CS 4705
HW2
MOVIE REVIEW CLASSIFICATION

Abhinav Saini (as3906@columbia.edu)

MACHINE LEARNING EXPERIMENTS ON A SET OF MOVIE REVIEWS

1. Background and Related Work

Turney (2002) found multi class movie reviews to be the most difficult. It was hypothesized that this was due to the tendency of reviewers to rate the individual elements of a movie differently from the movie as a whole within the same review, and addressing this issue remained a matter of future work. However, as Pang et al. (2002) suggest, the machine learning methods and features used when classifying movie reviews do not have to be specific to that domain.

Pang (2002) and Pang and Lee (2004) have compared the performance of various classifiers when determining the sentiment of a document, and also found that SVMs were generally the best approach. Unigrams, bigrams, part-of-speech (POS) tags and trigrams were used as features.

2. Problem Statement and Dataset

Classification Problem

1. 4-star rating Classifier
2. Positive Negative Classifiers
3. Reviewer Classifier Task

The 'movie-corpus' dataset consists of 5006 reviews which have been written by 4 authors. Three types of accuracy scores were obtained.

1. Accuracy scores obtained after building a model on the dataset and trying to fit the training data to that model
2. Accuracy scored obtained by performing a 10 fold cross validation
3. The original dataset was split by randomly selecting 4000 samples as training and the remaining 1006 as test samples.

3. Method

3.1 Classifiers Employed

1. **NaiveBayes** : This is a simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions (naïve assumptions) . This classifier was usually the fastest classifier and took very less time to build a model, usually less than a minute and gives good performance.

2. **J48 Decision Tree** : This classifier attempts to build a decision tree using the attributes of the available training data. Whenever it encounters a set of items it identifies the attribute that discriminates the various instances most clearly.
3. **Support Vector Machines** : Implemented using Platt's Sequential Minimization Algorithm, SVMs construct a hyperplane or a set of hyperplanes in a high dimensional space which separates the original data mapped into a higher dimension. SVM's give the highest accuracy in classification problems but are extremely slow and take a lot of time to build models (of the order of ~30mins to 1.5 hrs) Also, they require large amounts of memory and often the JVM running Weka used to run out of heap memory. It could handle upto 5000 features and 5000 instances or feature sets at most with 3GB of memory allocated to the JVM. Although SVMs are known to handle high dimensionality data well, the limitations arise due to weka running on java, which is slower. Implementations like SVMLight (Joachims) written in C/C++ tend to perform faster.

SentiWordNet : Available at (http://sentiwordnet.isti.cnr.it/) is a lexical resource for opinion mining. SentiWordnet assigns to each synset of **Wordnet** three sentiment scored : positivity, negativity and objectivity. The words from SentiWordnet having high positivity and negativity scores were extracted and used to build a dictionary of opinion words. The dictionary words were used as features for movie reviews, i.e., the number of times a word from the dictionary occurring in a particular review was a feature.

Data in SentiWordnet is in the form:

< POS>	<ID>	<PosScore>	<NegScore>	<SynsetTerms>	<Gloss>
<a>	<00026388>	<0.25>	<0.125>	<saltlike#1>	<resembling a …. >

3.2 Classification Process

1. **4 Star Rating Classifier Task** : This task was the hardest of all three classification problems, with the difficulty arising due to the subjectivity of the four reviewers and also the inability of the classification algorithms to comprehend the relationship between different classes (class 1 is closely related to 2, class 4 is not related to class 1 etc). Observed accuracies lied in the range of 40% to 50% with Cross Validation. Pang et all (2005) had achieved similar results with their best results being around 60% employing methods like 'Metric Labeling' and SVMs combined with heuristics like Positive Sentence Percentages.

   Features utilized for this task include :

   1. word counts of positve and negative words from SentiWordnet
   2. No. of positive words in the review / Total No. of words in the review
   3. No. of negative words in the review/ Total No. of words in the review
   4. Sum of the sentiment scores all the words in the reviews, positive words have been assigned a positive numerical value and negative words have been assigned a negative value.

   Support Vector Machines gave the best results in this case. The Polynomial and the Radial Basis Kernel with other parameters set as default were tried.

   The models built were able to determine the relationship between classes without explicitly having to code this fact. As can be seen from the confusion matrix, the main diagonal represents perfect matches and the entries adjacent to the main diagonals are those which were misclassified by an error margin of 1 class.

2. **Binary Rating Classifier** : This task was the easiest of the three tasks since the number of classes were just two rather than four. Since we were not required to provide fine grained results the number of errors that occurred was significantly lower. Also classifiers like SVMs which are primarily binary classifiers gave good accuracy results.
   The features collected the same as those for the 4 class problem. Accuracies of the order of 70% could be achieved using these methods.
   Unigrams and Bigrams collected from the whole dataset should be avoided since they tend to overfit the data and provide high accuracy on the training data when it is fitted to the model, but fail when an independent dataset is used.

3. **Reviewer Classification Task** : This problem was of intermediate difficulty with respect to the other problems of classification. The features employed were
   1. No. of characters in a review
   2. No. of sentences
   3. Avg. sentence length
   4. No. of words
   5. Avg. word length
   6. Lexical Diversity = (no of words in the set of words in the review)/(total number of words in the review)
   7. dictionary of words containing highly positive and negative words. This serves as a measure of what kind of words occur more frequently in the reviewer's vocabulary.

   This classification problem was different from the other two since here we had to look for characteristics of the reviewers than of the review per se. Features which distinguish one author from another are more important than those that distinguish one review from another.

Classifier Results

Task	Method	Feature	Feature Dimension	Accuracy	Validation/ Split	Time to build model (s)
Author	**SMO**	**Dict + Document stats**	**4816**	**86**	**10 fold CV**	**103**
Author	J48	Dict + Document stats	4816	76	10 fold CV	64
Author	NB	Dict + Document stats	4816	68	10 fold CV	9
Binary	**NB**	**Dict + Positivity/Negativity Scores**	**4811**	**68**	**10 fold CV**	**9**
Binary	J48	Dict + Positivity/Neg	4811	64	10 fold CV	1500

		ativity Scores				
4-class	NB	Dict + Pos/Neg Scores	1787	40	80:20 split	7
4-class	NB	Dict + Pos/Neg Scores	1787	38	10	6

4 class	**SMO**	**Dict + Pos/Neg Scores**	**1787**	**50**	**80:20 split**	**1740**

Q. Did early experiments guide your thinking for the final submission ? How ?

A. The features described in the above section were arrived at after playing around and observing the accuracy values with training, CV etc. Initially, features that I considered were: top 2000 and top 2000 bigram features occuring in the training corpus. They seemed to be giving good results with both a 66% split and 10 fold cross validation in Weka. Accuracies as high as 85% were observed. But when these models where tested on independent data (random sampling from corpus, with model not trained on it), they failed badly, giving accuracies in the range of 30-40 %. It was realized that taking the top unigrams and bigrams were overfitting the model to the dataset, producing high training set accuracy but low testing set accuracy. Later other features like those extracted from SentiWordnet were utilized.

External Resources used : SentiWordnet proved to be of great help with its positivity and negativity indices for each word. Further references etc, can be found in the next section.

Conclusions

The aim of this homework assignment was to build classifiers and models for classifying movie reviews from a given corpus. For each classification problem at least 3 different classifiers were tried and their performance recorded. Better performance may be achieved by incorporating more features like POS and even a larger sized dictionary. Also, if more context information may be utilized then the error rate arising out of usage of positive words in otherwise negative reviews can be reduced.

References

1. Bo Pang, Lilian Lee and Shivakumar Vaithyanathan. 2002. *Thumbs up ? Sentiment classification using machine learning techniques.* In ACL -02 conference on Empirical Methods in Natural Language Processing.
2. Bo Pang and Lilian Lee. 2005. *Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales*.In ACL-05 Proceedings of the 43rd Annual meeting on the Association for Computational Linguistics
3. A. Esuli and Fabrizio Sebastiani. 2006. *SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining.* Proceedings of LREC, 2006.
4. AdaBoost, http://en.wikipedia.org/wiki/AdaBoost
5. Huifeng Tang, Songbo Tan and Xueqi Cheng. 2009. *A survey on sentiment detection of reviews*. Expert systems and Applications, Elseiver-2009
6. Hang Cui, Vibhu Mittal and Mayur Datar, 2009. *Comparative Experiments on sentiment classification for online product reviews.* Proceedings of the National Conference on Artificial Intelligence, 2006
7. Peter D. Turney. 2002. *Thumbs up or Thumbs Down ? Semantic orientation applied to unsupervised classification of reviews.* Proceedings of ACL
8. *WordNet : A Lexical Database for English* (http://wordnet.princeton.edu/)