Yelp Review Based Prediction

Nguyen Duc Hien, Sherman Sze, John S Burke

Boston University CS 542: Machine Learning

August 13, 2017

Project Overview

- Yelp Reviews have both review text and rating stars
- Reviews are associated with one of five categories (1-5 stars)
- Predict rating of a review based on the text alone

Dataset

- Yelp Dataset Challenge
  - **4.1M** reviews by **1M** users for **144K** businesses
  - **11** cities in Germany, Canada, U.K and U.S.
- Reviews
  - **41%** 5 stars, **25%** 4 stars, **12%** 3 stars, **8%** 2 stars, **14%** 1 star
- Our analysis
  - **50,000** reviews for **restaurants** in Phoenix
- Structure
  - JSON file where each review is an object

yelp_academic_dataset_review.json

```
{
    "review_id":"encrypted review id",
    "user_id":"encrypted user id",
    "business_id":"encrypted business id",
    "stars":star rating, rounded to half-stars,
    "date":"date formatted like 2009-12-19",
    "text":"review text",
    "useful":number of useful votes received,
    "funny":number of funny votes received,
    "cool": number of cool review votes received,
    "type": "review"
}
```

Data Preprocessing

- Extract relevant information from JSON file
  - "stars": review rating
  - "text": review text
- Clean text
  - Remove punctuation, numbers and symbols
- Remove stop words
  - Words without information value (Examples: "the", "to")
- Balance data across ratings
  - 20% of samples from each class for a 5-class problem (1-5 stars)

TF-IDF & Document Term Matrix

- Term Frequency scales with term occurrence in a review
- Inverse Document Frequency shrinks across all reviews
- TF-IDF builds a matrix of input arrays associated with a rating

$$tfidf(t, d, D) = tf(t, d) * idf(t, D)$$

Where `t` is a given term, `d` is a review, and `D` is the collection of them. The `tf()` and `idf()` operations can range from raw counts to sundry normalizations on frequencies

Support Vector Machine (SVM)

- Use **80%** of data for training, **20%** for testing
  - Both training and testing sets have roughly equal amounts of samples from each class
  - For a 5-class classification problem (1-5 stars), 20% of samples from each class
- Linear kernel
  - Regularization parameter $C = 1$
  - Achieved **58.2%** accuracy
- Polynomial and Gaussian kernels
  - Was not able to complete due to computational constraints

Neural Network

- Same train-test split and data as SVM
- Multi-Layer Perceptron from `sklearn`
  - Sigmoid activation
  - Single hidden layer of 100 neurons because of computational cost
  - L2 regularization of 10^{-5}
- Just over 55% and up to 60% accuracy

Analysis and Comparisons

- Random guessing is 20% accurate
- SVM and MLP are up to 40% better
- This matches the results of recent papers

Future Work

- Word stemming
  - Reduce words to their root form
  - "Swim", "Swimming" and "Swimmer" all share the same root
- More computational power
  - Use more data (More than **4M** unused reviews) to train our models
  - Try more SVM kernels (Example: Gaussian kernel)
  - More layers and neurons in MLP Network
  - Tune SVM by varying parameters

Citations

- *Prediction of rating based on review text of Yelp reviews*, Channapragada & Shivaswamy
- *Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research*, Pedregosa et al.
- Image and Data from Yelp

Conclusion

Thank You

Contributions	Project Outline
Sherman	Yelp Dataset
SVM	Data Preprocessing
Data Processing	TF-IDF and DTM
Hien	Support Vector Machine
Research of Prior Experiments	Neural Network
Report	Prior Work and Comparison
John	Future Work
MLP & Data Processing	