

ECS 289F - Network Theory and Applications

Communities for A Better Yelp

Leyuan Wang and Edmund Yan

June 12, 2014

Abstract

Yelp.com, has become a cornerstone of the internet we know today, yet it lacks the basic features of personalization. In this paper, we implement a new method for calculating business ratings that is based on community detection. We place users into communities of like-minded people, and see if they provide a better predictor of what a user will rate a business. It turns out though, that counter to our original thoughts, the consensus of those in your community are no better predictors than the consensus of the rest of the population and in fact, can be worse because of the small sample size.

1 Introduction

Yelp is a website that provides business listings and allows customers to write reviews for ratings. It has become an immensely popular website, climbing to being the 28th most popular website in the United States [1]. Part of its success can be attributed to their focus on building a great internal community of "Yelpers" through combining elements of social networking with friends and incentivizing users with badges and "Yelp Elite" status, an invite-only exclusive club. Yelpers rate businesses by giving them stars in the range of 1 to 5 and the business has an average rating that is shown to users.

We think we can do a better job at providing a rating to users than just an average. Yelp has accumulated a wealth of information on businesses, users, and activities between them, yet they use none of that information in providing better results for users. Our approach will be to use community detection methods to analyze the social graph of friends and also the bipartite network of users to businesses to give a more personalized result. Our hypothesis is that people in your community know what you like more

than the rest of the population. So if all of your friends are raving about a particular restaurant, we think this will make it more likely that you will enjoy the restaurant as well. Additionally, by analyzing the bipartite network, we can potentially unveil interesting pairings that would otherwise not be apparent. Users who frequently visit the same restaurants as a particular user, shows a starking similarity in their food preferences. We think that by categorizing users to communities, we can give more weight to the ratings from other users in the same community and thus provide a more accurate business rating.

2 Previous Work

For this project, we're going to be focusing on two main aspects of the Yelp data. First will be the social network of Yelp users and their friends. The second will be the bipartite network connecting users and businesses, where edges are reviews by that user to the business in question. After detecting communities in this two graphs, we are going to personalize the business stars according to which community the user is in. For the first part, we found different models for community detection. In [2], they introduced the louvain method which is based on modularity optimization. It outperforms other known community detection methods in terms of computation time which is a good fit for the large yelp social network. For the second part, we want to evaluate the user's credibility directly instead of the reviews he/she wrote, and thus the bipartite structure between users and businesses is similar to the hub and authority set-up in HITS[4]. As the current existing community detection methods in bipartite graph are quite slow for large data like yelp, we decide to transform the users-to-businesses bipartite graph to a users-to-users social graph by using Jacard similarity. And the label propagation method in [3] works for the second graph for it targets not necessarily on dense graphs and it can detect overlapping communities which happens for yelp users.

3 Data

Yelp has provided an academic dataset that contains a sample of their data from Phoenix, Arizona. This dataset has 15,185 total businesses, 335,022 reviews, 70,817 users, and a slew of other information that we don't track in this study. We construct two graphs from this data. The first is a social network of users and their friends, with the 70,817 nodes, 151,516 edges and an average degree of 4.279. As seen in the degree distribution of Figure 1, the graph is scale-free and follows a power law distribution.

The second is a bipartite graph of users and businesses, with an edge connecting between them signifying

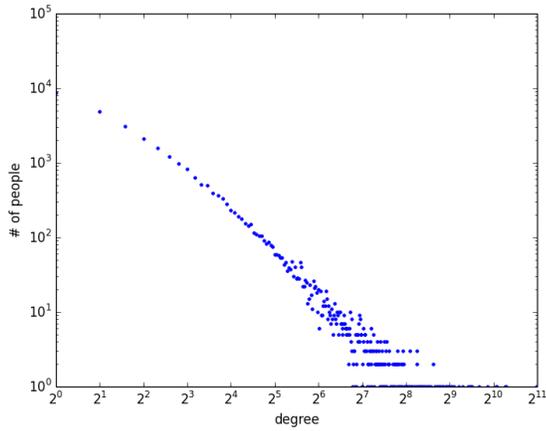


Figure 1: Degree distribution of social network

that user has rated the business. The weight on the edge is the 'star' attribute associated with that rating. As said above, there were 70,817 users, with an average degree of 4.560, and 15,579 businesses, with an average degree of 20.727. Figures 2/3 shows the distribution for both types of nodes.

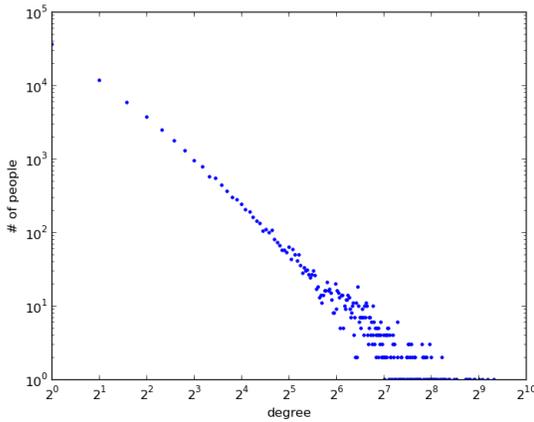


Figure 2: Degree Distribution of Users

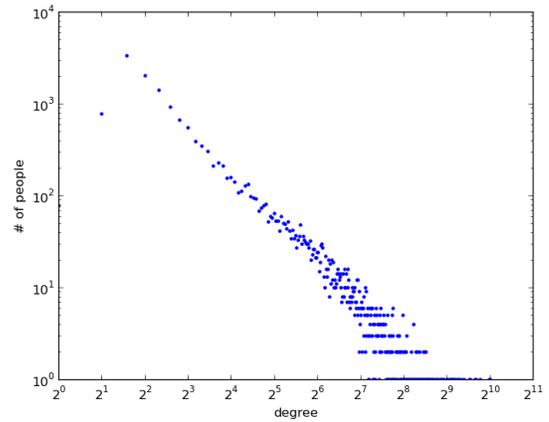


Figure 3: Degree Distribution of Businesses

We actually use a shrunken version of both graphs because of computational restrictions, which we will discuss in the next section.

4 Methods

4.1 Bipartite Graph

To do community detection on the bipartite graph, we first do a one-mode projection onto the Users. Simply, if user A and user B both rate a business X, we add an edge between A and B in the projected graph; the new graph only has user nodes. This algorithm leads to an immensely dense graph, with a projection of the original bipartite graph having 2,945,687 edges with only 70,817 nodes. Community detection algorithms on this graph were both time consuming and produced bad results. We thus add two methods to shrink this projection into a more usable graph. The first is a "shrink" operation that removed nodes (both users and businesses) in the bipartite graph that have a degree less than certain threshold. In our results below, we assigned the minimum degree to 10 for both users and businesses. The reasoning is that users or business without enough edges won't have strong bonds with others in their community for this method to be effective. Secondly, when projecting, we compute the Jaccard index between the two users to see if the bond is strong enough to create an edge. The Jaccard index is a statistical property of two users by computing the ratio of number of intersecting businesses to total businesses that these two users have in common. We again apply a minimum threshold of 0.10, or 10% of shared businesses. After both of these methods, we are left with a graph of 5,571 nodes and 5,034 edges.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

4.2 Community Detection

For community detection, we test out two different methods: label propagation[3] and louvain[2]. In the Figures 4 and 5, we see a histogram of community size. Our original hypothesis was that we would get good results with more medium sized networks as this would group users to a small community of other users they are closely related to, which is what label propagation gave us. The louvain method ended up creating a handful of extremely large communities. Unfortunately, the label propagation method did not work for the social graph, so we use the louvain method for the social graph throughout the study. The size distribution is shown in Figure 6.

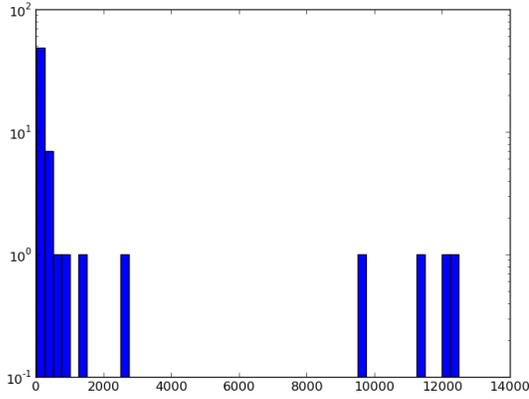


Figure 4: Community Size Distribution of projected graph using Louvain Method

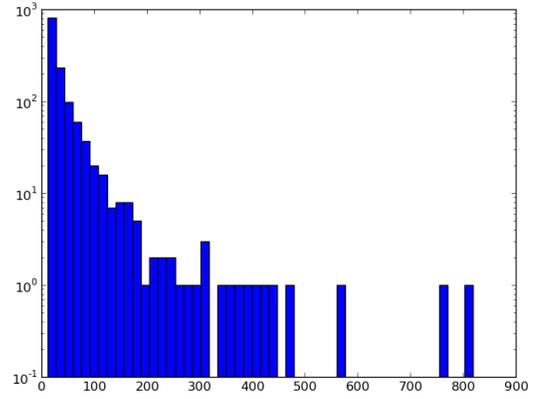


Figure 5: Community Size Distribution of projected graph using label propagation

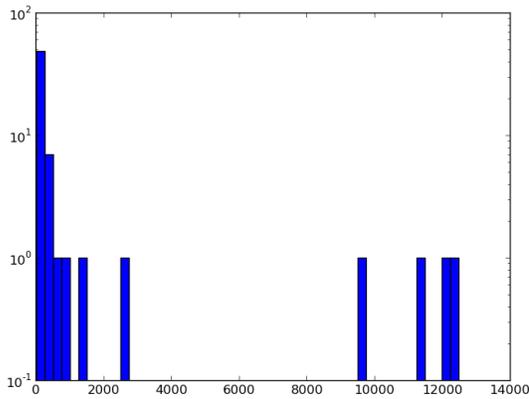


Figure 6: Community Size Distribution of social graph using Louvain Method

4.3 User Credibility

In our implementation, we use a HITS-based scoring algorithm from [5] to obtain the user credibility and business new star. Instead of using Java, we implement it in python and it turned out to be very fast by operating it in an EM manner and with the stopping criteria 99% businesses' new star changing less than 1%. In the method, the user and business are acting as the hub and authority roles in HITS algorithm. Firstly, we calculate the user credibility from the business every user has reviewed before. If $Biz = \{b, \text{where user has reviewed}\}$, is the business set every user has reviewed, $star_b[i]$ is the score each user i gives to business b , and S_b is the original Yelp star for business b . The credibility of user i is calculated as follows:

$$C[i] = \frac{1}{1 + \sum_{b \in Biz} (star_b[i] - S_b)}$$

The +1 in the denominator is used to smooth the user credibility in case of $star_b[i] == S_b$. We can thus get a user credibility for each user.

Next step, we calculate the business new star according to the updated user credibility. If U_b is the set of users who have reviewed the business b , we can get the weighted score by the following equation:

$$S_b = \frac{\sum_{u_i \in U_b} \times star_b[i]}{\sum_{u_i}}$$

We also need to normalize and scale the user credibility to a maximum of 5 by the following equation:

$$u_i = \frac{u_i}{\max_{j \in v} u_j} \times 5$$

5 Results

To calculate a new business rating, we have three sets of people to pull information from: those inside our bipartite network community, those outside our bipartite network community, and those inside our social network community. The weights of each group of users can be weighted differently. We decided to test all different permutations of weights to see which one optimized the results the most. For a certain weight permutation, we compute a new business score using Eq.2. For our results, we calculate this new business score for every user and every business and calculate the difference between what the user actually rated the business and what our new score is. We imposed some size restrictions, such as forcing users to be in communities of at least 10 users or more. The results for both community detection methods are shown in

Figures 7 and 8.

$$\begin{aligned}
 NewScore(U, B) = & peer_rating(U, B) * peer_weight + outsider_rating(U, B) * outsider_weight + \\
 & social_rating(U, B) * social_weight
 \end{aligned}
 \tag{2}$$

From the data, it's clear that our method is consistently worse than both Yelp and the HITS method. The X axis is an index of a certain permutation. A weight permutation of (0,0,1.0) would have an index of 0, while a permutation of (.1,0,.9) would have the index 11. The Y axis is the mean difference between what a user rated a business and what we calculate that business score to be. The red 'Yelp' line is the difference to Yelp's standard rounded half-star rating; the yellow 'HITS' line is the difference to the HITS business score in Section 4.3; the blue line is our diff using community detection.

Our best results occurred when peer_weight=0.2, outsider_weight=0.8, and social_weight=0.0 for both detection methods, which is permutation index 21.

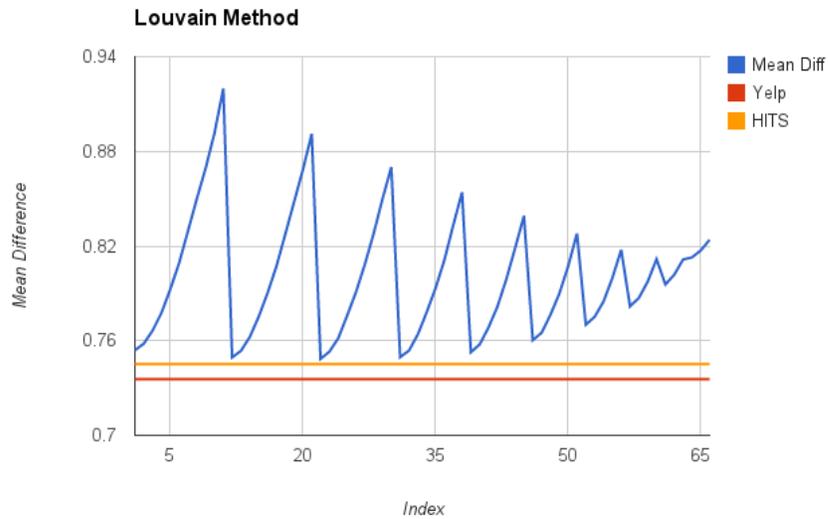


Figure 7: Mean difference between Louvain method and actual ratings

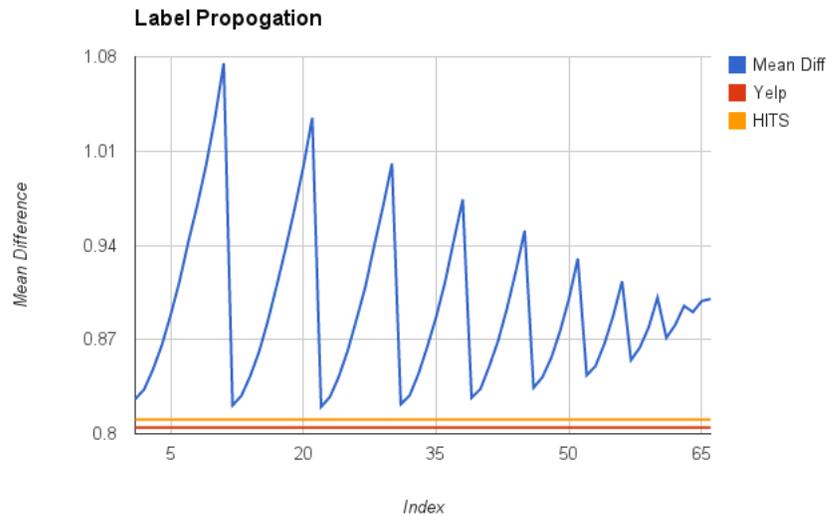


Figure 8: Mean difference between label propagation method and actual ratings (lower is better)

6 Conclusion

It's unfortunate that our results were not very good, and there are many things that could be improved on. For one, because of our reliance on networkx and python, we were quite computationally limited. This meant we were working with an incomplete set of data. If we had more computation power or wrote our code more efficiently, it would've been amazing to see how all of our tunable parameters changes the final mean difference. Things like our Jaccard threshold, min community size, min number of friends, min number of reviews, etc. From our proposal, we really wanted to see if there was some phase transition when our method would suddenly become more effective once a user hit a sweet spot of friends/reviews - unfortunately we had neither the time nor coding skills to implement this efficiently.

We would've loved to do more community detection algorithms and see what difference that would've made. We did not fully understand the clustering algorithms we were doing, and some of the attributes like weights and overlapping communities weren't utilized to its fullest.

While there are many improvements on the technical side, if we take our results at face value, we can still come to an interesting conclusion. From the data, it would seem that your friends or those people who review similar restaurants as you are no better predictors of your rating than the general population. In fact, as we start including more and more weight to peer_weight than outsider_weight (which can be seen as

the minimum peaks in the difference graphs), we slowly increase. In other words, the general consensus on a restaurant tends to be a more accurate predictor for any user simply because of larger sample size.

Code for this project can be viewed at <https://github.com/edmundyan/yelp>

References

- [1] Alexa. *yelp.com Site Overview*. May 2014. URL: <http://www.alexa.com/siteinfo/yelp.com>.
- [2] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008), P10008.
- [3] Steve Gregory. “Finding overlapping communities in networks by label propagation”. In: *New Journal of Physics* 12.10 (2010), p. 103018.
- [4] Mark EJ Newman and Michelle Girvan. “Finding and evaluating community structure in networks”. In: *Physical review E* 69.2 (2004), p. 026113.
- [5] Hao Zhao and Haoran Li. *Community-Based Yelp Personalization*. 2013. URL: <http://www.stanford.edu/class/cs224w/projects2013/cs224w-032-final.pdf>.