CS 446 – Sentiment Analysis

Ratish Garg(rgarg7)

Bhopesh Bassi(bbassi2)

Introduction

- In this project, we used state of the art NLP techniques and classifiers to classify IMDB movie reviews to positive or negative. We then compared performance of these algorithms.
- We obtained this data from Stanford artificial intelligence laboratory website.
- We have total of 50000 movie reviews, 25000 positive and 25000 negative. These 50000 reviews come divided into train and test sets of 25000 each. Both train and test set has 12500 positive and 12500 negative reviews each.

The Problem-comparative study

We used following classification algorithms and compared their performance.

1. Perceptron.
2. SVM with both linear and gaussian kernel.
3. Logistic regression.
4. Decision trees.
5. Naive Bayes with multinomial distribution.

Features

- Unigrams, bigrams and trigrams of all reviews in training set.
- Removed stop words.
- Total 4302432 words in vocabulary.
- Used TF-IDF weighting.
- Did feature selection with chi-square
    - On our laptops, SVM, Decision Trees take a lot of time to train in original ~4 million features space.
    - Naive Bayes and Decision Trees sensitive to a lot of (irrelevant) features.
    - Number of features to be used with a particular algorithm was tuned using 5-fold cross validation.

Parameter tuning

- 5-fold cross validation.
- Features size tuning
  - Top k relevant features selected using chisquare.
  - k in {all_of_vocabulary, 100, 500, 1000, 10000 and 100000}
  - For each k perform 5-fold cross validation.
  - k with largest average accuracy selected.
- Other parameters for different algorithms were tuned along with feature size.
- Algorithms and parameters to be tuned.
  - Perceptron: Learning rate, regularization(l1 or l2) and epochs
  - SVM: Kernel(Linear or RBF)
  - Logistic Regression: Regularization(l1 or l2) and C(inverse of regularization strength)
  - Decision Tree: Criteria(gini or entropy) and maximum depth.

Parameter tuning results

Classifier	5-fold CV Avg. Accuracy	Feature size	Other tuned Parameters
Perceptron	0.897	100000	Learning rate = 0.005, Regularization = L2
SVM	0.893	100000	Kernel = linear
Logistic Regression	0.913	100000	Regularization = L1, C = 100000
Multinomial Naive Bayes	0.836	10000	-
Decision Tree	0.676	500	Criteria = gini, Max Depth = 500

- Decision tree work best with just top 500 features.
- Naive Bayes needs to use 10 times less features than other linear classifiers.

Training and Testing

- Algorithms trained using best parameters found while tuning.

Classifier	Feature size	Precision	Recall	F-1 Score	Accuracy
Perceptron	100000	0.86	0.86	0.86	0.855
SVM	100000	0.88	0.88	0.88	0.883
Logistic Regression	100000	0.84	0.84	0.84	0.837
Multinomial Naive Bayes	10000	0.86	0.86	0.86	0.859
Decision Tree	500	0.73	0.73	0.73	0.73

Results discussion

- Mostly linearly separable in space of n-grams bag of word representation. All four linear classifiers gave accuracy of around 85%.
- Because the problem is well solved by linear classifier, using RBF kernel with SVM didn't help
- Decision trees are sensitive to lots of attributes and also to maximum depth. Only with 500 features and tree of maximum depth 500, we got accuracy > 70%.
- Verified our discussion in class that even though naive bayes assumptions don't hold in practice, it is still a very useful classifier.
    - Used 10 times less features and still gave comparable performance to other linear classifiers.

Future Work

- Explore neural networks for the problem and understand if increase in accuracy is worth the complexity.
- Explore techniques which can add/remove features as new data/information comes in.