy. Another method is when we associate weight with the commonly used classifiers words, which is called scaling system. Classification of review is then based on over-all weight – if total weight is positive then review is considered positive or if total weight is negative then review is overall negative. Total weight is sum of the signed weights of all words in that review and it defines the polarity of the review.
There are many machine learning algorithms available to build classifiers. We will Support vector machine in detail. Generative Classifiers like naïve bayes build a model of each class. Given an input, this classifier returns the class most likely to have generated that input. Discriminative classifier like Logistic regression instead learn what features are mostly used to differentiate between possible classes. Discriminative type of classification are often more accurate and hence more commonly used. Other classifiers include Support Vector Machines (SVMs), ransom forests, perceptron, and neural networks.

APPROACH

As discussed above, there are many algorithms available but we chose Support Vector Machine because it gives the highest accuracy. SVM uses a function called a kernel to map a space of data points in which the data is not linearly separable onto a new space in which it is, with allowances for erroneous classification.
SVM is not probabilistic but a large margin classifier. The name “support vector” is because the point on the margin between hyperplane and the nearest data points are called support vectors. SVM finds a parameter vector alpha that maximize the distance between the hyperp lane and every training input. One thing to pay attention is that the distance of the points from hyper plane margin will be no less that $1/ ||\alpha||$. The solution of the problem is $w = X\ j\ \alpha j\ cjdj$, $\alpha j \geq 0$. Once the classifier is built, classification of new new movie reviews simply involves determining which side of the hyper plane that they fall in. In our case, there are only two classes (positive or negative). So need to consider non-linear classification. However, the SVM is flexible in non-linear situations using the kernel, which transforms the input to a higher dimensional space. For every classifier, we performed a 4-fold cross validation and

then found the average. N-fold cross validation requires splitting the training data into N sets. One of those sets is considered as test set to measure accuracy, and all other sets are used to train the classifier. This is repeated N total times, one for each separate data partition. Afterwards, the accuracies were averaged and reported.
Also, we measured precision and recall values for each of the labels, "positive" and "negative". Precision here is measured as the number of true correct results over the all the positive results,

Precision = tp/(tp + fp)

where, tp = true positive (correct result)
fp = false positive (undesired result)

Recall is the true positive rate of the classifier, and is measured as:

Recall = tp /(tp + fn)

where, fn = false negative (missing result)
tp = true positive (correct result)
fp = false positive (undesired result)

This algorithm also measures the complexity of the hypothesis function based on the margin which separates planes.

DATASET

We were given 25000 reviews as training set which include 12500 positive reviews and 12500 negative reviews. We were given a test set which contains 11000 reviews.

EVALUATION

I. Text Pre-processing
This is the process of preparing and cleaning the data from the dataset. We followed the following steps to have the data properly per-processed. It is important to clean the data because it will help to speed up the classification, thus aiding in real time sentiment problems.

a) Tokenization:
It is the process of chopping the input line into words called tokens and simultaneously we are removing all the punctuation marks (spaces, tabs, commas, full stop). A token is the instance of sequence of characters that are grouped together as a useful semantic unit for processing.

b) Stop Words:
Stop Words are those words which are the most common in that language. It may differ in different languages. Every Language is provided with the list of stop words, so we downloaded it form the internet. Some of the more frequently used stop words for English include "a", "of", "the", "I", "it", "you", and "and" these are generally regarded as 'functional words' which do not carry meaning. So we ignore these words as they appear often and provides no information for classification.

c) Stemming:
It refers to the process where words are reduced to their root. It is a very common algorithm called Stemming Algorithm. A simple stemmer looks up the inflected form in a lookup table, this kind of approach is simple and fast. The disadvantage is that all inflected forms must be explicitly listed in

table.eg. “developed”, “development”, ”developing” are reduced to the stem “develop”.

II. Transformation:

Next task to assign the weight to each word. The weight is being calculated with the help of the function TF-IDF. It helps to determine which words might be more good to use in further classification. TF-IDF function calculates weight for each word defined as

wd = fw,d* log(|D|fw,D)

where, D is collection of documents,

w represents word,

d is individual document belongs to D,

|D| is size of corpus,

fw = f*w that is the frequency of that word

III. Feature Selection:

This feature makes classifier more efficient by reducing the amount of data to be analyzed as well as identifying relevant features to be considered in classification process. Feature Selection process will refine the features, which will be the inputs to the classifier during training(learning) stage.

IV. Classification:

At this stage, we classify the data to either positive or negative classes. These are both pre-defined classes. Since SVM is a supervised learning problem, we transform the text into csv file, which makes it suitable to be fed into the classifier. This leads to attributed value representation of text. Each word corresponds to feature with, number of times word occurs in document, as its value. Words are considered as features only if they are not stop words (like “and”, “or”, etc). Scaling the dimension of feature with IDF improves the performance.

EXPERIMENT SETUP

For cross-validation we divided the training set into 60% training data and 40% test data and got an accuracy of ~89%

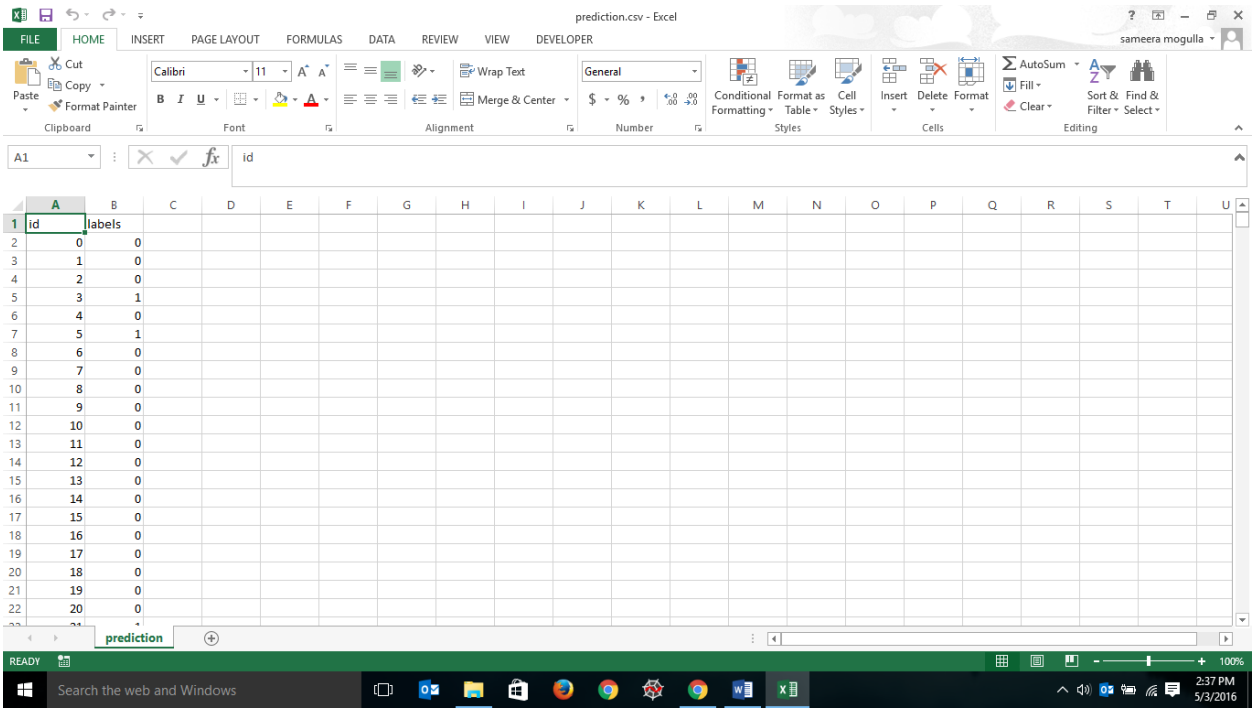
RESULTS

For cross-validation

```
In [6]: correct = 0.0

In [7]: for i,j in zip(test_labels,prediction_linear):
   ...:     if(int(i)==int(j)):
   ...:         correct+=1
   ...:

In [8]: correct
Out[8]: 8875.0

In [9]: accuracy = correct/len(test_labels)

In [10]: accuracy
Out[10]: 0.8875887588758876
```

	precision	recall	f1-score	support
0	0.90	0.88	0.89	5007
1	0.88	0.90	0.89	4992
avg / total	0.89	0.89	0.89	9999

OUTPUT TEST FILE



TEAM ROLES

SAMEERA MOGULLA: Writing Code and Presenation
AARTHY MOHAN: Code and Presentation
PARIDHI SAXENA: Code and Report

REFERNCES

1. Turney, Peter (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". Proceedings of the Association for Computational Linguistics.
2. Pang, Bo; Lee, Lillian (2005). *"Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales"*. Proceedings of the Association for Computational Linguistics (ACL).
3. Snyder, Benjamin; Barzilay, Regina (2007). "Multiple Aspect Ranking using the

Good Grief Algorithm". Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL). pp. 300–307.
4. Thelwall, Mike; Buckley, Kevan; Paltoglou, Georgios; Cai, Di; Kappas, Arvid (2010). *"Sentiment strength detection in short informal text"*. Journal of the American Society for Information Science and Technology.
5. Thorsten Joachims: Text categorization with support vector machines: learningwith many relevant features, Proc. of ECML-98, 10th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE,pp. 137-142,1998.
6. FabrizioSebastiani: Machine learning in automated text categorization, ACM Computing Surveys (CSUR), Vol. 34 Issue 1, ACM Press, New York, NY, USA, pp. 1-47,2002.