# ECE 681 Course Project 25% Status Report

**SHENGXIN QIAN**

## 1. Problem Description

### 1.1. Project name

Classification on sentiment of movie reviews

### 1.2. Project details

I plan to use sentiment classification to analyze movie reviews. The result of sentiment analysis could represent sentiment behind reviews. I plan to use the SVM and Naïve Bayes classifiers to determine the sentiment of reviews on IMDB website. It is a binary classification which would label reviews with “positive” and “negative”.

### 1.3. The reason why I choose this topic

In the proposal before, I planned to analyze the sentiment of product reviews on Amazon. However, there is no standard database of Amazon reviews and linguistic features of product reviews are varied corresponding to the type of products. So, I chose to do classification on standard movie reviews database and the linguistic features of one type of product are easier to choose.

### 1.4. The importance of sentiment classification

As I said in the proposal, sentiment classification would label the reviews which do not contain star rating indicators and help normalize the different rating schemes of different reviewers use. Sentiment classification could also help companies to do preanalysis on a free-form survey in natural language format.

### 1.5. Previous work on this problem

Sentiment classification was used in many business intelligence applications such as MindfulEye’s Lexant system, message filtering system (Spertus, 1997)[2]. Sentiment analysis is a very popular topic in machine learning area. In 1994, Daniel Jurafsky finished “The berkeley restaurant project” [1] which is a research on the sentiment of restaurant reviews and built the corresponding database. In 2005, Pang and Lee [2] collected many linguistic structures from movie reviews. Many PPML algorithms were used including SVM, NB and many other advanced ML algorithms [3]. The accuracy of sentiment classification has been more than 80% [4].

## 2. Data Description

### 2.1. The source of dataset

I used standard movie reviews database[1] as my dataset for training and testing. The reviews were gathered from IMDB website and preliminary steps were taken to remove rating information in HTML files. There are 1400 processed down-cased txt files in standard reviews database and half of them were labeled positive in “pos” folder and the rest were labeled negative in “neg” folder. Each txt file only contains the content of reviews and each line in each text file corresponds to a single sentence, as determined by Adwait Ratnaparkhi's sentence boundary detector MXTERMINATOR.

### 2.2. Training data classification decision

The original HTML files do not have consistent formats -- a review may not have the author's rating with it, and when it does, the rating can appear at different places in the file in different forms. The providers only recognized some of the most explicit ratings, which are extracted via a set of ad-hoc rules. In essence, a file's classification is determined based on the first rating providers were able to identify. With a five-star system (or compatible number systems), three-and-a-half stars and up are considered positive, two stars and below are considered negative.

## 3. Approach

### 3.1. Pre-processing

Intuitively, we might suspect that there are certain words people tend to use to express strong sentiment. So, it might be helpful if we simply select a group of sentiment words by introspection and rely on the selected words to classify. This is the logic behind my first stage approach.

I use only 7 positive words and 7 negative words to represent positive and negative sentiment at this stage (bigrams). The selection of those words was based on Pang and Lee ‘s research [2]. I use the following bag-of-features framework with those selected words. Let $\{f_1,\ldots, f_{14}\}$ be the set of features and $f_i$ represent one pre-defined sentiment word. Let $n_i(d)$ be the number of times feature $f_i$ occurs in review d. Then, the program would produce a vector $(n_1(d), \ldots, n_{14}(d))$ for each review in the database as decision statistics. So, with 1400 reviews, those samples could be represented as a 1400 x 14 matrix.

### 3.2. Classifier selection

I plan to choose SVM and Naïve Bayes classifier at the first stage [3]. According to Pang and Lee ‘s research [2], SVM classifier could achieve the best accuracy with bag-

[1] This data was first used in Bo Pang and Lillian Lee,``A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization. Based on Minimum Cuts'', Proceedings of the ACL, 2004.

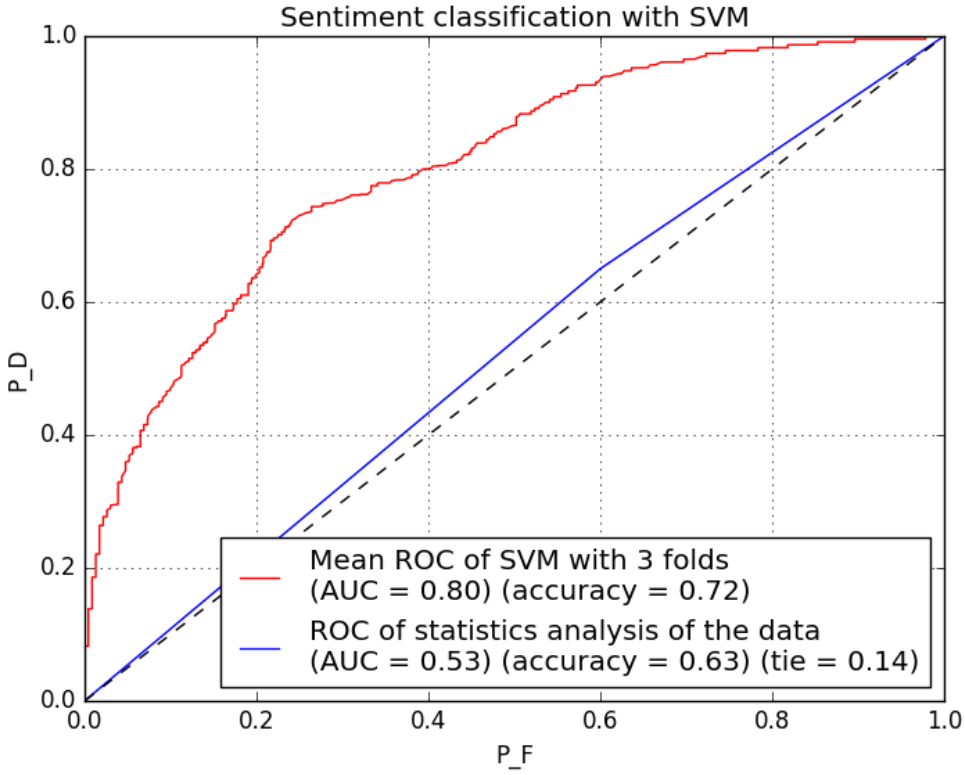
of-words feature framework compared with Naïve Bayes and Maximum Entropy classifier. I only focus on the accuracy of the classifier. In the first step, I only choose SVM classifier with linear kernel.

### 3.3. Results

At the first stage, I use my program to produce the result for baseline result for SVM classification. As we know, for normal binary classification problem, the minimum accuracy is 50%. However, without machine learning methods, I could make accuracy better simply with the statistics of feature words and that is the new baseline.

The program would calculate the number of positive words occur and negative words occur. If the times of positive words occur is higher, the classification decision is "positive". If the times of negative words occur is higher, the classification decision is "negative" and the rest is "tie". After using this algorithm, the program could achieve 63% accuracy with 14-word feature framework. At the same time the percentage of "tie" is 14% which limits the accuracy of this simple approach. It means the accuracy should be higher than 63% with machine learning method and 14-word feature framework.

With SVM classifier and three-fold cross-validation, the average accuracy of



classification is 72% which is 9% more than our baseline. The ROC curve was shown below:

### 3.4. Challenges and Next Steps

As we can see, the accuracy of random choice is 50%. With just statistic data of 14-word feature framework, the accuracy becomes 63% which proves this feature framework really work. With this feature framework and SVM, the accuracy becomes 72% which proves my approach is on the right track. However, the “tie” rate is very high in simple statistic approach, which means many words in feature set did not occur and the decision statistics matrix is sparse. The sparse of that decision statistics matrix would also affect the accuracy of classification with SVM.

Actually, the 14-word feature framework is too simple and it is just a prototype feature set. In the next step, I plan the choose more sophisticated feature frameworks and use PCA to extract the proper sentiment words. The option could be:

- More sentiment words in a feature set with PCA extraction
- Use unigrams or n-grams but not bigram in feature set

## 4. Reference

[1] Jurafsky, Daniel, Chuck Wooters, Gary Tajchman, Jonathan Segal, Andreas Stolcke, Eric Fosler, and Nelson Morgan. "The berkeley restaurant project." In *ICSLP*, vol. 94, pp. 2139-2142. 1994.

[2] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79-86. Association for Computational Linguistics, 2002.

[3] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer, Berlin: Springer series in statistics, 2001.

[4] Socher, Richard, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. "Recursive deep models for semantic compositionality over a sentiment treebank." In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631, p. 1642. 2013.