Charles University in Prague

Sentiment Analysis: Polarity Dataset

Duc Tam Hoang

March 4, 2014

1 Introduction

Sentiment analysis (opinion mining) has been a classic topic of natural language processing. It is the field of study that analyses people's opinions, evaluation, attitudes towards an object. The object could be a product, a service, a movie, a particular individual and their attributes.

Sentiment analysis is obvious instead of vague in the era of information. People nowadays are surrounded by a huge load of sentimental expressions from social media, news and from other people. For example: Mike bought an Iphone yesterday. Here is what he said on Facebook "Oh! that was so cool, I have the best smart phone ever". Then the question is: Does he like the mobile phone?. This is a question that can be answer by sentiment analysis research.

Throughout history, the term *sentiment analysis* first appeared in 2003. However, the research on sentiments may began earlier in 2001. Although linguistics and natural language processing (NLP) have a long history, research about people's opinions was considerable young.

Since 2000, this field has become an attractive branch of NLP. There are several reasons for this. First, it has a wide range of applications. Especially, the commercial applications have flourished recent years. For instance, a company sends a new product (TV, cellphone, shampoo, milk, etc) to the market. They want to know how customers think about their product. Does users like it? Which part of the product is criticized? All those things are closely related to opinion mining. Second, it offers a challenging research problem, which has never been studied before. This task is totally different from searching/ranking, question/answering, text categorizing. So far it results in novel approaches and a lot of enthusiasm from researchers. Third, there is huge volume of opinionated data in the social media. Data is the key of research. Without an appropriate set of data, a lot of research would not have been possible, especially for statistical approach. Hence, research in sentiment analysis not only has an important impact on computation linguistics, but may also have great effect on other aspects.

1.1 Task Description

The project **Sentiment Analysis** focuses on the task of binary classification. Given two predefined classes, the binary classification task is to predict an instance belong one class. The performance of a classifier is normally measured by the accuracy. The reason is that two classes are equally important.

The dataset used in the project is *Polarity dataset.* It is a dataset in the domain of movie reviews, provided by various critics. The classes are *positive reviews*, which give a compliment about a specific movie, and *negative*, which criticizes the movie. The corpus is a collection of reviews classified by two labels *positive* and *negative*, associated with two classes. An instance is the raw text of review. The movie title is anonymous. The reviewer is also unknown. Details of the dataset are provided in section 2.

The main objective of the task is to experiment the sentiment analysis task with a number of method in feature extraction/selection and learning methods. For the task of sentiment analysis, the method of features selection and learning algorithm may behave differently. Out of all possible technique, and additional goal is to select a good method for the sentiment analysis. So far the scope of project is limited in term of domain by the *dataset.* However, there is no limitation in term of Machine Learning methods.

In the project, many experiments have been conducted. There are two main phases: *feature analysis* and *learning algorithms.* The first phase is responsible for selecting best feature sets. Many initial experiments (without parameters tuning) are conducted in this phase to compare the performance of two feature sets. Finally, 8 sets of features are drawn. They are the result of a combination of feature selection methods (document frequency, information gain, Fast Correlation-Based Filter, and so on). The above feature sets are input of the learning algorithm phase. For each algorithm, the set of parameters are tuned. Some embedded algorithms are a combination of other algorithms.

The report is organized as follows: Chapter 2 provides a description of the data used in my project. Chapter 3 presents an analysis of feature extraction and feature selection. In chapter 4, there is a description and comparison of experiments and results of different methods. The summary is drawn in chapter 5. In chapter 6, I provide a picture of related works. Finally, a brief conclusion is delivered in chapter 7.

2 Polarity Dataset

The *sentiment analysis* project was implemented with data of movie reviews, archive of the `rec.arts.movies.reviews` newsgroup. The dataset is **polarity dataset**, published by Bo Pang and Lillian Lee, available online at `http://www.cs.cornell.edu/people/pabo/-movie-review-data/`. According to authors, they selected solely reliable reviews (reviews with stars or some numerical value) from the Internet Movie Database (IMDb). They converted and extracted ratings into three categories: positive, negative and neutral. For this dataset, they concentrated on discriminating positive and negative statements. The newest version of *polarity dataset* is 2.0 (PLII). The version consists of 1000 positive reviews and 1000 negative reviews.

Throughout history of *sentiment analysis*, polarity is one of the most well-known dataset. It is a typical dataset for document-level analysis. The term *document-level* is to distinguished from three other categories *paragraph-level, sentence-level and phrase-level.* The category is characterized by the syntactic structure of an instance. For example, the instance of a *sentence=level* dataset is a complete sentence. For *document-level*, an instance has various sentences. Some sentences are *subjective.* Subjective sentences express the opinion/judgement of author towards the topic. The other type of sentences is *objective.* An objective sentence is indeed a *context* sentence, providing plots of topics. In term of linguistics, the objective sentences complete the meaning of writing. However, in term of sentiment analysis, they are noise.

Here is an example of *negative review* from the corpus. This review was about the movie Soldier (1998/I), reviewed by Tim Voon `http://www.imdb.com/reviews/156/15606.html`. However, the review only contains raw text:

> "Numerous comparisons can be made with this movie to past sci-fi, suspense thrillers. Soldier is a multicrossbreed between the likes of Terminator, Aliens and offspring. The problem with such mixed genes is that the final product is a real mongrel not well made and should have been put down before production got off the ground. Besides this, the action is mediocre when compared to the standard action flicks of this day and age. The fight scenes between Jason Scott Lee and Kurt Russell seem laboured, slow and sluggish, and could have done with better choreography. Russell who is usually a good actor in B-Grade action flicks is unusually hampered by his character Sergeant Todd, who seems more like a Sergeant Toad. Besides having almost no dialogue, his character appears stunted, zombie-like which is in line with his screen persona, but scores little points of empathy with the audience. This movie has not made me change my opinion about director Paul Anderson, whose last epic Event Horizon has left an unusually bitter taste in my mouth. Although this movie does not come anywhere close to the strangeness of former, it is still a long way from anything considered desirable."

In the example, Tim Voon begins with an explanation about the content of movie, what is called "Soldier". Then he puts out his assessment about the fighting scenes, the characters. The review also mentions "Paul Anderson", the director, with a brief evaluation towards his career. Finally, Tim thinks this film "far from desirable".

Evaluation of the project is conducted by 10 fold cross validation. 10 fold CV is also the same evaluation method all other publications use. In my project, the dataset is randomly divided into 10 subsets, each subset contains 100 positive and 100 negative reviews. The performance of one fold is measured by accuracy. Then the average of all accuracy values is the final accuracy for 10

3 Feature Extraction and Selection

There are many practical researches to select the best feature set. But even the most complicated and gigantic set could be a poor solution. The procedure of getting a qualified set of features includes *feature extraction* and *feature selection*

3.1 Feature extraction

The project considered two subsets of features: sentimental words and ngram. Between them, the sentimental lexicon of words is considered the auxiliary part. The ngram is the main focus.

A list of sentimental words was retrieved. From a simple assumption that positive review contains positive words. On the contrary, negative review is expected to have negative words. An instance is characterised by the number of positive words and the number of negative words. Then, the relative frequency of positive and negative with regard to the document length are another two features. The odd value between these two is another more feature. Overall, it contribute to 5 additional features. As 5 is a small number, the feature selection section will ignore them. However, the sentimental words could become another source of feature set if each word becomes a features. I have not examined this possibility yet!

Bag-of-words is generally refered as an effective method in linguistics processing. The weak point is its lack of expression in the text alignment. For example, the expression "the movie is not good" has a *negative* meaning. However, all single terms does not indicate *negative*. A *ngram* may carrier the information that a word misses. In this project, 1,2,3-grams are taken into account. The chain of more than fourth gram seems to be too sparse. In fact an unigram is a single word. The collection of unigram is bag of words.

Given a ngram n, there are two states of an instance: *having* n or not having n. Then, the next problem is how to represent n as a value. Binary value is the simplest way. It is also proved to be effective. There are another two methods: term frequency and tf-idf. These two methods are actually identical if the learning algorithm requires normalization. There are several techniques to compute the tf-idf value, but the most famous are logarithm measurement. The tf-idf of a value is measure as follows:

$$TFIDF(n) = (1 + log_{10}(tf_n)) \times log_{10}(\frac{N}{df_n}) \tag{1}$$

Preliminary experiment with SVM has shown that tf-idf value does not overweighted binary value in feature representation. Even it does not reject the valuable of tf-idf value, the conclusion for this problem was drawn. The next phases (feature selection and learning methods) will be conducted on 0/1 value. Finally, with the best feature set and learning algorithms, another comparision between two representations (binary and tf-idf) will select the final classifier.

The dataset was randomly divided into 10 folds. Each fold consists of 100 positive reviews and 100 negative reviews. When doing cross validation, the ngram from 9 folds will be extracted. On average, the quantity of unigram is 48000, the quantity of bigram is 520000 and the quantity of trigram is 1440000. Figure 1 show the statistics of unigram and ngram in the dataset.

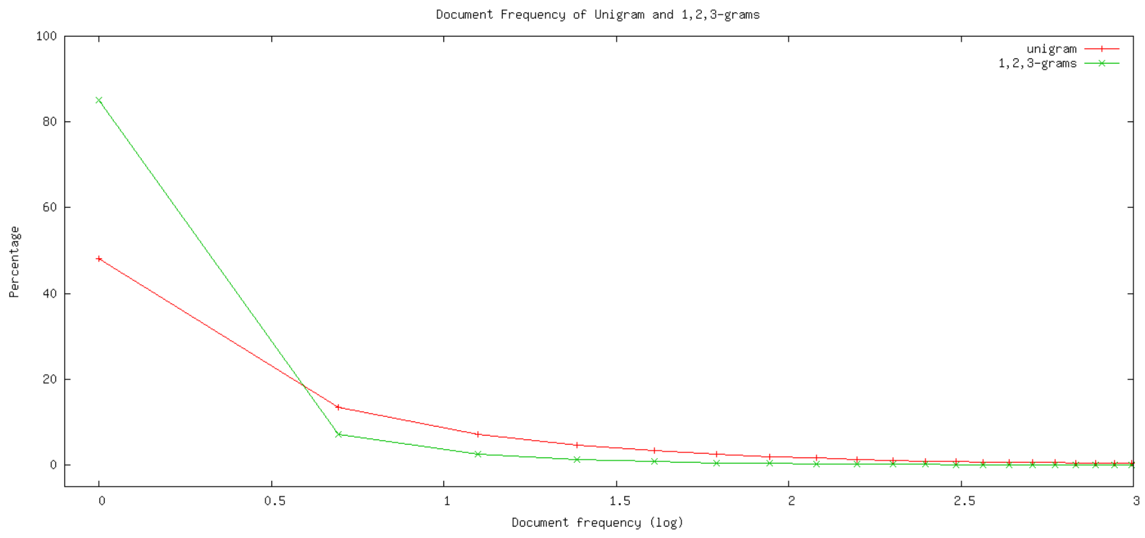

Figure 1. Document frequency of unigram and trigram

To examine the effectiveness of ngram over unigram, two sets of 60000 unigram and 2000000 ngrams are inputs of the feature selection. So far it is not a wise choice to select all 2 millions ngrams as the feature set of an algorithm. First, it leads to high cost of memory and time. Second, it is apparent over-fitting issue. The next step *feature selecion* is to select a robust and effective feature subsets. In general, the goal of selection phase is to minimize feature space but maximize accuracy.

3.2 Feature selection

When the selection process takes over, of its activities is to select an appropriate criteria. This project will take consideration of following methods in exploiting the valuable unigrams and ngrams.

If the system only consider unigram as the feature set, the space is relatively small. This project tests whether adding bigram and trigram would improve the performance of the whole system or not. The following methods were considered

- Document Frequency (DF)
- Information Gain (IG)
- Chi-statistics (CHI)
- Fast Correlation Based Filter (FCBF)
- Odd Value between positive and negative classes
- Principal Component Analysis
- Adaboost weighting
- SVM linear weighting

DF is the most straightforward method. If the term appear in a small number of training documents, it is less valuable. For example: if only one document has the term "bright-shine", then the term is considered *rare*. Probability that this term is in the testing data is almost zero. This method works by filtering out all features which have the DF smaller than a threshold. In case of ngram, this method remove 97% of features when the threshold is 5 (Figure 1). After this process, 60000 features remain for subsequent method.

Information gain is frequently employed as a term-goodness criterion. It measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document. In fact, IG could be referred as the mutual information between a feature and the label class. Mutual information is a criterion commonly used in statistical language modelling of word associations and related applications. It measures the information between two events share: how much knowing one of these variables will reduce the uncertainty about other variable.

$$IG(n) = H(class) - H(class|n) \tag{2}$$

So far the entropy of class $H(class)$ is constant. Selecting top k variables with highest IG is equivalent to selecting top k variables which lower the conditional entropy most. Another methods is to select the IG threshold. IG represented

After filtering all features of ngrams which have IG smaller than 0.002, there are 7109 remaining ngrams. The figure 2 below illustrates the IG in decreasing order.

CHI statistics measures the lack of independence between *class* and n and can be compared to the distribution with one degree of freedom to judge extremeness. Using the two-way contingency table of a term n and class, where A is the number of times n and class c co-occur, B is the number of time the n occurs without c, C is the number of times c occurs without n, D is the number of times neither c nor n occurs, and N is the total number of documents, the measure is defined to be:

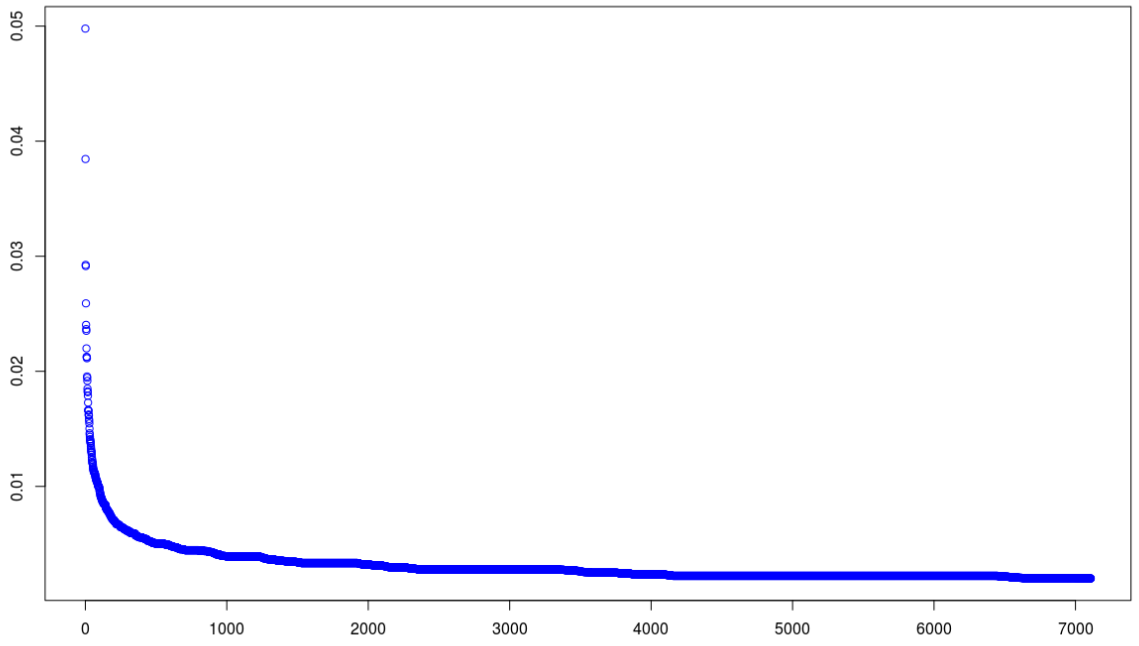

Figure 2. **IG** The information gain of 7109 features with $IG(n) > 0.02$

$$CHI(n) = \frac{(AD - CB)^2}{(A + C)(B + D)(A + B)(C + D)} \tag{3}$$

Obviously, CHI of n is 1 if they are identical and 0 if they are independent. Early experiment has pointed out that CHI does not outweighed IG when it comes to selecting the best ngram. Figure 3 shows the correlation between CHI and IG with regards to the class.

The correlation between information gain and CHI statistics shows the similarity between two measurements. Although there are some inconsistency, top 2000 highest features of chi statistics and information gain have achieved similar accuracy.

FCBF refers to an information theoretical algorithm. It has been proved to be effective in feature selection. It select the best subset of features according to their own principles, independent of outside size measures. Therefore, there are another version of FCBF called FCBF# which allows the control of feature space.

FCBF iteratively removes features when there is another feature (which is called predominance) which has a higher contribution to the class and two features are highly dependent.

A greedy algorithm that addresses the correlation between features [Fleuret and Guyon, 2004].

- First, it ranks the features according to their information gain so that $IG(A_1) \geq IG(A_2) \geq ... \geq IG(A_m)$.

- In the second step, it iteratively removes any feature A_k if there exists a feature A_j such that

$$IG(A_j) \geq IG(A_k) \quad \text{and} \quad I(A_k; A_j) \geq IG(A_k),$$

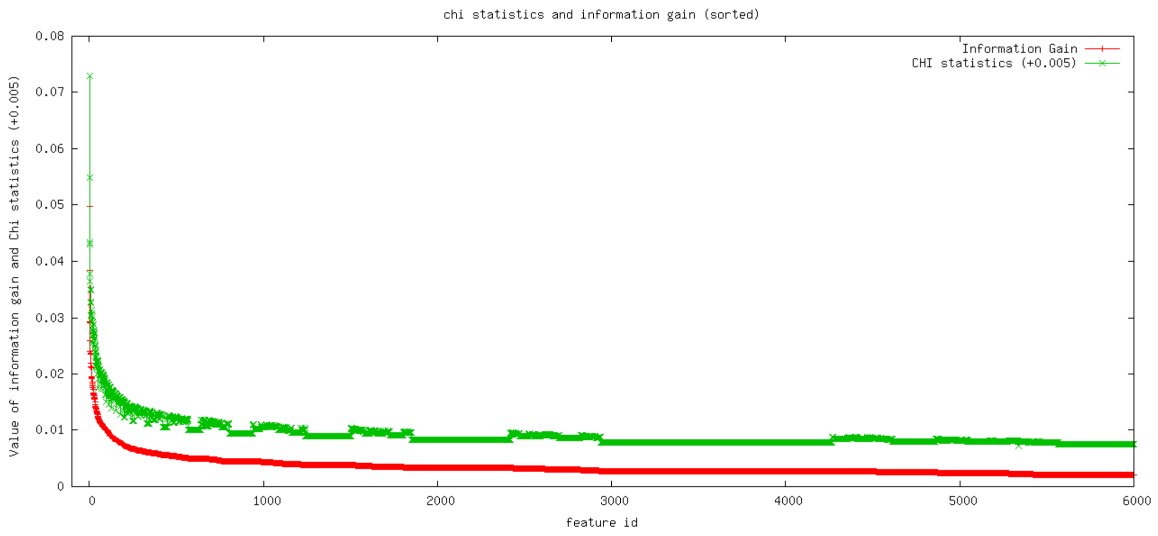

Figure 3. **CHI** Chi statistics and decreasing IG

which means that A_j is better as a predictor of C and A_k is more similar to A_j than to C.

Another greedy algorithm of FCBF [Yu and Liu, 2003]:

- First, it ranks the features according to their information gain so that $IG(A_1) \geq IG(A_2) \geq ... \geq IG(A_m)$.

- In the second step, it iteratively removes any feature A_k if there exists a feature A_j such that

$$IG(A_j) \geq IG(A_k) \quad \text{and} \quad I(A_k; A_j) \geq \frac{1}{2}(IG(A_k) + IGA_j),$$

which means that A_j is better as a predictor of C and A_k is more similar to A_j than to C.

Early experiments showed that two algorithms are both sensitive to the set of features. They achieved inconsistent performance in different folds. This paper refers to algorithm in [Fleuret and Guyon, 2004].

Principal Component Analysis finds valuable features by selecting the ones which have high variance or two features which are less dependent (small co-variance). Among various functions to do PCA, the *princomp* function in R was selected to do the PCA. However, experiment has shown that PCA is highly sensitive. The features and accuracy are not proportional or related.

In the experiment, PCA shows that it is not helpful in dealing with ngrams. The accumulation of first 800 features are 99%. However, the performance of these 800 features with SVM classifier are not good.

For a specific fold 2 (testing on the second fold and training on other folds), first 800 features achieves 78% of accuracy, the highest result was 91.5% with 1992 principal features. However, it does not indicates any reliable method to select a good set of features. The table belows shows experiments with PCA.

Number of features	*800*	*1990*	*1991*	*1992*	*1993*	*2000*
SVM	78%	85.5%	80.5%	91.5%	87%	85%

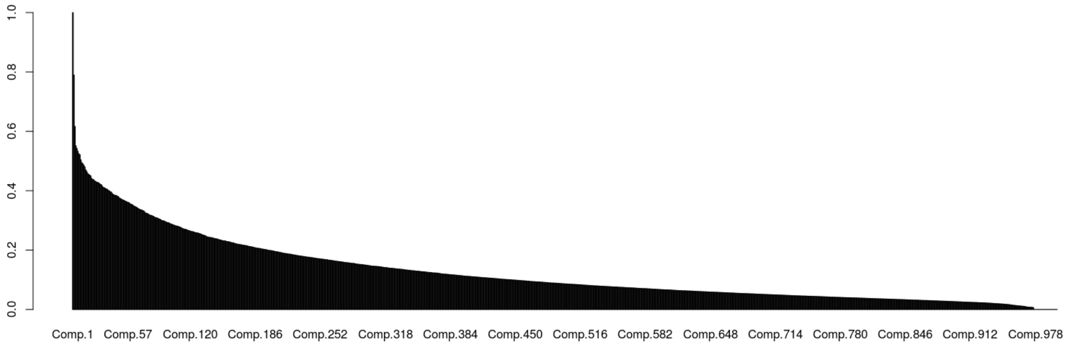

Figure 4. PCA Variances of each principal component

SVM weighting: after obtaining a linear SVM model. The weight w can be used to decide the relevance of each feature. The larger —w_i— is, the ith feature plays a more important role in the decision. Only w in linear SVM model has this indication, so this approach is restricted to linear SVM. In this project, we use package *e1071* to train the model with *linear kernel.* Then, the weighting is computed.

$$w = t(svmModel\$coefs)\% * \%svmModel\$SV \tag{4}$$

By ranking the absolute value of w, this method outputs a list of ordered features. The top k features will be selected based on preference. However, SVM linear weighting is designed for linear only. The experiments with other kernels do not achieve promising accuracies. The experiment with linear kernel itself is not promising as well. Therefore, the SVM weighting methods is evaluated as *inappropriate* for ngram and Polarity dataset.

Odd value represents how one feature contributes to a class agains the contribution to another class. The measurement of odd value is as follows.

$$Odd(n,c) = P(n)log\frac{P(n,c)+\lambda}{P(n,\overline{c})+\lambda} \tag{5}$$

The λ value is a smoothing parameter to avoid hard 0. However, initial experiments have shown that this method does not prove it effectiveness over other selection methods.

Adaboost feature selection is closely related to Adaboost learning methods. This method will be discussed in section of learning methods in more details.

3.3 Summary

Many experiments with DF, IG and TF-IDF does not achieve promising result. Finally, each feature set will be the result of a combination of several feature selection methods (a pipeline). One pipeline leads to a unique set of features. The list of feature sets are as follows:

1. **Uni-DF** Top 2000 DF of Unigram
2. **Uni-IG** Top 2000 IG of Unigram
3. **Uni-FCBF** DF threshold 5 of Unigram → IG threshold 0.02 → FCBF modification
4. **Uni-Odd** DF threshold 5 of Unigram → IG threshold 0.02 → Odd selection
5. **Ngram-DF** Top 2000 DF of Ngram
6. **Ngram-IG** Top 2000 IG of Ngram
7. **Ngram-FCBF** DF threshold 5 of Ngram → IG threshold 0.02 → FCBF modification
8. **Ngram-Odd** DF threshold 5 of Ngram → IG threshold 0.02 → Odd selection

4 Learning methods

This section is not designed to provides background about learning methods. Instead, it describes the behaviours of learning methods (including Decision Tree, Random Forest, Naive Bayes, SVM, Adaboost and voting method). The behaviour of a learning method includes result, running time issue, tuning parameters and other technical issues. Each learning method will be described in one subsections.

4.1 Decision Tree

Decision Tree (DT) is the most straightforward learning method. The best parameter *cp* is selected from *S1* is 0.0061.

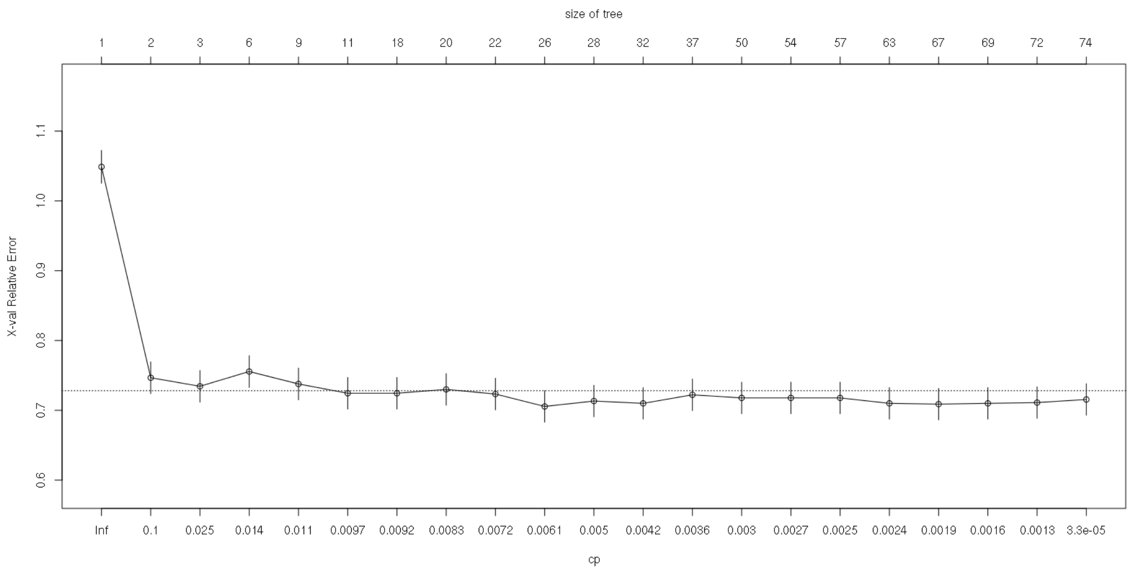

Figure 5. Tuned parameter *cp* in DT

The performance of DT regarding to each feature sets is presented on table belows:

Feature Set	*DF*	*IG*	*FCBF*
Unigram	64.5%	64.5%	64.5%
Ngram	65.5%	65.5%	**67%**

It can be seen from the table that the accuracy of feature sets filtered by DF and IG are identical. The main reason is that the default method of DT in spliting node is "information gain". Two methods lead to the same decision tree.

The performance of FCBF is higher than DF when it comes to Ngram. The t-test has pointed out that the improvement is significant with 95% of confident. The confident interval of the highest result is [65.920, 68.080].

4.2 Random Forest

Random Forest (RF) has two parameters the value of *mtry* and *ntree*, representing the depth of subtree and number of subtrees. The ranfomForest package in R has a function *tuneRF* to find the mtry value.

The performance of RF are as follows:

Feature Set	*DF*	*IG*	*FCBF*
Unigram	84.5%	85.5%	80.5%
Ngram	84.5%	**87%**	86.5%

In comparison with DT, Random Forest has proved its effectiveness. RF outperformed DT in all feature sets. The highest result of RF is the selection method by IG.

For RF and the polarity dataset, ngram gets a higher result than unigram. By filtering the top 2000 ngram which have highest IG value, RF get the accuracy of 87% and confident interval is [85.130, 88.870]

4.3 Naive Bayes

Naive Bayes is a straightforward but effective method. The Polarity dataset has the same numbers of positive and negative class. Laplace smoothing is not important in this task.

The performance of NB are as follows:

Feature Set	*DF*	*IG*	*FCBF*
Unigram	NA%	71%	72%
Ngram	**79%**	70%	73.5%

Although NB achieved higher accuracy than DT, it is concluded as not good enough for the task. Surprisingly, the highest result of NB is defined by the DF filtering.

Overall, Naive Bayes achieves better result than DT. However, it is not as good as RF. The plus point of NB is the speed of this algorithm is faster than RF.

4.4 SVM

According to other publications, SVM was so far the best learning algorithm for document-level sentiment analysis. This is the central method for the task.

The parameters of SVM are tuned based on grid search. Therefore, they are only a local optimal. For SVM algorithm, *polynomial kernel* is inefficient in the task of sentiment analysis. All dataset will be test with three *kernels*: linear, sigmoid and radial.

Feature Set	*linear*	*radial*	*sigmoid*
Unigram-DF	82.5%	87.5%	84.5%
Unigram-IG	82%	83.5%	85%
Unigram-FCBF	81%	86%	87%
Ngram-DF	81%	84.5%	85.5%
Ngram-IG	84.5%	87.5%	**89.5%**
Ngram-FCBF	87.5%	86.5%	86.5%

The highest result was achieved by information gain filter, 89.5%, confident interval is [87.289 91.711]. The FCBF modify has been sensitive. When testing with *linear* kernel, FCBF produced a good feature set with *ngram*. However, it is ineffective with *unigram*. Meanwhile, FCBF does not get a better result when testing with *linear* and *sigmoid* kernel. Due the the same nature of unigram and ngram, FCBF concluded as *inconsistent*.

The experiment also shows the difference between SVM and RF in favouring feature set. While RF achieves high result on ngram, SVM shows there is no significant difference between unigram and ngram if they are processed in the same method.

4.5 Voting by SVM kernels

The voting method require classifiers which get similar accuracy. Otherwise, if one strong classifier and one or more weak classifiers are embedded, the accuracy will drop. In that case, the weak classifiers become a burden.

In the project, voting method is based on SVM classifiers with three different kernels: *linear, radial and sigmoid*. Early experiment points out that *polynomial* kernel with degree greater than 2 are not suitable for the task. Three kernels have the same weight in a linear formula.

$$classify(n) = sign(\frac{1}{3}(SVM_{linear}(n) + SVM_{radial}(n) + SVM_{sigmoid}(n))) \tag{6}$$

in which $SVM_{kernel}(n)$ is the probability of SVM classifier with specific kernel when it predicts instance n.

The highest result is SVM voting technique is 87.5% when running with top 2000 features measured by information gain. Although t.test statistics reject the hypothesis that SVM classifier which achieves 89.5% is more significant than 87.5%, its accuracy is smaller indeed, its complexity is higher. SVM voting is evaluated as not promising.

4.6 Adaboost

Adaboost feature selection: AdaBoost stands for Adaptive Boosting. It is an algorithm for constructing a "strong" classifier as linear combination $f(x) = \sum_{i=1}^{t} \alpha_i h_i(x)$ based on "weak" classifier $h_i(x)$. Boosting applies the classifier repeatedly on the training data, then alter the adaptive weight for the classifiers.

In this report, I would like to focus on my experiment using package *adabag* in R. The detail description of algorithm was described in section 2.1 Boosting. In this packages, authors provide several versions of boosting algorithm (which are called AdaBoost.M1 and SAMME). Basically, two algorithms are the same, only the formula to update adaptive weights is different.

There are three functions with regard to the boosting method in the adabag package.

boosting function enables to build an ensemble classifier using AdaBoost.M1 or SAMME and assign a class for training samples. In this sample, value *importance* indicates the contribution of a variable (feature) to the classifier. The measure of importance takes into account the gain of the Gini index given by a variable in a tree and the weight of this tree in the case of boosting.

predict.boosting classifies data frame using a fitted model object of class boosting. It allows some new parameters to prune the ensemble. *boosting.cv* run v-fold cross validation with boosting. Therefore, cross validation can be used to estimate the error of the ensemble without dividing the available data set into training and test subsets. However, according to my point of view, this function is correct if and only if we finish the feature extraction part first.

The *importance* indication of package *adabag* is a method for feature selection. The algorithm exploits the best feature set based on its own criteria. Generally, around 40% of features will be evaluated *valuable* (characteristics by the *importance* > 0). The algorithm is non-deterministic, which makes it unstable. Besides, Adaboost selection is time and memory consuming. It is not able to process the number of vector higher than 2000.

Adaboost Running

Due to the technical problem of R memory, the method only runnable for less than 2000 features. Therefore, it was selected to built on top of other feature set. From the set of 1600 features which are the result of FCBF modification, the Adaboost algorithm is run. The result of Adaboost is 84% with default coefficient learning Breiman. This is 2.5% less significant than the original set of FCBF which achieve 86.5% with SVM.

Adaboost function evaluate 673 features as valuable. The new compact feature set achieve 83.5% of accuracy. Due to the cost of adaboost function, this method is evaluated as *inefficient.*

The poor performance of adaboost could be explained by the size of dataset. The core of adaboost ensemble method require a substantial amount of data. The polarity dataset contains 1800 training instances, which could be considered a small set. For adaboost, the data is divided into smaller set of separated data to build different classifiers. If the data is divided into three subsets for three classifiers, then each subset has 600 instances. The performance of classifier trained on the subset of data is undoubtedly lower.

5 Experiment summary

The result of experiment in both feature selection phase and learning method phase leads to several conclusions about the methods which have been used. Unfortunately, most of conclusion are negative.

First, the learning algorithms are classified into two type: ineffective and effective. The ineffective methods are *Decision Tree*, *Naive Bayes* and ADAboost. The simple methods do not achieve a high result when it comes to a task of document-level sentiment analysis. Ensemble method ADAboost is marked as *costly.* The effective methods are *Random Forest* and *SVM.* They achieve promising result, especially SVM.

Second, no method for feature selection is significant, compared to other methods. All methods were experimented are DF, IG, FCBF, Adaboost, PCA, CHI and SVM linear. In particular, PCA is not helping to deal with ngram selection.

Finally, the highest result is 89.5% with ngram selected by information gain value, the learning method is SVM. The confident interval is [87.289 91.711].

As mentioned in *feature extraction* section, the tfidf representation is used to replace the binary representation of the most successful feature set. Surprisingly, the accuracy of model with tfidf value is lower than binary value. It achieves 87.4% accuracy of 10-fold cross validation.

6 Related work

This section will provide the notable research publication researches closely related to the polarity dataset.

The dataset polarity review was first introduced by [Pang et al., 2002]. The first experiment was conducted on 3-folds cross validation of polarity review 1.0. The feature set was a combination of unigram, POS and position. The set was tested by three learning methods: NB, Maximum Entropy and SVM. The accuracy was reported as 81.6%. However, the paper does not provide details information how the feature was selected from unigram.

Most of approaches for sentiment analysis conducted on polarity dataset are statistical methods. An exception is [Taboada et al., 2011] where author presents a lexicon-based approach to extracting sentiment from text based on semantic orientation. Semantic orientation is a measure of subjectivity and opinion in text. They assign the semantic orientation value of each type of word, including noun, verb, adjective. The final value of the text is computed by combining the value of all tokens. Finally, this approach achieves 73.67% of accuracy. Based on this idea, [Taboada et al., 2009] extract 100 most positive and negative unigram features with SVM reach 85.1%.

[Kennedy and Inkpen, 2005] proposes two methods for determining the sentimen expressed by a movie review:

- The effect of valence shifters on classifying the reviews: negations, intensifiers and diminishers
- ML algorithm, SVM with unigrams then add bigrams (bigrams consists of a valence shifter and another word)

First, the positive and negative terms are initially taken from the General Inquirer(Stone et al.1966). Then, the review is considered positive if it contains more positive than negative terms, and negative if there are more negative terms. The valence shifters change the semantic orientation of another term, which change it value. For example: a *negation* preceded a positive word will change the word to negative meaning. Finally, the bigrams consisting of a valence shifter and another sentimental term are added to unigram to build SVM model. The result is 86.2% of 10 fold CV.

[Wang and Manning, 2012] proposes the use of bigram features in the sentimen classification task. This paper puts forward a method of SVM model using NB log-count ratios as features. They reported the accuracy of 10 fold CV are 89.45%. Another claim is that NB resolve the sentiment analysis task with short snippet better than SVM.

Using Appraisal Groups was introduced as a new approach [Whitelaw et al., 2005]. Appraisal theory is in the form of system networks denoted by a taxonomy of expressions. A full appraisal expression is a piece of text (usually a clause, but possibly larger) expressing appraisal of some sort. Appraisal expressions are the basic atoms for analysis of how attitudes are expressed in a text, and so extracting them is the basic task for appraisal analysis.It was reported to achieve the accuracy of 90.2% with 10 fold CV.

[Martineau and Finin, 2009] introduced a method in selecting and representing feature. Delta DFIDF was compared with traditional TF-IDF to see the advantages. In sentiment documents, sentimental words like “love”, “hate”, “good”, “bad”, “great”, and “terrible”, tend to be used in a large number of these documents, giving poor IDF scores. Simultaneously, the TF is low. Hence, TFIDF is not helpful. They reported the accuracy of 10 fold CV accuracy is 88.1% with initial feature set is unigram.

[Tu et al., 2012] example the support of convolution kernel in sentiment analysis. Convolution kernels support the modelling of complex syntactic information in machine learning tasks, but it is sensitive to the type and size of syntactic structure used. Authors claim that Pand and Lee 2004 use a flat feature vector (e.g. bag of works) to represent the documents, but it does not capture important information obtained from structural linguistic analysis of the documents. The paper studies diverse linguistic structures encoded as convolution kernels with the feature is the sequence of words, sequence of POS tags and combination of both. The final result of 10-fold cross validation is 88.50% bag of words features using vector kernel and POS replacement.

[Pang and Lee, 2004] is indeed a report of experiment that author performed in the branch of sentiment analysis and polarity review. It aims to remove objective sentences (such as plots summary in a movie review). The method is to define a *minimum cuts* in graph theory. The method is complicated using the subjectivity features, but it is reported to achieve 87.2%. Another experiment is the comparison between N-most subjective sentences, N-least subjective sentences, N-first sentences and N-last sentences. It shows that the last sentences are closer to the meaning of document than the first sentences.

[Ng et al., 2006] introduced a method in feature selection by weighted log-likelihood ratio. The ratio is measure as follows:

$$W(w_t) = P(w_t|c_j) * log\frac{P(w_t|c_j)}{P(w_t|\overline{c_j})} \tag{7}$$

Finally, 10000 unigrams, 5000 bigrams and 5000 trigram was selected to get 89.2%. Then they add polarity info of subjective, discard objective materials to get 90.50%. They collect the subjective materials like this: Construct a lexicon of positive and negative, generate bigram, unigram, take the intersection between two lists.

To sum up, there has been a number of approaches to solve the sentiment analysis task of Polarity dataset.

7 Conclusion

The report has provided my experience with polarity dataset in a binary classification task. It consists of many experiments in feature extraction/selection and learning mechanism. It also provides a valuable experience with over-fitting issue.

The task of sentiment analysis seems straightforward, but it is difficult to achieve. Human would easily detect the true sentiment of the review, but bag-of-features classifiers would presumably find the task difficult, since there are many words indicative of the opposite sentiment to that of the entire review.

In general, the results of SVM and RF are quite good in comparison to the DT baselines discussed in Section 4. In terms of relative performance, Naive Bayes tends to do the worst and SVMs tend to do the best. The Random Forest achieves promising result but it is good.

On the other hand, I were not able to achieve accuracies on the sentiment classification problem comparable to those reported in other publications. No method of feature extraction and selection has been proved to be superior. Both document frequency filter of unigram and ngram presence information turned out to be effective; in fact, none of the alternative features we employed provided consistently better performance once unigram and ngram presence was incorporated.

Finally, the best result which I get is 89.5% accuracy of 10 fold CV. It is selected by 2000 ngrams which have highest information gain. The learning method is SVM sigmoid kernel.

Acknowledgments

References

Francois Fleuret and Isabelle Guyon. Fast binary feature selection with conditional mutual information. volume 5, pages 1531–1555, 2004.

Alistair Kennedy and Diana Inkpen. Sentiment classification of movie and product reviews using contextual valence shifters. pages 110–125, 2005.

Justin Martineau and Tim Finin. Delta tfidf: An improved feature space for sentiment analysis. In *ICWSM*, 2009.

Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *ACL*, 2006.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

Bo Pang, 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*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Maite Taboada, Julian Brooke, and Manfred Stede. Genre-based paragraph classification for sentiment analysis. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '09, pages 62–70, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-64-0.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. volume 37, pages 267–307, Cambridge, MA, USA, June 2011. MIT Press.

Zhaopeng Tu, Yifan He, Jennifer Foster, Josef van Genabith, Qun Liu, and Shouxun Lin. Identifying high-impact sub-structures for convolution kernels in document-level sentiment classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 338–343, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Sida Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 625–631, New York, NY, USA, 2005. ACM. ISBN 1-59593-140-6.

Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. pages 856–863, 2003.