Sentiment Analysis and Emotion Classification of Movie Reviews

G1: Riki Saito, Sijie He, Weiwen Leung

Introduction

Sentiment analysis refers to the use of techniques such as natural language processing to extract subjective information, such as opinions and emotions, from texts. A seminal paper which arguably started this area is Pang, Lee and Vaithyanathan (2002); Pang and Lee (2008) provide a good review. We want to extract sentimental and emotional information contained within movie reviews using machine learning techniques. In particular, we will develop classification models to predict the sentiment polarity and emotional tendency of movie reviews.

We primarily rely on two datasets. The first is the Cornell Movie Review dataset, which contains labelled data: 1,000 negative and 1,000 positive movie reviews. The second dataset contains movie reviews which are labelled with one or more of eight emotions. This dataset is scraped by IMDB and annotated by UvA and NLeSC, and we use this for multi-label emotion classification. These datasets are also suitable for domain adaptation, which is a project extension we subsequently elaborate on.

Our report continues as follows: we describe our methods and algorithms, as well as experimental setup and analysis of results. Finally, we conclude and describe possible extensions.

Methods and Algorithms

In our project, we use two real-world movie review datasets for sentiment analysis and emotion classification. It is difficult to use the reviews directly. As such, we need to apply feature engineering and dimension reduction to generate a proper feature space for the movie reviews. After preprocessing, we introduce different classification algorithms to solve the two-class classification problem and the multi-label classification problem.

Feature engineering

First, we construct a term-review matrix and remove words of which the frequency is less than a minimum specified frequency and larger than maximum specified frequency. We consider such words as not containing effective information for the sentiment, and the remaining words are treated as the original features for each reviews.

Second, instead of using single words (unigrams) as features, one way we reconstruct the feature space is based on 2-gram features, which considers the feature as a contiguous sequence of 2 items from the given reviews. For example, we may consider “not good ” as one term instead of considering “not” and “good” as two different features, which will convey a more precise meaning.

Another way is to introduce the Term Frequency-Inverse Document Frequency (TF-IDF) matrix. The word frequency of each review is weighted inversely by the number of reviews the word appears in. In this way, the word that appears frequently, which may not have much contribution for the information, will have small weight.

Dimension reduction

Since each review contains hundreds of different words, the dictionary which covers all the words is large and the term-review matrix is sparse. There is much noise and redundancy in the original term-review

matrix, 2-gram matrix and TFIDF matrix. We applied several different dimensionality reduction algorithms to reduce the noise and redundancy in the original data. The methods considered in our project includes Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), Multidimensional Scaling (MDS), and Locally Linear Embedding (LLE). For the PCA and KPCA algorithms, the original data are projected to the dimensions which have the largest variances, while KPCA will project the data to high-dimensional space with kernel functions. MDS computes the low dimensional embedding that best preserves pairwise distances between data points. LLE computes the low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs.

Classification algorithms

We have two-class classification of sentiment polarity and multi-label classification of emotional tendency associated with the movie reviews.

For our two-class classification problem, we use two classical classification algorithms, Naive Bayes algorithms and Regularized Logistic Regression. Naive Bayes algorithm computes the posterior probability of both class for each review and chooses the class with highest probability. As for the Regularized Logistic Regression, we apply different regularized methods, such as L1, L2 and Elastic Net, based on logistic regression.

In the multi-label classification problem, prediction can be attained in three ways, the first two of which are classical approaches. In the first way, a probabilistic multi-class classifier (e.g. Naive Bayes) is used to obtain probabilities for each label. Decision on labels are determined at a specified threshold of the probability. The second way is the One-vs.-Rest (OvR) method. In OvR, independent binary classifiers are trained to distinguish one class from all others, and these classifiers are built for every class. This method is strictly for the multi-label classification. One possible extension to the multi-label classification is the RAndom K-labELsets (RAKEL). RAKEL consists of an ensemble of classifiers where for each classifier a small random subset of labels are considered as one class (PowerSet) and a binary classifier is learned to distinguish labels in the PowerSet vs labels outside of the PowerSet. Prediction is achieved by a ranking system across the ensemble of classifiers.

Experimental Setup

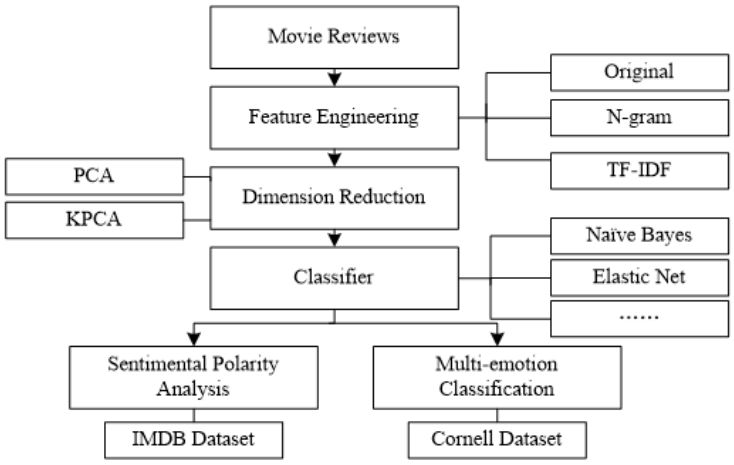

Figure 1: Experimental Setup

The experimental setup of our Sentiment Analysis problem and Emotion Classification problem can be seen in the above figure. The first step is to perform feature engineering by generating a set of features using the movie review text data. This is done in one of three ways (single-words, 2-grams, and TFIDF).

The feature set size is as follows:

Data Set	Sample Size	Single-Word Features	2-Gram Features	TF-IDF Features
Cornell	2000 (reviews)	13,290	21,944	13,290
IMDB	629 (sentences)	728	1,205	728

The second step is to perform dimension reduction on the feature sets generated previously. We ultimately proceeded with three methods (no reduction, PCA, and Kernel PCA). In the Kernel Principal Components, we used the radial basis function as the kernel, and kept all non-zero components. In general, the number of components kept for the Cornell data and IMDB data were 2,000 and 626 respectively. Between the first and second step, we generated a total of 9 different sets of features.

Finally, in the third step we applied several different classifiers. The methods that we proceeded with are Naive Bayes and Regularized Logistic Regression (L1, L2, and Elastic Net).

Evaluation

Our classification methods were evaluated on the testing data using a 5-fold cross validation. We primarily used two methods of evaluation. The first is the error rate, which is the proportion of inaccurate predictions to the total sample size. While this measure is simple and widely used, it is not the only measure of effectiveness of a classifier. Thus we also computed the F1 measure, which is the harmonic mean of the precision and accuracy. For the multi-label problem, a set of predictions (one for each emotion) was produced for each review, and evaluation measures were computed using all emotions.

Results

Problem	Classifier	Dimension Reduction	Feature Engineering	Training Error	Testing Error	Precision	Recall	F1
Sentiment Analysis	Naive Bayes	Kernel PCA	Single-words	0.067	0.082	0.920	0.916	0.918
	L1 Logistic Regression	Kernel PCA	Single-words	0.005	0.007	0.987	1.000	0.994
	Elastic Net (l1 ratio = 0.85)	Kernel PCA	Single-words	0.000	0.044	0.946	0.968	0.957

Emotion Classification (Multi-Label)	Naive Bayes	None	Single-words	0.104	0.219	0.321	0.380	0.348
	Naive Bayes	PCA	Single-words	0.179	0.197	0.351	0.330	0.340
	Elastic Net (l1 ratio = 0.15)	None	TF-IDF	0.186	0.177	0.375	0.232	0.286

The table shows the best three results from Sentiment Analysis and Emotion Classification. In Sentiment Analysis, we saw that the use of KPCA had the largest increase in the performance of the classifier among the dimension reduction methods. The effect of KPCA on the performance was also the largest on the single-word features. Among the candidates of classifiers, the L1 Logistic Regression had the best performance, achieving both the lowest error rate of 0.7% and the highest F1 measure of 0.994. In general, we were able to achieve an almost perfect classification of the sentiment in movie reviews.

In the multi-label emotion classification, we were not able to determine any one method as the best method. In general we were able to achieve an error rate around 17-20%, but our F1 measure indicate poor performance of our classifiers in identifying the true positives (i.e. predicting the presence of emotions correctly). Additionally, we saw that there was a trade off between error rate and F1 measure: in general, a decrease in error rate is correlated with a decrease in the F1 measure. However out of our best results, the Naive Bayes classifier was the most common classifier.

In general, we saw that for feature engineering, we found that the use of n-grams and TF-IDF did not improve our results. This could be because we did not consider the part of speech of each word. Previous results have shown that taking into account part of speech can improve performance, e.g. (Kouloumpis, Wilson, Moore, 2011). Dimension reduction does help improve the performance, but is highly dependent on the specific method of dimension reduction. Among the dimension reduction methods we have attempted (MDS, LLE, PCA, and KPCA), the use of Kernel PCA in sentiment analysis drastically improved the performance of the classifiers. However for emotion classification, Kernel PCA decreased the performance of classifiers.

Conclusions and Future Work

For sentiment analysis, we found that the Regularized Logistic Regression, in particular the L1 Logistic Regression, produced the best performance. However in emotion classification, the Naive Bayes classifier generally performed better than all other classifier candidates. As such, our results are consistent with Ng and Jordan (2002), who show that generative classifiers perform better in small samples, even though discriminative classifiers have better asymptotic performance.

Of our two problems, we found emotion classification to be much more difficult than sentiment analysis. There are a few possible reasons. The first is the relative size of the datasets. For the IMDB dataset, as our sample was too small, the classifiers we trained did not have the ability to distinguish between different emotions. The second is related to the inherent nature of the problems. Words generally describe either a

negative or positive sentiment, but they can describe multiple emotions, and these emotions are often highly correlated.

We have attempted different extension strategies for our project. For the sentiment analysis, we tried to use Support Vector Machine, but the results were poor because it set almost all the samples as support vectors. Dimension reduction only slightly improved the results. For the multi-label classification part, our plan was to use RAKEL, since RAKEL is designed to consider multiple labels together as one class. Unfortunately, since RAKEL relies on appropriate sizes of all k-label sets, and we did not have enough samples, the RAKEL approach performed poorly.

Future Work: Domain Adaptation

While sentiment analysis has become very popular in recent years, and it has been applied to many domains, standard techniques have their limitations. One key limitation is that the models trained on one domain may perform poorly in another domain. This is often because words can convey different sentiments in different domains; for example, "quick" may be a positive word in a restaurant review, but is often used negatively in electronics reviews (e.g. the battery wears out quickly).

Suppose we were provided with a dataset of reviews in related domains, such as reviews of comedy shows on TV. Manually labelling the dataset may be time consuming. Instead, one can adapt the classifiers we have created for our movie dataset.

In principle, we would adapt our classifier as follows (adapted from Wu and Huang (2016)). First, we would divide our movie dataset into genres. We would then train a global classifier for the entire movie dataset, as well as a "local" classifier for each movie genre. Thereafter, we would construct a domain similarity index between each of our existing domains and each of our new domains as follows:

$$R_{i,j} \equiv \frac{n_{i,j}^{S} - n_{i,j}^{O}}{n_{i,j}^{S} + n_{i,j}^{O}}$$

$$n_{i,j}^{S}, n_{i,j}^{O} = \text{frequency of words i and j sharing the same and opposite sentiment}$$

$$DomainSimilarity_{m,n} = \frac{\sum_{w=1}^{D} \sum_{v \neq w} \left| R_{w,v}^{m} + R_{w,v}^{n} \right| \cdot \min\{N_{w,v}^{m}, N_{w,v}^{n}\}}{\sum_{w=1}^{D} \sum_{v \neq w} (\left| R_{w,v}^{m} \right| \cdot N_{w,v}^{m} + \left| R_{w,v}^{n} \right| \cdot N_{w,v}^{n})}$$

($R_{i,j}$ is a measure of the extent to which words i and j convey the same sentiment in a given domain. It varies from -1 to 1; the more positive the value, the more often the two words convey the same sentiment. This measure is then used to calculate *DomainSimiarity*, which is essentially a weighted average of $|R^{m}_{i,j} + R^{n}_{i,j}|$, the absolute value of the sum of similarities of word pairs in different domains.)
Finally, we would adapt our classifiers according to the domain similarity index. For example, comedy TV shows would be much more similar to comedy movies compared to horror movies, so the local classifier for comedy movies would be weighted much more.

References

Buitinck, Lars, et al. "Multi-Emotion Detection in User-Generated Reviews." *European Conference on Information Retrieval*. Springer International Publishing, 2015.

Kim, Seungyeon, et al. "Beyond Sentiment: The Manifold of Human Emotions." *AISTATS*. 2013.

Kouloumpis, Efthymios and Wilson, Theresa, and Moore, Johanna, "Twitter Sentiment Analysis: The Good the Bad and the OMG!" Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011

Maas, Andrew L., et al. "Learning word vectors for sentiment analysis."*Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.

Ng, Andrew and Jordan, Michael. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". *Advances in neural information processing systems*, 2002.

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

Pang, Bo and Lillian Lee. "Opinion Mining and Sentiment Analysis". Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135

Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290.5500 (2000): 2323-2326.

Saul, Lawrence K., and Sam T. Roweis. "An introduction to locally linear embedding." *unpublished. Available at: http://www. cs. toronto. edu/~ roweis/lle/publications. html* (2000).

Tsoumakas, Grigorios, and Ioannis Vlahavas. "Random k-labelsets: An ensemble method for multilabel classification." *European Conference on Machine Learning*. Springer Berlin Heidelberg, 2007

Underhill, David G., et al. "Enhancing text analysis via dimensionality reduction." *2007 IEEE International Conference on Information Reuse and Integration*. IEEE, 2007.

Wu, Fangzhao, and Yongfeng Huang. "Sentiment domain adaptation with multiple sources." *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*. 2016.

Cornell dataset: (http://www.cs.cornell.edu/people/pabo/movie-review-data/)
Movie reviews, annotated for emotion classification:
(https://github.com/NLeSC/spudisc-emotion-classification/blob/master/README.rst)

Appendix

Table 1: Two class classification problem

			Training Error	Testing Error	Precision	Recall	F1
Naïve Bayes	Original		0.030	0.337	0.655	0.690	0.672
	2-gram		0.001	0.267	0.695	0.832	0.757
	TFIDF		0.008	0.351	0.638	0.690	0.663
	KPCA	Original	0.067	0.082	0.920	0.916	0.918
		2-gram	0.121	0.187	0.785	0.862	0.822
		TFIDF	0.254	0.361	0.731	0.442	0.551
L2 Regularized Logistic Regression	Original		0.000	0.160	0.843	0.835	0.839
	2-gram		0.000	0.148	0.855	0.848	0.851
	TFIDF		0.028	0.160	0.831	0.853	0.842
	KPCA	Original	0.000	0.499	0.501	0.800	0.616
		2-gram	0.000	0.499	0.501	0.800	0.616
		TFIDF	0.000	0.486	0.517	0.432	0.471
L1 Regularized Logistic Regression	Original		0.000	0.186	0.816	0.812	0.814
	2-gram		0.000	0.174	0.824	0.829	0.827
	TFIDF		0.209	0.247	0.726	0.812	0.767
	KPCA	Original	0.005	0.007	0.987	1.000	0.994
		2-gram	0.041	0.041	0.987	0.930	0.958
		TFIDF	0.333	0.428	0.571	0.577	0.574
Elastic Net	KPCA	Original	0.000	0.044	0.946	0.968	0.957

Table 2: Multilabel classification problem

			Training set error	Error Rate	Precision	Recall	F1
L2 Regularized Logistic Regression	Original		0.0369	0.171	0.400	0.220	0.284
	2-gram		0.0174	0.176	0.365	0.197	0.256
	TFIDF		0.1155	0.157	0.433	0.075	0.128
Naive Bayes	Original		0.1043	0.219	0.321	0.380	0.348
	2-gram		0.0582	0.197	0.342	0.304	0.322
	TFIDF		0.1009	0.217	0.319	0.366	0.341
	PCA	Original	0.1788	0.197	0.351	0.330	0.340
		2 gram	0.1765	0.196	0.340	0.292	0.314
		TFIDF	0.1524	0.175	0.314	0.119	0.173
	KPCA	Original	0.2793	0.490	0.147	0.454	0.222
		2 gram	0.2872	0.461	0.167	0.502	0.251
		TFIDF	0.4159	0.501	0.172	0.595	0.267
Elastic Net	Original		0.0702	0.211	0.308	0.296	0.302
	2-gram		0.2047	0.234	0.255	0.270	0.262
	TFIDF		0.186	0.177	0.375	0.232	0.286
	KPCA	Original	0.0015	0.154	0.500	0.032	0.061
		2 gram	0.0009	0.164	0.404	0.136	0.203
		TFIDF	0.001	0.154	0.412	0.009	0.018