Course Project: COMP 7745/8745
Due date: April 20, 2019

Predicting Ratings from Text

In this project, you will predict rating scores (1 – 5) from the text in online reviews. You are given a dataset with reviews and their rating scores. The task is to take as input the review text and predict what is the rating score for that review. You will experiment with several supervised learning algorithms using this dataset. It is recommended that you use python for this project

Overview

Dataset

There are 2 files, training.txt and test.txt. Each file contains review-text and the score separated by a tab. There are 10K reviews in training.txt and 1K reviews in test.txt. These reviews were taken from outlier detection datasets (http://odds.cs.stonybrook.edu)

Pre-Processing and Feature Extraction

You will need to first extract features from the review text. You will implement the bag-of-words model we discussed in class. To do this, you will need to use packages in python sklearn. These packages convert documents into vectors after pre-processing the document (e.g. removing stop words, etc.) automatically. You should use the TF-IDF vectorization here. **sklearn.feature_extraction.text.TfidfVectorizer.**

Supervised learners

You will experiment with the following learners. All of them have similar interface for learning and classification (e.g. fit() for learning and predict() for classification)

i) Neural networks (MLPClassifier in sklearn)
ii) Naïve Bayes (MultinomialNB in sklearn)
iii) Logistic Regression (LogisticRegression in sklearn)
iv) AdaBoosting (AdaBoostClassifier in sklearn)
v) SVM (svm.svc in sklearn)

Tasks to perform

i. Run 5-fold Cross Validation on the training.txt using the 5 learning algorithms. Report the average-precision, average-recall and average-F1-scores. You can do this quite easily using the cross_val_score() function in sklearn
   In each algorithm, try to explore different settings of the parameters to achieve best possible results (this step is largely experimental, try to automate as much as possible). Parameters that you should try to change include
   a. In neural networks change the number of hidden layers and number of units in each layer
   b. In SVMs, change the penalty parameter C and the kernel type
   c. In Adaboosting change the number of estimators (n_estimators)
   d. In Logistic regression change the penalty: L1 regularization that can also perform feature selection and L2. Also change the regularization strength parameter (C)

ii. Include some additional knowledge into your model. Specifically, not all words are useful in predicting rating scores. Words that express sentiments are more likely to be useful. Use the sentiment words in pos_words.txt and neg_words.txt to filter words from the review text, and then evaluate the algorithms once again. These sentiment words are taken from https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. What changes did you observe?

iii. Perform evaluation on the test dataset using the optimal parameter settings that were obtained from the training set. How did each algorithm perform? Report its precision, recall and f-scores. Which types of reviews were the hardest to predict? To answer this, for the best performing algorithm, compute precision and recall for every rating score separately (e.g. rating score 1 precision/recall, rating score 2 precision/recall, etc.).

iv. Discuss some ideas that could help improve your predictions (even if you did not implement this it is fine)

What to submit on dropbox?

1. A report that describes your experimental results (I would expect this to be around 2 or 3 pages at most)
2. Source code

Policies

Please do not plagiarize from the internet, etc. You can discuss with other groups but please do not share any code. Plagiarism is treated very seriously by the department and the university and are subject to harsh penalties.