Smart Recommender

Pooja Govindappa
Computer Science
University of Southern California
Los Angeles, USA
govindap@usc.edu

Suhas Udeda
Computer Science
University of Southern California
Los Angeles, USA
udeda@usc.edu

Vigneshwar Selvaraj
Computer Science
University of Southern California
Los Angeles, USA
vselvara@usc.edu

Limian Zhang
Computer Science
University of Southern California
Los Angeles, USA
limianzh @usc.edu

ABSTRACT

Founded in 2014, Yelp has accumulated massive amounts of data, with over 100 million reviews written about different businesses and over 80 million users using their services every month. It is almost impossible for users to manually sift through thousands of reviews to form an educated opinion about a business and it is equally challenging to recommend a business to a user, merely based on the business ratings. In this paper, we describe a prediction inference approach to solve this challenge, by incorporating textual user reviews and social network data into collaborative filtering (CF) algorithms. For our hybrid recommendation system, we first extract user preferences from reviews using sentiment analysis and map these preferences on a rating scale that is understood by existing CF algorithms. This is achieved by learning sentiment embedded vectors using Convolutional Neural Networks (CNN), through which we have tried to capture the semantic relationship between business reviews and ratings. We have also attempted to utilize the information gathered through social network analysis, by forming communities of users using the Clique Percolation Method. We have found that the results obtained by our Smart Recommender fares well in comparison to other baseline methods.

KEYWORDS

Collaborative Filtering, Sentiment Analysis, Convolutional Neural Network, Clique Percolation Method, Random Forest Classifier

I. INTRODUCTION

Not until long ago, the best way to find a business was through word of mouth. This idea found tremendous success in the internet world; with over 92% of the consumers reading online reviews before deciding to purchase a product or try out an establishment [1].

However, more often than not, user generated reviews suffer from inconsistencies and irregularities in reflecting the true value of a business. While user inconsistencies and personality bias are major concerns, people still largely view Yelp ratings as one of the success metrics of an establishment. Two users might have semantically similar reviews but drastically different ratings.

Yelp is not only a mammoth database of rating and textual reviews, but also a social network of reviewers with profiles and friends. This motivated us to explore different ways to rate a business: based on review text and network of friends.

II. RELATED WORK

A lot of work has been done in the area of textual data analysis [3] and social network analysis [5]. Dave et al. [4] developed a system to associate a sentiment score with the review text and to distinguish between positive and negative reviews. Rotimi et al. [5] mined the user's social network for information that would help to serve unique predictions about a his/her future reviews. Fan, Khademi et al. [7] discuss an approach to predict a restaurant's rating based on the review text alone. Our work is built upon the ideas presented in the above literature.

III. DATASET

We have used the dataset from the Yelp Dataset challenge 2018 [8]. There are a total of 69 states and 1111 cities. We computed the number of businesses in each city and identified the top 10 cities with highest business counts and have chosen to work with *Tempe*, *AZ*. Tempe has a total of 4492 businesses with an

overall review count of 182638. We filtered this dataset to extract only those businesses which have at least 5 ratings. A total of 3523 businesses have at least 5 ratings (number of ratings that users have given). A total of 179404 users have reviewed businesses in Tempe. A total of 145238 users have written at least 5 reviews and we have considered only these users.

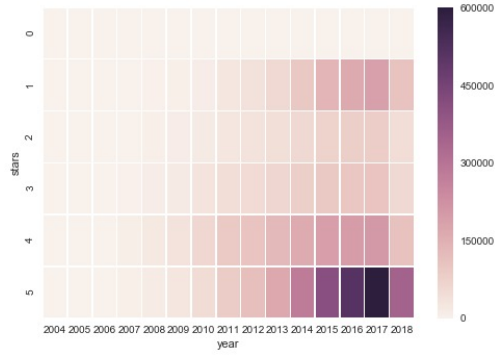

Figure 1: Ratings and number of ratings year wise

IV. METHODS

1) Baseline

Collaborative Filtering is commonly used for recommender systems. We have used the Spark library's implementation of this algorithm using Alternating Least Squares as a baseline for our review sentiment-based recommendation system.

2) Sentiment Analysis

a) Experimental Setup

For sentiment analysis, the data is presented as: 5-star reviews are used as a positive class and 1-star, 2-star reviews as the negative class. The 3-star and 4-star reviews are subjective with no definitive sentiment attached to them, hence are ignored. The positive class is assigned the value 1 and negative class is assigned the value 0.

b) *Method*

To perform the sentiment analysis on user review data, the model architecture, introduced in Kim et al. [9] is used. Here we consider $x_i \in R$ be the k-

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 500, 32)	3200000
conv1d_1 (Conv1D)	(None, 500, 32)	4128
max_pooling1d_1 (MaxPooling1	(None, 125, 32)	0
flatten_1 (Flatten)	(None, 4000)	0
dense_1 (Dense)	(None, 500)	2000500
dense_2 (Dense)	(None, 1)	501

Total params: 5,205,129
Trainable params: 5,205,129
Non-trainable params: 0

Figure 2 CNN Model Summary

dimensional word vector corresponding to the i-th word in the sentence. The method we adopted to obtain these vectors is described below: Each word is converted to a positive integer using Keras Sequence, which is a utility class for creating batches of temporal data. The number of words in each sentence are made equi-length by truncating the longer sentences and zero padding the smaller sentences. The sequences are then converted to vectors using Keras Embeddings, that turns the sequences into dense vectors of fixed size. The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used. The position of a word in the learned vector space is referred to as its embedding. A sentence of length n can now be represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$$

where $\oplus$ is the concatenation operator.

In general, $x_{i:i+j}$ refers to the concatenation of words x_i , x_{i+1}, . . . , x_{i+j}. This layer of vectors is used as the input to the Convolution Neural Network. A convolution operation involves the application of a filter $w \in R^{hk}$, applied to a window of h words to produce a new feature. For example, a feature ci is generated from a window of words

$$x_{i:i+h-1} \text{ by } c_i = f(w \cdot x_{i:i+h-1} + b)$$

Here $b \in R$ is a bias term, which is zero in our case and f is a non-linear ReLU function. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}\}$ to produce a feature map

$$c = [c_1, c_2, \ldots, c_{n-h+1}] \text{ with } c \in R^{n-h+1}.$$

After this, we apply a max-pooling function with pool-length 4 [10] over the feature map to get the

maximum value ĉ = max{c} as the feature for this filter. We have further applied the Flatten and Dense layers with ReLU and sigmoid activation. (Figure 2)

```
Train on 100577 samples, validate on 25145 samples
Epoch 1/3
100577/100577 [==============================] - 687s 7ms/step - loss: 0.1178 - acc: 0.9540 - val_loss: 0.0854 - val_
acc: 0.9691
Epoch 2/3
100577/100577 [==============================] - 420s 4ms/step - loss: 0.0417 - acc: 0.9860 - val_loss: 0.0906 - val_
acc: 0.9696
Epoch 3/3
100577/100577 [==============================] - 608s 6ms/step - loss: 0.0176 - acc: 0.9942 - val_loss: 0.1173 - val_
acc: 0.9692
Accuracy: 96.92%
```

Figure 3 Sentiment Analysis Results

3) Social network based

Social factors influence how people make decisions. We want to utilize this social network structure between people to help us predict ratings/ give recommendations.

The number of users in Tempe is approximately 200k and the edges between users are 400k. Since graph algorithms are memory intensive and given memory restrictions, we decided to run the algorithm on 40% of data. We calculate the predicted rating based on a weighted average of the ratings that a user's friends give.

$$r_{xi} = \frac{\sum_{y \in N} sim(x,y) \cdot r_{yi}}{\sum_{y \in N} sim(x,y)}$$

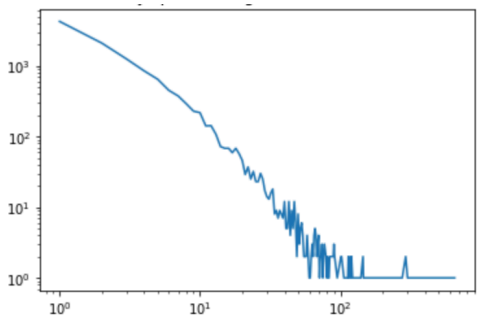

Figure 2 Degree Distribution Graph for our dataset

In order to compute this, we first use clique percolation (we use a 3 clique here) algorithm to compute the community clique that this user belongs to, this community then shapes the N close friends of this user x. For the similarity score between two users x and y, we can simply say all x's friends have equal influence, so the similarity will be 1.

4) Business Potential

In this section, we propose an approach to identify businesses which are performing well and the ones that are on the decline. This is especially useful for older businesses for which there's no direct way to quickly find their current standards from Yelp.

Ideally, a successful business should have both high ratings and known widely at least in its neighborhood. We have tried and tested with a variety of ratings and review combinations to classify a business as successful. We have used a pretty high standard of 4.5 average reviews and at least 5 reviews in the last year to define a business to have been successful. We found this combination to work well for the wide range of geographical regions encountered.

a) Experimental Setup

All user reviews except those given in the year 2018 is used as training data. We used a binary classification model and labeled the training data with 0's and 1's based on the established criteria for success. We set aside the reviews of 2018 for all businesses as test data to find the effectiveness of our method, and to find the best classification algorithm.

b) Features Used:

We extracted the below features to be used as inputs to train our classification models.

- Total number of all reviews given to each business ever.
- Average number of reviews given to a business each year and average rating of each business.
- Total number of years each business was able to maintain an average rating more than 4 with at least 3 reviews.
- Success of Failure label for each business for the last year in training data.

c) Methods:

We used the following methods to classify the business as successful or failure after training on the same dataset as described above.

a.1. Decision Tree Classifier:

We trained a decision tree classifier based on the training set and were able to predict the success/failure of test data with a good accuracy.

a.2. Random Forest Classifier:

We then used Random Forest classifier, an ensembled algorithm to predict the successful businesses. This classifier combines a set of weaker decision trees to form a stronger classifier. This algorithm uses averaging to improve predictive accuracy and control over-fitting to training data. Using this, we were able to classify the businesses with an improved accuracy than normal Decision Tree method.

a.3. Gradient Boosting Classifier:

The Gradient Boosting is another ensemble algorithm using multiple weaker decision tree classifiers. It adds new decision trees to the ensemble to optimize the loss function for the overall classifier. We were able to predict the successful businesses with a much better accuracy than both the Decision Tree and Random Forest classifiers.

Model	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.74	0.73	0.74	0.74
Random Forest	0.77	0.78	0.78	0.78
Gradient Boosting	0.81	0.80	0.81	0.80

5) Results

- Our baseline model returns an RMSE of 1.57
- The recommender we built using sentiment analysis of the review text achieved an RMSE of 1.48
- The recommendation system that takes into account the social factors achieved an RMSE of 1.16 (*note: we considered 40% random sample from Tempe*)

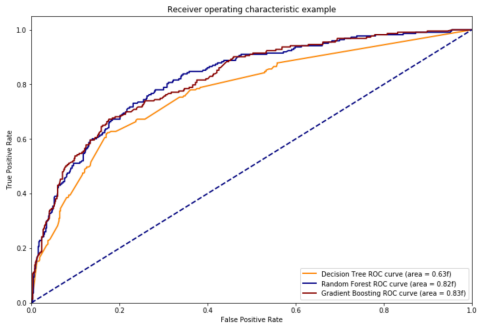

Figure 3 ROC Comparison

V. DISCUSSION AND NEXT STEPS

Our recommendation system primarily focuses on analyzing the review text and utilizing latent information from the social graph. As discussed above, we can observe that our recommendation system has better RMSE values in comparison with the baseline. In addition to achieving a low RMSE, we have also refined our recommendations to include only the businesses which show a positive trend and are found to be successful. Although this wouldn't directly reflect in RMSE improvement, this would certainly be a value add to the recommendation system.

As future work, we are also looking into vectorizing the reviews, finding the most important criteria for individual users, and pick the best restaurants matching their criteria. Since our work also involves identifying businesses that are doing well, we can also extend our recommendation system to provide insights to businesses about their performance. This would help businesses take necessary measures to rectify bottlenecks.

APPENDIX

Code Repository:
https://github.com/vigneshwarselvaraj/SmartRecommender

Contributors:
Pooja Govindappa – Sentiment Analysis
Vigneshwar Selvaraj and Suhas Udeda – Data Preparation, Baseline models and Business' Potential
Limian Zhang – Social Network Analysis

REFERENCES

[1] https://fitsmallbusiness.com/yelp-for-business/
[2] https://www.yelp.com/factsheet
[3] https://arxiv.org/pdf/1612.01556.pdf
[4] https://www.kushaldave.com/p451-dave.pdf
[5] http://snap.stanford.edu/class/cs224w-2015/projects_2015/Predicting_Yelp_Ratings_From_Social_Network_Data.pdf
[6] https://arxiv.org/pdf/1710.05978.pdf?fbclidIwAR2hE4n mDvzsuvtuVj8U vJd9H-4X5XC1iR5OAtsjbn0BTOcorMsH57H3LdE
[7] https://pdfs.semanticscholar.org/130e/cc92626b32b89a27 dbcda7357cd4b18abdc5.pdf
[8] https://www.yelp.com/dataset/challenge.
[9] https://arxiv.org/pdf/1408.5882.pdf
[10] https://arxiv.org/pdf/1103.0398.pdf