A Comparison of Graph-Based Recommendation System with Classical Recommendation Methods

Anshul Rawat
University of Southern California
anshulr@usc.edu

Arunkumar Rajendran
University of Southern California
arunkumr@usc.edu

Ragini Chugh
University of Southern California
ragini@usc.edu

Zhilin Xu
University of Southern California
zhilin@usc.edu

ABSTRACT

The goal of this project is to predict user and business connections in Pennsylvania with the Yelp dataset via a graphical approach and compare its effectiveness with existing recommendation system techniques such as model based technique, user based technique, item based technique, neural network technique and the BiRank method.

KEYWORDS

Graph, Random walk, LDA, Recommendation System, ALS, SVD, Neural Networks, User-Based, Item-Based

1 INTRODUCTION

For most people dining out is the only time of their week where they get to relax and unwind from their stressful daily life. So, dining with an unpleasant experience affects both the restaurant and the user. With traditional recommendation methods, it is difficult to work with sparse data and given precise item recommendation without consulting to a similarity metric, we hope to alleviate these problems and improve the solution with the graph based approach.

Section 4 explores classical recommendation systems and their performance on the Yelp dataset. These include a) kNN and tf-idf vector cosine similarity based prediction (Memory based technique) b) Singular Value Decomposition (Model based technique) c) Find similar businesses based on its attributes and the reviews that a business gets (Item based technique). d) Neural networks with auto-encoder. e) BiRank method.

Section 5 deals with our work of link prediction using graphical technique, random walk.

We will compare the results of all the mentioned approaches in Section 6

2 DATASET

We are using the Yelp dataset from the Yelp 2018 dataset [4] challenge. By examining the data, we found majority of the users visits restaurants within the same state and restaurant business type by far has the majority number of business within the yelp dataset. So we decided to performed our experiment using review data, user data, and business data from the state of Pennsylvania (PA) only. The PA dataset is the 5th largest by state. This makes it a good representation for the whole data.

For the PA dataset, we first split the review data into a training set, validation set and test set by a 80-10-10 split on user review date, ordered chronologically with training set contain the oldest user review and test set containing the most recent review. Business dataset and user dataset are filtered accordingly based on matching userID and businessID to that within the split review dataset.

3 PRELIMINARY FINDINGS

On inspecting the review data for Pennslyvennia, we found there exists 3357 unique businesses and 11915 unique users, which suggested users may commonly share multiple visited businesses. This led us to believe user-based approach and analysis on latent factors in review text to extract this relationship can model this pattern.

For business data in train, after converting JSON data to csv format, we see the table is rather sparse over a number of attributes. In order for item-based approach to work, we had to look at each attribute and determine if that attribute has a even divide across the business dataset. Finally, we decided on 16 business attributes to use for model our data and predicting new data. Imputation for missing data is done by randomly picking True/False values for True/False attributes and converting categorical attributes to numerics of 1 or above and setting missing attribute cell to a value of 0.

4 CLASSICAL RECOMMENDATION METHODS

4.1 Model Based Technique

In a model based technique, we try to learn the latent vectors of each user u_i and each business b_j. The predicted rating for a user business pair will be $r_{ij} = u_i^T.b_j$. We minimize the squared error of the rating and the prediction. Since most user-business pairs do not have any rating, the ratings matrix will be very sparse. In this case, an approach like Singular Value Decomposition (SVD) will be doing extra computation which is unnecessary. We will use the Alternating Least Squares (ALS) method to optimize the vectors u_i and b_j. We will also add a regularization parameter λ to control the weights of all the vectors. The equation to minimize will become,

$$\min_{u,b} \sum_{(u,i)} (r_{ij} - u_i^T.b_j)^2 + \lambda(|u_i|^2 + |b_j|^2)$$

We use spark MLlib to learn the vectors for users and businesses. We achieved a rating prediction RMSE score of 1.39.

4.2 User Based Technique

In user-based approach, we find top k=10 similar users to user u_i and averages similar user's business attributes as our prediction value. To capture the features within review text, a tf-idf vector of

500 features is computed for review texts of each user from review training data with the users review rating, "xstars" appended to review text as hint of similar review ratings. Stop words were stripped using nltk library. Similar users were found using cosine similarity score between user's tf-idf vector **??**.

The 16 business attributes with even split of the business dataset we used are Ambience - casual, Bike Parking, Business Accepts Credit Cards, Business Parking - street, Has TV, Restaurants Delivery, Good For Meal - breakfast, Good For Meal - brunch, Good For Meal - dessert, Good For Meal - dinner, Good For Meal - latenight, Good For Meal - lunch, Stars, Alcohol, Noise Level, and Price Range.

Using the average of user's nearest neighbors' business attributes as our prediction, we obtained A RMSE vector for train, test, and validation set. This is shown in Table 1. Hit ratio and average RMSE after normalizing all values to 1 for test set is shown in Table 4.

4.3 Item Based Technique

In the item based technique, we we find the similarity between the businesses based on the 16 business attributes. Cosine similarity is used to compute the similarity score of each business-business pair. After computing a Similarity Matrix with the above given approach, we predict the rating a user would give to a business. For a user u_i and business b_j, we predict the rating given by user u_i to the business b_j, $r_{i,j}$ as follows

$$r_{i,j} = \sum_{(k)} (s_{j,k} * r_{i,k}) / s_{j,k}$$

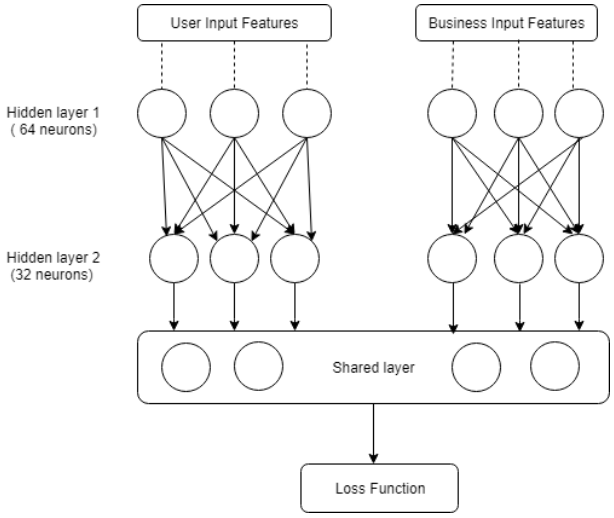
where k is a business in the intersection of the set of businesses rated by user u_i and the businesses similar to business b_j. In case we have a new user(who hasn't rated any business), we provide the rating by multiplying the average rating (3.0) with the mean of the similarity scores of it's top 10 similar business. In the case where the user exists, but the business hasn't received any ratings before, we assign the mean of the ratings given to all the businesses by the user. The RMSE value and Hit ratio for the test set are provided in the Table 4. Besides predicting the rating that a specific user might assign to a business, and the RMSE values for each dataset is given in Table 2.

4.4 Deep Neural Network using Reviews

A large amount of information can be deduced from how user has reviewed the businesses. It reflects aspects of a business that are appreciated or disliked which in turn affects the rating of a business. In this approach, we thus modelled users and business with their respective attributes alongwith the reviews. To encode reviews, we ran LDA (in section 6.3) and modelled each review into a normalized vector of length 20. The feature vector for user is comprises of a total of 32 attributes - compliment_photos, compliment_list, compliment_phots, topic_1, topic_2, topic_3, topic_4 etc upto topic_20. Similarly each business is modelled with 337 attributes such as Ambience - casual, Bike Parking, Business Accepts Credit Cards, Business Parking - street, Has TV, Restaurants Delivery, along with a topic vector of size based on LDA categories on reviews for business. Additionally, the business categories such as Coffee, Spanish are also represented in the form of one hot categorical representation. Each of the missing field in user or business features is imputed with a value of 0 and all the text-categorical field values are enumerated with numerical values.

4.4.1 Architecture. The proposed neural architecture trained to predict the star rating of a business by a user consists of two parallel neural networks coupled on the last layer which shares the output of these to produce the resultant star prediction. The first neural network takes user features with two hidden layers of 64 and 32 neurons respectively. Similarly, the second one takes business features as input with the same hidden layer composition. The output produced by these two networks is fed to a shared layer of 32 neuron and finally a softmax activation is applied to produce an output vector of size 5 depicting the probabilities for rating from 1 to 5. The highest probability in the vector is considered as output rating.

Figure 1: Neural Network Arhitecture



We also predicted other attributes for the business such as GoodForMeal.breakfast, GoodForMeal.brunch,Ambience.casual etc with a single neural network of two hidden layers of 64 and 32 neurons each followed by sigmoid activation to produce a probabilistic value from 0 to 1.

4.5 BiRank

BiRank [3] is a novel ranking and recommendation method that can model the relationship between user and businesses as a bipartite graph structure. It takes into account prior information about the users (ratings given to different businesses). BiRank iteratively assigns scores to users and businesses and then converges to a unique stationary ranking. This method can be easily extended to an n-partite graph. We tested this method as a comparison for our Random walk based method. The BiRank convergence formula is:

$$p_j = \sum_{i=1}^{|U|} w_{ij} u_i; \qquad u_i = \sum_{j=1}^{|P|} w_{ij} p_j$$

We tested the claims of the paper with the Yelp dataset. Unfortunately, we could not replicate the results that were provided in

the paper. The ranking predictions per user were getting a hit ratio of 0. Since the authors have not given the exact parameters they have used for training and testing their dataset, we are unable to get better results with this method.

5 GRAPH RECOMMENDATION USING RANDOM WALK

5.1 Overview

User-item recommendation can be modeled by random walk **??**, which uses transitional probability from current node to his neighboring nodes. The random walk method gives prediction on exact item hits from it's list of visited vertices as opposed to traditional methods' dependency on similarity score evaluation between their predicted value and the train corpus.

5.2 Approach

The approach for our random walk is to build a directed tripartite graph of 3 layers, users, LDA review topics, and businesses using train dataset. The first step is to build user->LDA topics->business relationship. To do so, top 3 LDA scores for each user u_i, business b_j are used as edge weights $w_{k,i}$ and $w_{k,b}$ connecting the graph. To permit exploring, edges from review topic to user of weight $w_{k,i}/2$ and edges from business to review topic of weight $w_{k,b}/2$ are also added.

The graph also displays the importance of direct user->business relationship by having a directed edge from user to business weighting $1/count(r_{i,b})$ where $count(r_{i,b})$ finds the number of businesses the user has reviewed on.

Finally, to build business<->business relationship, business to business edges are added from every business nodes to its top 50 cosine similarity business candidates with a weight of $(1 - cosine_similarity)/S$ where S is an adjusted hyperparameter, 10 in this case, to prevent overshadowing other edge weights. The example graph is shown in Figure 2.

5.3 Finding LDA topics

We find the top 20 topics in the review corpus using the Latent Dirichlet Allocation (LDA) [1] method. Latent Dirichlet Allocation (LDA) is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

Our goal is to use the LDA classified latent topics as the hidden layer in our tripartite graph, capturing topic relevance between user and business.

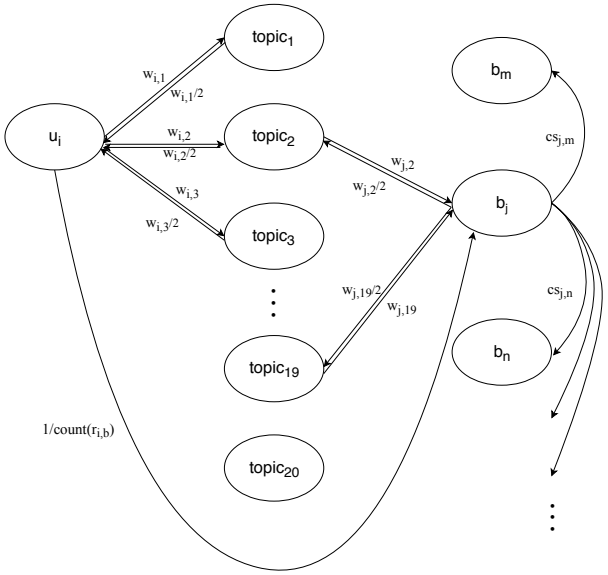
5.4 Execution

The random walk was performed on the test dataset with a walk step of 5000, a random restart alpha of 0.05. The results are captured in Table 4. Hit ratio are computed similarly to traditional methods.

6 RESULTS

For traditional methods, we find the top 50 similar business to the predicted score for each review within test based on cosine similarity between true label and predicted candidates. We record a hit of 1 if the targeted business id appears in the top 50 candidate and 0 otherwise. Finally, we average the number of hits for the test dataset and record it as our hit ratio.

Figure 2: Random Walk Graph Layout



From Table 4 we find that random walk did not have a clear advantage over traditional methods such as user based or item based. However, it is better than BiRank and SVD methods we explored. We believe several reasons are the cause of this. First, random walk algorithm depends heavily on graph setup. A tripartite graph may present obstacles for the random walker to reach its designated business nodes given there are too many out-degree edges from any given LDA topic nodes. shows a spread of each LDA topic with the total number of user who has the highest LDA score on that topic, similar trend is found for business too.

The alpha value which controls LDA deviation, as suggested by gensim's LDA model, proved to be very difficult to adjust as topic specificness decreases as we increase alpha value and distinctiveness decreases as we decrease alpha value.

More over, we believe a tripartite graph may not be able to fully capture the transitional probability from user->LDA topic->business as each user and their targeted business may have different LDA score peaks causing the random walker to hit irrelevant businesses.

7 FUTURE WORK

We would like to explore the BiRank method further to evaluate the claims made in the paper. The authors of the BiRank paper are not clear on how they have implemented their methods, so it would be interesting to find out their evaluation methods to compare with our own methods. For user-based method, we could use a sentiment survey result together with user text to build user profiles to help determine similar users. For random walk method, the tripartite graph could be improved by adding more layers such as user-group and business regions. For item-based method, we could

Attributes	$RMSE_{train}$	$RMSE_{val}$	$RMSE_{test}$
Ambience.casual	0.3184	0.5699	0.5769
BikeParking	0.2674	0.5011	0.5186
AcceptsCreditCards	0.1586	0.2684	0.2691
BusinessParking.street	0.2947	0.6012	0.6048
HasTV	0.3254	0.5777	0.5787
RestaurantsDelivery	0.2806	0.4848	0.4913
GoodForMeal.breakfast	0.2127	0.3659	0.3606
GoodForMeal.brunch	0.2155	0.3949	0.4114
GoodForMeal.dessert	0.1773	0.3202	0.3289
GoodForMeal.dinner	0.3157	0.5549	0.5578
GoodForMeal.latenight	0.2369	0.3776	0.3777
GoodForMeal.lunch	0.3355	0.5832	0.5846
Stars (max 5)	0.3883	0.7117	0.7100
Alcohol (max 3)	0.6886	1.2035	1.2203
NoiseLevel (max 4)	0.6104	1.1024	1.1462
PriceRange (max 4)	0.3933	0.7023	0.7430

Table 1: User-Based Business Attribute Prediction Errors

Attributes	$RMSE_{train}$	$RMSE_{val}$	$RMSE_{test}$
Ambience.casual	0.5952	0.6375	0.5981
BikeParking	0.4881	0.6832	0.4860
AcceptsCreditCards	0.3415	0.3121	0.3441
BusinessParking.street	0.5516	0.5835	0.5504
HasTV	0.6632	0.5206	0.6631
RestaurantsDelivery	0.4936	0.5001	0.4965
GoodForMeal.breakfast	0.2614	0.2731	0.2625
GoodForMeal.brunch	0.2473	0.2659	0.2495
GoodForMeal.dessert	0.1776	0.1897	0.1795
GoodForMeal.dinner	0.5738	0.6094	0.5748
GoodForMeal.latenight	0.2222	0.2344	0.2230
GoodForMeal.lunch	0.6086	0.6442	0.6091
Stars (max 5)	0.7942	0.7608	0.8049
Alcohol (max 3)	1.2510	1.2735	1.2501
NoiseLevel (max 4)	1.2301	1.1627	1.2273
PriceRange (max 4)	0.9118	0.8500	0.9221

Table 2: Item-Based Business Attribute Prediction Errors

use the review texts for the businesses as an additional measure of similarity.

8 CONCLUSION

To conclude, link prediction proves to be a very challenging task, random walk based link analysis did not seem to merit greatly over traditional collaborative filtering recommendation methods.

REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. "Latent Dirichlet Allocation." Journal of Machine Learning Research, 3:9931022, January 2003.

[2] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM '17)

[3] Xiangnan He, Ming Gao Member, IEEE, Min-Yen Kan Member, IEEE and Dingxian Wang. "BiRank: Towards Ranking on Bipartite Graphs"

Attributes	$RMSE_{train}$	$RMSE_{val}$	$RMSE_{test}$
RestaurantsDelivery	0.4936	0.5001	0.4965
GoodForMeal.breakfast	0.261	0.273	0.262
GoodForMeal.brunch	0.28	0.28	0.2799
GoodForMeal.dessert	0.26	0.189	0.179
GoodForMeal.dinner	0.2694	0.6094	0.5748
GoodForMeal.latenight	0.201	0.1639	0.1633
GoodForMeal.lunch	0.27	0.64	0.61
stars (max 5)	0.40	0.39	0.39
Alcohol (max 3)	0.70	2.86	2.6
NoiseLevel (max 4)	0.29	3.66	3.57

Table 3: Neural-Network Business Attribute Prediction Errors

Method	Prediction Score (RMSE)	Ranking (Hit Ratio)
Model Based	N/A	0.57%
User Based	0.4176	5.188%
Item Based	0.3226	5.369%
Neural Network	0.644	N/A
BiRank	N/A	0.0%
Random Walk	N/A	4.870%

Table 4: Prediction and Ranking results comparisons

Topic	Top Words	Percentage
1	order,go,food,time,...	15.6
2	place,great,service...	14.9
3	good,try,love,...	14.4
4	restaurant,steak,dinner...	10.0
5	mushroom,garlic,downtown,...	6.6
6	sauce,bread,fresh,...	3.8
7	dish,entree,waiter,...	3.6
8	chicken,fry,food,...	3.3
9	coffee,business,work,..	3.1
10	table,room,party...	2.9

Table 5: LDA Results

[4] https://www.yelp.com/dataset

[5] Liben-Nowell, D., and Kleinberg, J. (2003). The link prediction problem for social networks. Proceedings of the Twelfth International Conference on Information and Knowledge Management - CIKM 03. doi:10.1145/956863.956972

[6] Maria, T., and Matthew R. (2014). Text-based User-kNN: measuring user similarity based on text reviews. UMAP 2014: User Modeling, Adaptation, and Personalization pp 195-206

APPENDIX

Github link to source code and results: https://github.com/uyuyuyjk/inf553