CS598 - Project 4

Xiaoming Ji

Computer System

Hardware

- Dell Precision Tower 5810
- CPU: Intel Xeon E5-1607 @ 3.10GHz
- Memory: 32GB
- GPU: Nvidia GeForce GTX 1080 (2 cards)

Software

- OS: Windows 10 Professional 64bit
- R: 3.5.1
- R Packages:
  - text2vec_0.5.1
  - tokenizers_0.2.1
  - xgboost_0.71.2
  - glmnet_2.0-16

Overview

The goal of this project is to build a binary classification model to predict the sentiment of IMDB movie review.

The given movie review dataset file `Project4_data.tsv` has 50,000 rows (reviews) and 3 columns. Column 1 "new_id" is the ID for each review (same as the row number), Column 2 "sentiment" is the binary response, and Column 3 is the review.

The given dataset file `Project4_splits.csv` is used to split the 50,000 reviews into Three sets of training and test data. The dataset has 3 columns, and each column consists of the "new_id" (or equivalently the row ID) of 25,000 test samples.

The required performance goal is to achieve minimal value of AUC 0.96 over the three test datasets. For model interpretability, the vocabulary size should less than 3000.

For this project, I choose logistic linear regression with Lasso regularization model to select vocabulary and logistic linear regression with Ridge regularization model for training/prediction. Such approach achieved `0.97` AUC with only `1000` words in vocabulary.

Build Vocabulary

Building vocabulary is basically a feature selection process. I first build the vocabulary using all 50,000 reviews with tokenization process as:

- Filter out HTML tag.
- Remove stop words:

`i`, `me`, `my`, `myself`, `we`, `our`, `ours`, `ourselves`, `you`, `your`, `yours`, `their`, `they`, `his`, `her`, `she`, `he`, `a`, `an`, `and`, `is`, `was`, `are`, `were`, `him`, `himself`, `has`, `have`, `it`, `its`, `of`, `one`, `for`, `the`, `us`, `this`

- Extract 1 to 4 ngram terms

Then I prune this vocabulary with conditions:

- minimum 5 occurrences over all documents.
- minimum 0.1% of documents which should contain this term.
- maximum 50% of documents which should contain this term.

This gives me a vocabulary size of `29,035`.

I have tried several approaches to reduce the size of the vocabulary:

1. Normalized Difference Sentiment Index as described in A Quick R Demonstration on Sentiment Analysis and Textmining.
2. Discriminant Analysis for Two-sample T-test as illustrated by Professor. Feng.
3. Use (XG)Boosting to build a model on all review data and select features/terms using `xgb.importance()` ordered by Gain.
4. Use logistic linear regression model with Lasso regularization to fit the review data, then select the none-zero coefficient term (3200+). For these terms, sort by the absolute value of coefficient, then select the 1000, 2000, 3000 terms as vocabulary candidates.

My testing shows approach #3 and #4 work better than #1 and #2. #4 works extremely well and I can achieve above `0.965` AUC using only `1000` words in vocabulary. Adding more words results in bigger AUC (see the performance number in next section).

Note: I also tried tokenization with word stemming but got lower AUC with same vocabulary size. Thus, no word stemming is used.

Classification Model

For classification model, I used logistic linear regression with Ridge regularization. `cv.glmnet` is used to select the nominal λ to rebuild the model and make prediction.

Three datasets are tested on vocabulary of size 1000, 2000 and 3000 respectively. And I get the performance matrix as:

	VocabularySize:1000	VocabularySize:2000	VocabularySize:3000
Performance for Split 1	0.9700653	0.9808982	0.9835644
Performance for Split 2	0.9699608	0.9815026	0.9839045
Performance for Split 3	0.9694170	0.9815380	0.9837165

Computation time: 449 seconds

Improvement Discussion

Since I use all review data to build the vocabulary, the performance is expected to be better than real world case where we don't have label for test data/reviews.

By checking some misclassified reviewed, I could improve both the vocabulary and the model.

Considering the following misclassfied review,

> I thrive on cinema....but there is a limit. A NAME `isn't enough` to make A MOVIE!. The beginning of the movie puts us in a mood to expect the unseen yet. But we remain hungry (or angry..) till the end . Things are getting `so confused` that I admit that I DID NOT

> UNDERSTAND THE END or was there an end to this nonesense. The opportunity to make an outstanding movie was there but the target was totally missed. Next... "

"`isn't enough`" and "`so confused`" seem good candidates to expand the vocabulary.

To improve the model, more fine-tuned boosting and SVM models could give me better AUC performance (but slower to build).

Some reviews are hard to predict just by bag-of-word approach. Considering the following review:

> I'm relieved the later reviews have turned sour - reading all the positive feedback, I was starting to worry that my understanding of movies (and life) was completely different than everyone else's in the world. Everything in this movie rang false to me... the characters, the dialogue, the manipulative soundtrack, the corny narration, all of it. As each scene unfolded I kept thinking, "People don't act like this." It's relentlessly heavy-handed and maudlin. In a way I think the movie bullies you into liking it, or pretending to like it, because it's Serious and about Real People and confronts Issues. But man, it really did not work for me."

Larger ngram could help but may not feasible, more sophisticated model (for example: RNN/LSTM) could be used to achieve better results.