Friend Suggestions to Yelp Users based on Similar Interest*

Monisha Shivarudrappa
University of Southern California
shivarud@usc.edu

Sanjana Srinivasa
University of Southern California
srin317@usc.edu

Shravya Gorur Sheshadri
University of Southern California
gorurshe@usc.edu

Vijetha Parampally Vijayakumar
University of Southern California
vijethav@usc.edu

ABSTRACT

User's circle of friends influence decisions on services they would want to take. We are more likely to avail a certain service previously used and liked by our friend. Therefore, it becomes more and more important to identify community of like-minded people and then suggest friends based on their interest match. Present social networking sites recommend friends to users based on their social connections or geographical location, etc. In this paper, we describe a state-of-the-art approach of recommending friends that is based on the idea of grouping users who review businesses similarly into a community and then recommending friends within their community who are similar with respect to certain user attributes. It may be worthwhile for users with similar reviewing patterns and behaviors to see each-other's activity. Our approach involves applying various community detection algorithms on a graph of users and further suggesting friends from same community who are similar to them.

KEYWORDS

Yelp, Interest based Friends Suggestion, Community Detection Algorithms, Modularity, Girvan-Newman

1 INTRODUCTION

Social Media has become a primary influencer in our daily lives. People's decision of where to visit or what services to avail are subject to other's opinions, especially of those that they can rely upon. The social network data has treasure trove of information on people's interests. Today, in many websites like Yelp, users can post their reviews about various businesses and services which consist of natural language text and a numeric star rating, usually out of 5. These online reviews serve as a 'word-of-mouth' and a criterion for users to choose between similar products. This crowd-sourcing way of obtaining users' satisfaction on a service has succeeded in providing different opinions about a certain service. Important information can be easily obscured unless users are willing to spend a great deal of time and effort on reading the reviews thoroughly.

Some Yelp users mentioned on *Yelp Talk* that their *'Friend's Activity'* Yelp. 2018. "How do I make more friends on Yelp to get more Friend's Activity to read?" Retrieved from [1] feed allow them to follow people on restaurants, local places, things to avoid, etc. and were curious to know how to gain more friends on Yelp to be able to follow their 'Friend's Activity'.

In this paper, our work is centered around being able to recommend friends who are similar to users based on their reviewing patterns and other user related attribute similarity. 'Friend's Activity' and recommendation can help when you know too little or nothing about the service.

Our proposed solution aims to create communities of users where each community represents a group of people who have same reviewing habits. We have implemented different community detection algorithms and compared how communities were formed. Once the communities are identified, we then, compute cosine similarity between each user and other community members based on user-specific attributes to recommend top users as friends.

*Produces the permission block, and copyright information

2 RELATED WORK

Social networks are one of the most popular ways to model interactions among people in a group or community. In this section, we discuss previous and existing work related to suggesting friends based on similar interest. Some of the current work to suggest friends in social networking sites involve suggesting users with respect to mutual connections rather than considering similarity aspect between two users. For example, Facebook's 'friend finder' feature suggests people whom we might already know but not yet connected through Facebook rather than suggest people whom you might find interesting enough that the new connection may lead to actual friendship in real life. [3]

To our knowledge, there is very little research carried out to suggest friends based on user's similar interest.

3 METHODOLOGY

3.1 Data

We collected our data from Yelp Dataset Challenge 2018 [4]. We have incorporated location as a means to restrict yelp users. As part of our exploratory analysis, we considered reviews given by users for businesses in a location , since users' location information is not available. After performing analysis on businesses, we found that the data for Arizona state was good enough to narrow down scope amongst other states. We found that 18,77,589 reviews out of 59,96,996 reviews overall, belong to reviews to businesses in *Arizona*, roughly 30% of overall reviews provided in the yelp dataset. There were 4,75,865 unique users who had provided reviews to businesses in Arizona. We further restricted our scope to just a city in Arizona. Please find below Table 1 that displays the number of reviews for each cities in Arizona.

We have run our proposed solution on different cities in Arizona. Subsequent sections in this paper, will talk about results from *Surprise* city in Arizona, in which we have 30,448 reviews for businesses. We have considered only reviews pertaining to open businesses in *Surprise* city of Arizona.

Table 1: Number of reviews in each city in Arizona

City	# of reviews
Phoenix	659878
Scottsdale	351158
Tempe	182610
Mesa	154278
Chandler	141090
Gilbert	114409
Glendale	88969
Peoria	50969
Surprise	30448
Goodyear	25169
Avondale	19578
Cave Creek	11554
Fountain Hills	6375
Litchfield Park	6195
Paradise Valley	4737

3.2 Graph Creation

We construct an unweighted, undirected graph G = ⟨V,E⟩ representing the topological structure of a network in which each vertex represents a user who has reviewed businesses and edge e = ⟨u,v⟩ ∈ E between user u and user v represents that they both have positively co-rated on businesses. Notice that the edge is created in case of positive co-relation only. We compute Pearson correlation based similarity score between users with the below formula:

$$\begin{aligned} \text{sim}(u,v) &= \frac{\hat{\mathbf{u}} \cdot \hat{\mathbf{v}}}{\|\hat{\mathbf{u}}\|\|\hat{\mathbf{v}}\|} \\ &= \frac{\sum_i \hat{r_{ui}} \hat{r_{vi}}}{\sqrt{\sum_i \hat{r_{ui}}^2}\sqrt{\sum_i \hat{r_{vi}}^2}} \end{aligned}$$

$\hat{\mathbf{r_{ui}}}$ is the normalized rating $r_{ui} - \mu_u$ Our graph with respect to Surprise city consists of 2335 nodes and 22920 edges.

3.3 Community Detection Algorithms

The graph created had one largest connected component / sub-graph with most of the users and rest of the connected components were less populated. We picked the largest connected component and applied various community detection algorithms on it namely, edge betweenness centrality approaches using Girvan Newman with different flavors such as removing least heaviest weighted edge, most heaviest weighted and

most central edge and optimal modularity based approaches namely, fast greedy and eigenvectors.

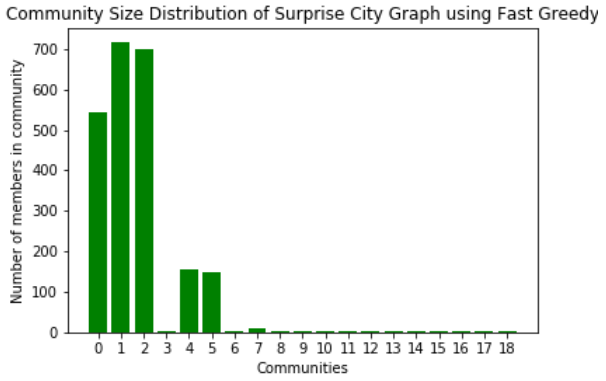


Figure 1: Community Size Distribution using Fast Greedy

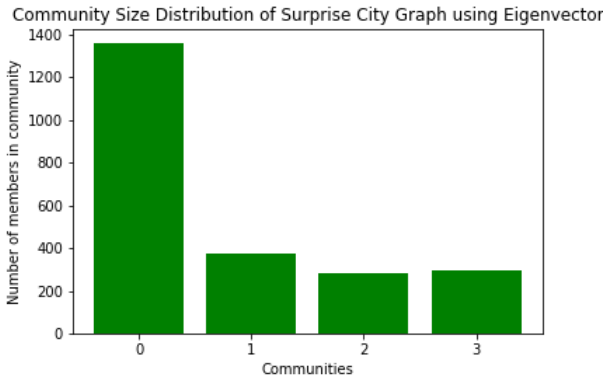


Figure 2: Community Size Distribution using Eigenvector

Clearly from the figure, eigenvectors give better community sizes. Typically, eigenvectors are known to be more accurate than fast greedy.[5] Although understandable and simple, Girvan Newman Algorithm has its own limitations. The algorithm is not very time efficient with networks containing large number of nodes and data. Communities in huge and complex networks are difficult to detect and therefore, Girvan Newman is not favorable for very larger number of data sets.[6] Therefore, we chose other methods of community detection which work on the idea of "Modularity". [7] [8]

3.4 Feature Engineering and Selection

Feature Engineering transforms the raw data into feature vectors that are suitable for modeling. We identified the indicator variables and engineered new features to fit the needs of the model. Friends count, number of years active and average length of reviews are some of the features we added.

We performed feature selection using the logistic regression model. This helps in ranking the important features that affect community creation. We considered around 21 variables as features and community to which the user belongs to as target variable.

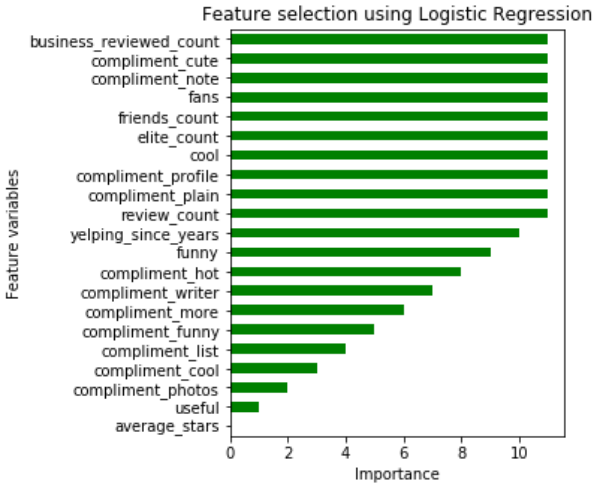


Figure 3: Features vs importance

3.5 Friends Suggestion

Top 10 features thus obtained were used to create feature vector for computing cosine similarity and top users corresponding to top values were suggested as friends.

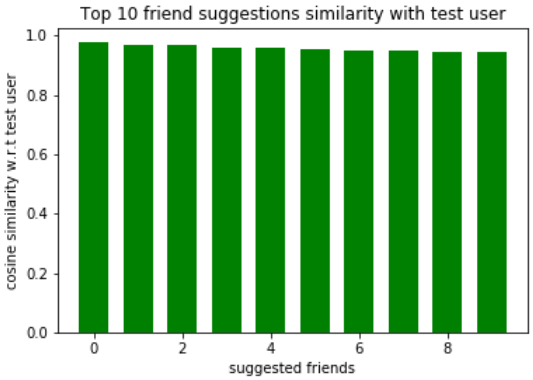
3.6 Handling Cold-Start

Elite users in Yelp have a high contribution and have large impact on the site with meaningful and high-quality reviews.[2] Due to this reason, we have chosen the Elite user count to tackle the cold start problem. For those users who are not part of any community, we pick one user with the highest elite count from each community and recommendations are made based on top cosine similarity values.

4 EXPERIMENTS AND RESULT

Initial baseline experimentation was carried out by considering Jaccard similarity between users who reviewed common businesses, but this approach ignores correlation of business ratings and hence Pearson Correlation was used for graph creation. GN algorithm require immense computation effort,thus not feasible on large graphs. We tried Cosine similarity between users for graph creation and finding co-rated business post GN failed miserably as there were hardly any users within community who reviewed same businesses.

Final model was constructed by taking above concerns into consideration. Pearson correlation based graph created for Surprise city users were fed to different modularity community creation algorithms. Cosine similarity is calculated on the users within community based on user feature vector and top most similar users are suggested to any given user. The graph below shows the cosine similarity of a user with his top 10 friends suggestion.



5 CONCLUSION AND FUTURE WORK

In this paper, we propose a friend recommendation problem and an approach that involves forming a network of users, applying community detection algorithms on it and further recommending top similar users in community as friends for each user in community. We tested our algorithm on various networks i.e, different cities in Arizona and show that our algorithm is able to suggest relevant friends and our results are convincing.

To improve the suggestions, the team brainstormed on novel ideas for future work which include feature engineering by applying Natural Language Processing techniques on text reviews and finding mutual connections of friends in the network. Additionally, we could provide intelligent recommendations for the cold start problem by handling different use cases appropriately. Finally, current recommendations can be improvised and validated by conversion rate of suggestions to actual connection and feedback from users in Yelp.

A APPENDIX

Team individually invested time on literature survey and performed exploratory data analysis. Monisha Shivarudrappa designed feature selection, cold start problem and K-clique, Bi-partitions and Label Propagation based community detection. Sanjana Srinivasa used Jaccard similarity to form graph and implemented Girvan Newman algorithm for community detection. Shravya Gorur Sheshadri worked on feature engineering and edge betweenness approaches for community detection. Vijetha Parampally Vijayakumar worked on graph creation using Pearson correlation and modularity based approaches such as Fast greedy, Walk Trap and Eigenvector for community detection.

Link to code on Github is provided in [9]

REFERENCES

[1] https://www.yelp.com/topic/manhattan-how-do-i-make-more-friends-on-yelp-to-get-more-friends-activity-to-read. *How do I make more friends on Yelp to get more Friend's Activity to read?*

[2] Kevin Crain, Kevin Heh, and Johhny Winston. 2016. An analysis of the elite users on yelp. com.

[3] http://be.amazd.com/link prediction/. 2014. *Link Prediction Algorithms.*

[4] https://www.yelp.com/dataset/challenge. 2018. *Yelp Dataset Challenge.*

[5] https://yoyoinwanderland.github.io/Community Detection/. 2017. *Community Detection in Python.*

[6] Mark EJ Newman. 2004. Detecting community structure in networks. *The European Physical Journal B* 38, 2 (2004), 321–330.

[7] Mark EJ Newman. 2006. Modularity and community structure in networks. *Proceedings of the national academy of sciences* 103, 23 (2006), 8577–8582.

[8] Karsten Steinhaeuser and Nitesh V Chawla. 2008. Community detection in a large real-world social network. In *Social computing, behavioral modeling, and prediction*. Springer, 168–175.

[9] Friend Suggestions to Yelp Users based on Similar Interest. 2018. GitHub repository. https://github.com/vijetha35/FriendsRecommendationForYelpUsers.