# CIS 520, Machine Learning , Fall 2011: Final Project

Team: **Angry Classifier**

Members: Tao Feng, Yayang Tian, Wenbin Zhao

## Part One: Introduction

In this project, we tried a lot of machine learning approaches to address a sentiment analysis problems on Amazon. We first eliminated words that are non-related to ratings, most of which are objective words.We used methods including frequency-based word-probability，TF-IDF weighting, stemming, PCA and IG (F). Then we added many useful additional features, like data, title and bigram features. Furthermore, we tried and compared standard machine learning approaches. During the competition, we constantly utilized cross validation to evaluate our performance. After trying svm, adaboost, logistic regression, knn and our own kernel, we found that a combination of several features after performing PCA using liblinear could give the us best performance: RMSE of 0.8529.

## Part Two: Feature Selection

Features are a critical aspect in sentiment analysis. As features are important parts of data-driven approaches to text processing, well-formed features increase the correctness of machine learning methods. In order to achieve excellent features to enhance the performance, we should diminish redundant features as well as adding useful attributes.

### 1. ***Dimension Reduction***

(1) Option Mining

a) Part-of-speech (POS)

POS information is commonly exploited in opinion mining which is considered to be a crude form of word sense disambiguation [1]. Because not all words contain sentiment components, some high-frequency words may have no contribution to the sentiment of the review. As there is a high correlation between the presence of adjectives and sentence subjectivity [2], adjectives are good indicators of sentiment.

In our model, we refer to the lexicon to pick up the adjectives and adverbs and then identify them as positive, negative or neutral. We discard all the non-emotional words to increase the accuracy of our model. Because words expressing the feelings of the reviewer are selected by POS, the performance of our method is increased.

B) Stemming and stop-word-list

Stemming improved the overall performance. The principle behind this idea is that since there are many forms of a single words, it is reasonable to treat them as one word; Yet using a stop-word-list eliminating some high-frequency objective words  don't render good effect on RMSE.

## (2)Standard Unsupervised Learning

### a) PCA

PCA is a method for transforming a data set into a lower dimensional space while minimizing the loss of variance. It is widely used in the field of dimension reduction and it has also achieved good results in data processing. As the dataset is very huge which is impossible for us to do kernel things directly, PCA seems a suitable trial.

In our model, in order to deal with such a huge dataset in MATLAB we should preprocess both train and test data by POS and other tricks to reduce the scale to a reasonable size. We manage to make the number of feature down to 10000 by preprocessing and again make it as few as 500 by PCA. However, the performance of our model decreases apparently. The reason may be the initial preprocessing due to the restriction of memory. It may discard many meaningful pieces that PCA cannot select the truly 'representative' tokens in the features.

### B) Information Gain

We compute the information gain for all the features in the training data and then sort them from highest to lowest. In order to get the most useful features in the training data, we use cross validation to compute the mean RMSE and find out that the first 8000 words in the unigram and the first 80000 bigrams are most useful to improve the RMSE. This method works well in the training data and cross validation for both NB and SVM.  It improve the average RMSE in the cross validation from 1.18 to 1.05. However, in the test set, the RMSE is worse than using the whole features. We think this is because that there are a lot of words in the test set  that never appear in the train set and compute the IG in the train set and select the features will leave out many words that may be much more useful in the test set.

## 2. ***Adding Meta-data Information***

1. Text Length

Text length is the total number of tokens in a syntactic analysis of the review. Here we take the number of words in each review as our feature. It shows that RMSE of our model decreases apparently. This is because people tend to give quite long comments when they are under some extreme mood, which means the rating is either 1 or 5. The review length can successfully detect such distribution.

2. Category

An alternative way of summarizing reviews is to extract information on what category the produce belongs to. As different categories have different related tokens and different meta-data attributes which reviews in the same category shares some common traits, grouping reviews in different categories helps increase the performance of the model.

In our model, we add the feature of category to the original dataset and a better result has been observed. This improvement is due to the correlation of words and other features in the same category which increase the polarity of outcomes.

3. Positive & Negative Word Frequency

Another method we adopt to improve out model is the introduction of positive & negative word frequency. Regardless of negation, it is easy to reveal review's feeling simply by selecting both positive and negative words in the review and checking which one is dominative. Comments which have more negative words tend to receive a lower rating while more positive words always means a higher rating.

This method is also implemented in our model but achieve quite a little progress. The main reason is the information of positive & negative is contained in POS that this processing seems a little redundant. Although that, it also works.

4. Frequence vs Presence

We compare the presence of words as features with the count of words as features. And we find out that count the words as features are much better than the presence fo word as features. This is because certain words tends to appear more at a kind of reviews with certain stars, and take their numbers as features are more meaningful.

| | unigram | bigramt |
|---|---|---|
| Frequence | 1.10 | 1.03 |
| Presence | 1.22 | 1.18 |

5. Helpful , Date and Title

We try to add the feature of helpful as an additional feature. However, the performance decrease after adding the helpful. The mean RMSE for the cross validation increase from 1.18 to 1.21 when using SVM. As with the helpful, we add the month, year and day as features to the dataset, the performance significantly decrease and the RMSE increase from 1.13 to 2.2. Yet when we use traditional method like Naive Bayes, helpful, month and title information help RMSE from baseline to 1.4596 to 1.3486.

6. Negation (Bigram )

Handling negation can be an important concern in opinion- and sentiment-related analysis. While the bag of-words representations of "I like this book" and "I don't like this book" are considered to be very similar by most commonly-used similarity measures, the only differing token, the negation term, forces the two sentences into opposite classes.[3] Thus our aim is to find the negation in the reviews and process related data.

We use bigram tokens to detect whether there are some negation words or not. If some bigram begins with a negation word and then a positive word, this word should be negative, and vice versa. By processing words according to this rule, the performance of our model increases

apparently. However, as we only detect bigrams and find the negation relationship between two adjacent words, sentences such as “not quite well” cannot be detected.

## Part Three: Machine Learning Methods

We implement altogether 6 : unsupervised method PCA and Information gain, bigram features, a discriminative method, own kernel, and one instance-based method-KNN.

### 1. SVM

A support vector machine (SVM) is a concept in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis [4]. It is the most widely used machine learning algorithm to measure the effectiveness of machine learning approaches. Support Vector Machines are motivated by the notion of the maximum margin separating hyperplane which will minimize the hinge loss.

We implement the SVM method to help train our model. Based on liblinear, we implemented different kinds of SVM with parameters from ‘-s 1’ to ‘-s 5’. Most models have RMSE around 0.9 which are kinds of great results. In order to decrease RMSE as low as possible, we assume that the ratings come from a discretization of a continuous function method instead of classification. Although accuracy is decreased at the same time, this regression method outperforms classification apparently.

SVM has the best performance among all methods we have tired. SVMs are large-margin, rather than probabilistic, classifiers, in contrast to Naive Bayes and others. SVMs are not constrained by conditional independence of the features given the class as Naive Bayes, it is a more general method.

### 2. Discriminative Method: Logistic Regression

The performance of logistic regression is not bad. We think since its work mostly based on the MLE, in general cases that it will find out the vector of w to maximize the MLE. Since negative reviews are not always contain too many negative words, it works bad on these examples.Performance of Logistic Regression(RMSE)

### 2. Boosting

We implement the Adaboost.MH. It works as follows:

1. Expand the original N observations into $N \times J$ pairs,
$((x_i, 1), y_{i1}), ((x_2, 2), y_{i2}) \ldots$ , $i = 1, \ldots N$ response for class j and observation i.

2. Apply Real AdaBoost to the augmented datset, producing a function
$\mathcal{X} \times (1, \ldots., J) -> R; F(x, j) = \sum_m f_m(x, j)$

3. Output the classifier $argmax_j F(x, j)$

This method works well in general. However, since we do not have proper weak classifier and there are too many features, we think this method is not suitable for this task.

Performance of Adaboost.

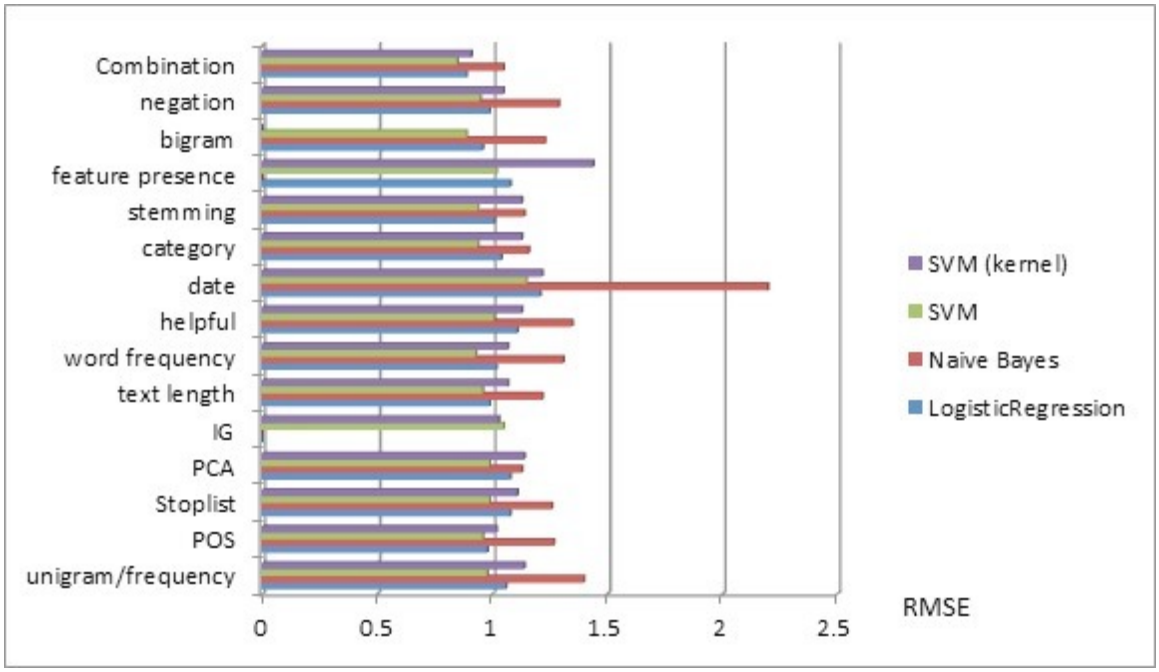
### 4. An instance based method : KNN and Own Kernel Method

These two methods share the same characteristic : not very suitable for such large dataset. We implement KNN and intersection kernel , both of which are extremely time consuming. For KNN, in the training phase, the system restore the feature vectors and class labels; in the testing phase, a test point, say, a customer's review, is classified in terms of metric calculated among the K most similar examples. In order to know which K examples are the most similar ones, we have to compare among 60000 examples. This would cost a lot of memory that is impossible to run on an average computer. So, as what has been discussed previously, we split the examples into random pieces and use dimension reduction approach PCA to select the most informative ones. Results show that KNN and kernel method don't render apparent better performance.

## Part Four: Conclusion

After comparing all these feature selection methods and machine learning methods, in our model we use POS, stemming, negation detection with unigram and add mega-data features such as text length, word frequency and category for feature selection. Based on SVM which is the best method we have ever tried, we find out that the combination of several methods can make the RMSE of our model be as low as 0.85. It shows the use of multiple classifiers in a hybrid manner with feature selection can result in a better effectiveness of RMSE than any other single methods.

| | LogisticRegression | Naive Bayes | SVM | SVM (kernel) |
|---|---|---|---|---|
| unigram/frequency | 1.06 | 1.4 | 0.98 | 1.14 |
| POS | 0.98 | 1.27 | 0.96 | 1.02 |
| Stoplist | 1.08 | 1.26 | 0.99 | 1.11 |
| PCA | 1.08 | 1.13 | 0.99 | 1.14 |
| IG | - | - | 1.05 | 1.03 |
| text length | 0.99 | 1.22 | 0.96 | 1.07 |
| word frequency | 1.02 | 1.31 | 0.93 | 1.07 |
| helpful | 1.11 | 1.35 | 1.01 | 1.13 |
| date | 1.21 | 2.2 | 1.15 | 1.22 |
| category | 1.04 | 1.16 | 0.94 | 1.13 |
| stemming | 1.01 | 1.14 | 0.94 | 1.13 |
| feature presence | 1.08 | - | 1.02 | 1.44 |
| bigram | 0.96 | 1.23 | 0.89 | - |
| negation | 0.99 | 1.29 | 0.95 | 1.05 |
| Combination | 0.89 | 1.05 | 0.85 | 0.91 |




In order to improve our work, we should consider more details in implement these methods to lower RMSE. For example, we could use Lexical Chain and Latent Semantic Analysis tools SentiWordnet to cluster positive and negative sentiment words into group; or we can use n-gram as well as dependency relation combined with PA, Winnow Classifier to further improve our performance.

## Part 5: Reference

[1] Yorick Wilks and Mark Stevenson. The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. Journal of Natural Language Engineering, 4(2):135–144, 1998.

[2] Vasileios Hatzivassiloglou and Janyce Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In Proceedings of the International Conference on Computational Linguistics (COLING), 2000.

[3] Bo Pang and Lillian Lee. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, Vol. 2, Nos. 1–2 (2008) 1–135

[4] Support vector machine. Wikipedia. http://en.wikipedia.org/wiki/Support_vector_machine.