Community Detection and Network Analysis on Random Acts of Pizza

Malina Jiang, Diana Le, Spencer Yee
malinaj, dianale, spencery

Abstract

Random Acts of Pizza is a subreddit on Reddit that encourages users to give random redditors a pizza. This community has been researched under the sphere of NLP and how to form a request that will be successfully result in a pizza. In this paper, we explore the properties of the Reddit community to see if a requester's network structure and placement can predict whether a requester will receive pizza. We use community detection algorithms, general network characteristics, and stochastic gradient descent to create a predictor that determines the success of a request. We show surprisingly good experimental results using these network properties.

1. Introduction

According to existing research on altruism in evolutionary psychology (Curry, Roberts, & Dunbar 2012) and social exchange theory (Maner et. al 2012), acts of altruism among humans are ultimately self-serving and become less common the less genetically close a giver and a recipient are. However, this has not prevented sites such as GoFundMe or KickStarter, whose sole purpose is to facilitate altruism, from becoming large Internet hubs and communities; the biggest success stories of these sites have raised millions in funding and reached millions more in viewership. These stories are evidence of other factors at play when it comes to altruism. J. Skgeby postulates that online gift-giving serves as a way to seal bonds between members of an online community and contributes to an economy of regard, where fulfilling earnest requests is exchanged with expressions of gratitude (Skgeby 2010). We look to examine a particular case of online gift-giving in the form of Random Acts of Pizza (RAOP), a subreddit on Reddit dedicated to fulfilling users requests for pizza. Using network analysis, we hope to uncover what factors may increase the likelihood of fulfilling a pizza request and possibly get to the sources of altruism in a part of the Internet community that proudly proclaims Restoring Faith in Humanity, One Slice at a Time.

2. Related Work

Under the lens of natural language processing, Althoff, Danescu-Niculescu-Mizil, and Jurafsky (ADJ) study the RAOP community on the website Reddit.com to examine the factors that contribute to the successful completion of a favor made by a member to the community. The data used in this study is derived from the entire history of the RAOP from its inception in December 2010 to September 2013. Using this data, they attempt to determine if factors such as gratitude, reciprocity, urgency, status, sentiment, and similarity can be used to predict the success of a favor asked of a community.

This study sought to measure user similarity and status. Users were defined by their interests in terms of subreddits, and similarity was defined by two metrics - the intersection size between the interest sets of the giver and receiver, and the Jaccard similarity of the two. Status was defined by the karma accumulated by the user and by the number of posts made by the user. This paper also introduced a number of text sentiment and linguistic patterns as factors in determining the success of the request, such as the linguistic indicators of politeness/gratitude, reciprocity, need, and mood.

The model generated in the study was a significant improvement upon the random baseline, and the results showed that gratitude, reciprocity, urgency, and status were significant factors in the success of a request, but that user similarity was in fact not a significant factor. There were a number of indicators of user-similarity that were not considered in this study that could have affected the conclusions of this paper, including similarity in geographical location, life situations, or language use. Another improvement in the representation of user-similarity could be a slight adjustment to the similarity metric. The current study models similarity as a set of user interests represented by Subreddits that were frequented (posted in) at least once, but these attributes were not weighted. In the future, it might be helpful to consider user interests as a set of weighted attributes, which might refine and add more dimension the user-similarity factor.

Althoff, Salehi, and Nguyen also studied factors that contributed to the success of a request for pizza on the RAOP subreddit community. It used many similar features, with a few notable differences. User similarity between two users was defined as the number of subreddits contributed to by both users plus the number of comments from the users in these shared subreddits.

Compared to the previous paper, Althoff et al. studied a slightly broader set of features, acknowledging certain complexities of reddit posts in its analysis. It approached user similarity differently and found that user similarity was significant, though not a large factor. This is a key difference from the other paper, which is of great interest to us due to it being a network property as opposed to a linguistic one. While this paper accounted for weighting shared subreddits, this paper did not account for what percentage of the users subreddits were shared.

While these papers made progress in determining the success of a request, we aim to take another look at how user similarity can be quantified by instead examining the community structure surrounding requesters and givers. By representing users and givers as nodes in a graph with weighted edges, we hope to use network properties such as betweenness centrality, shortest paths, degree, and community detection to be able to look at user similarity from a different perspective.

3. Dataset

Our dataset contains the entire history of RAOP from December 8, 2010 to November 16, 2016 (32036 posts total). For every user involved in RAOP, we also crawled the history of posts and comments across all subreddits to obtain a snapshot of their reddit activity prior to posting in RAOP. We used this information to perform basic community analysis on RAOP users, though we had difficulty identifying which requests were successful and who gave pizzas to those requesters. Therefore, we restricted our predictor to the requesters and givers in the ADJ dataset, which contains posts up to September 29, 2013, and contains 5728 requests with an average success rate of 24.6%. We were able to fill in some missing giver names for successful requests by examining the differences between the edited and unedited post texts, and checking for the presence of gratitude in the texts, resulting in an increase in giver data by over 16%.

We formed a graph of RAOP subredditors using the user activity that we scraped, where each node represents a subredditor (including both givers and requesters) and each undirected edge between two nodes is weighted by the similarity of the users. We calculated user similarity by representing each user's reddit activity as a vector of post and comment counts for each individual subreddit and taking the cosine similarity between users' vectors. In order to sparsify the data for our community detection algorithms, we kept only the 5 edges with the highest weight for each node.

4. Algorithms and Methods

Using three algorithms for community detection, we hoped to gain insight on the types of communities that requesters and givers were a part of and to also see which community detection algorithm had the highest accuracy in predicting the success of a pizza request based principally on the clusters that formed from the algorithms. We also incorporate various general network features based on the outdegree, betweenness centrality, and shortest paths calculations between nodes for the predictor.

4.1. General Analysis

Because we want to capture a requester/giver's place in the network and how their location relative to other nodes affect their ability to fulfill a request, we use outdegree, betweenness centrality, and shortest paths calculations to create some intuition on a node's "reach" within the network. For each of the three properties, we will be comparing the values between requesters and givers to see if it is possible to further classify a node as a requester or a giver.

4.2. Multi-way Spectral Clustering

In general, community detection can be thought of as attempting to find groups where the members within a group are similar to one another while being dissimilar to members of other groups. Spectral clustering is one way of finding these groups and refers to the technique of partitioning the rows of a matrix–often a Laplacian of the graph–by the components in the top k singular vectors of the matrix. In general, the spectral algorithm uses the rank-k subspace defined by the top k right singular vectors of A as an approximation of A and then projects all of the rows of A onto the rank-k subspace. These singular vectors then each define a cluster, and we can use various other methods of clustering such as k-means to map each of the points in A to a cluster (R.Kannan). Differences in spectral clustering algorithms generally lie in how the Laplacian is defined and in what measure of clustering is being optimized. We explore two such methods, ratio cut minimization and modularity maximization, in this paper.

4.2.1 Ratio Cut Minimization

Restating the community detection problem in a different way, we want to find a partitioning of the graph $S_1, S_2...S_k$ such that we minimize the sum of the weight of the edges that have one endpoint in S_i and another endpoint in S_j where $i \neq j$. This is formally known as the $cut(S_i)$. While minimizing the $cut(S_i)$, however, we also want to ensure that each cluster S_i is well defined without being too small. We can then define a spectral clustering algorithm that attempts to minimize the Ratio cut:

$$RatioCut(S_1, ..., S_k) = \sum_{i=1}^{k} \frac{cut(S_i)}{|S_i|}$$

In Hagen and Kahng's paper on ratio cut optimization, the trace of $X^T LX$, where X is the assignment matrix of each of the rows of A to a cluster S_i and L is the Laplacian of the graph, is proportional to the RatioCut$(S_1, ..., S_k)$. The ratio cut optimization problem is then equivalent to

$$\min_X tr(X^T LX)$$

$$\text{s.t. } X^T X = I, X = \begin{cases} \frac{1}{\sqrt{|S_r|}}, & \text{node } i \in S_r. \\ 0, & \text{otherwise.} \end{cases}$$

Because this is an NP-complete problem, we relax the second constraint and simply solve the problem where the only constraint is that $X^T X = I$. We can then find the k smallest eigenvalues of L and use k-means to cluster the rest of the points to these first k vectors.

4.2.2 Modularity Maximization

The motivation behind modularity maximization involves evaluating the quality of a particular partitioning. Modularity (Q) quantifies the quality by comparing the fraction of edges within the community with such a fraction when random connections between the nodes are made. A community should then have more links among its members than a random cluster of nodes. A smaller Q value, closer to 0, would mean that the fraction of edges within communities is not better than a random ordering while a Q value closer to 1 would mean that the community structure has the highest possible strength. The modularity is defined as (M. E. J. Newman and M. Girvan):

$$Q = \frac{1}{2|E|} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2|E|}] \delta_{e_i, e_j}$$

. We then want to maximize the modularity to find an optimal partitioning of the graph. Chen et al. shows that maximizing the modularity is similar to maximizing the trace of $X^T L_Q X$, where X again refers to the assignment matrix and L_Q refers to the "Q-Laplacian." This maximization problem also contains the constraint of finding the assignment matrix X that maximizes Q, which is again an NP-complete problem. We again relax this constraint, assuming that the elements of X take on real values, and we can then take the k largest eigenvectors of L_Q and cluster around these eigenvectors to form the communities.

4.3. Graph Coarsening

Graph coarsening is an agglomerative clustering algorithm that is composed of three phases: coarsening, partitioning, and uncoarsening. More detailed descriptions of this algorithm can be found in (G. Karypis and V. Kumar).

4.3.1 Coarsening Phase

The graph G_0 is coarsened by constructing a sequence of smaller graphs $G_1, G_2, ...G_n$ through a matching algorithm, in this case, heavy-edge matching. Through each subsequent round of matchings, the number of nodes in the graph decreases while the weights of the edges increases. Heavy-edge matching is executed by a random traversal through the nodes, where the matching for each node is defined to be the node that shared the heaviest edge weight with the initial node. Nodes in each matching are clustered together and their edge weights combined.

4.3.2 Partitioning Phase

After coarsening the graph, the resulting coarsened graph is partitioned using multi-way spectral clustering and k-means.

4.3.3 Uncoarsening Phase

The uncoarsening phase unravels the clustered coarsened graph into its original form. Each original node in the aggregated nodes of the clustered graph become members of the same cluster when the graph is restored to its original uncoarsened state. As an example, if a coarsened graph contains the clusters c_1, c_2, and c_3, and the cluster c_1 contains the aggregated nodes a_1 and a_2, then nodes $n_1, n_2, ..., n_i$ in a_1 will also be members of c_1.

4.4. Binary Classifier Using Stochastic Gradient

Using the results from the community detection algorithms and general network features, we will create a binary classifier that will take in as input a requester on RAOP and determine whether or not the request will be successful. We will be training the classifier using stochastic gradient descent, which iteratively minimizes an objective function to produce a weights vector. This weights vector will be used to classify the test dataset. We will be testing the binary classifier on each of the three different clustering algorithms to determine which of the algorithms performs best on the RAOP data.

5. Results and Findings

5.1. Network Analysis

Network analysis of the RAOP subreddit separates the community into three sets of users defined by their role within the community: requester (successful or unsuccessful), or giver.

From user activity, we were able to examine the commonalities between different members of the community. Table 1 shows some examples of top subreddits for requesters and givers normalized by their subscriber count, showing which subreddits they participate in more than the average subredditor. We observed that both requesters and givers are fairly likely to be members of other "pay it forward"-type subreddits, and that givers are also likely to be contributing members of other subreddits with helpful and liberal inclinations. These findings approximately match our expectations for RAOP community behavior.

Requesters	Givers
promos	RandomActsOfCookies
Loans	secretsanta
RandomActsofCookies	Loans
Assistance	techsupport

Table 1. Top subreddits per user type

5.1.1 Out Degree and Edge Weights

The average out degrees calculated from the network of RAOP users show that givers have the highest out degree, followed by unsuccessful requesters and successful requesters.

User Type	Average Out Degree
Successful	5.4086
Unsuccessful	6.3683
Giver	6.4334

Table 2. Average out degrees by user type

The average edge weights from the RAOP network show that unsuccessful requesters have the highest average edge weight, followed by givers and successful requesters.

User Type	Average Edge Weight
Successful	0.7037
Unsuccessful	0.7216
Giver	0.7167

Table 3. Average edge weights by user type

Calculating the average edge weights for different pairs of users, we see that giver to giver edges have the highest average weights, meaning givers are highly similar to each other. The second highest average is the average of the weights between successful requesters and givers, which is consistent with the idea that givers are more likely to give to users they are most similar to. In contrast, the average weights between unsuccessful requesters and givers are significantly lower.

User 1	User 2	Average Edge Weight
Successful	Successful	0.6534
Successful	Unsuccessful	0.7071
Successful	Giver	0.7277
Unsuccessful	Unsuccessful	0.7229
Unsuccessful	Giver	0.7067
Giver	Giver	0.7643

Table 4. Average edge weights by type of user pairs

Using the values from the tables above, we ran 2 sample t-tests to determine if the network features in this section could be valid predictor features. Comparing the average out degrees for successful and unsuccessful features, we find a p value of 0.00032, which is well below the significant threshold of 0.05. So, the null hypothesis is incorrect and the average out degrees are significantly different. Similarly, the p value from the t-test of the average edge weights for successful and unsuccessful requesters also yields a significant value, 1.70×10^{-5}, meaning the average edge weights are also significantly different between the two groups of users. Since both results are significant, we use both the average out degree and average edge weight in our predictor.

5.1.2 Betweenness Centrality

The average betweenness centrality show that givers have the highest average betweenness centrality, followed by unsuccessful requesters and successful requesters.

User Type	Average Betweenness Centrality
Successful	8308.8582
Unsuccessful	8499.7119
Giver	8931.1270

Table 5. Average betweenness centrality by user type

Running 2 sample t-tests on the betweennes centrality values for successful versus unsuccessful requesters returns a p value of 0.75, meaning the averages for these two sets of users are not significantly different from each other.

5.1.3 Shortest Paths

The average length of the shortest paths is greatest between unsuccessful requesters and givers, followed by giver-giver shortest paths and successful requester-giver shortest paths.

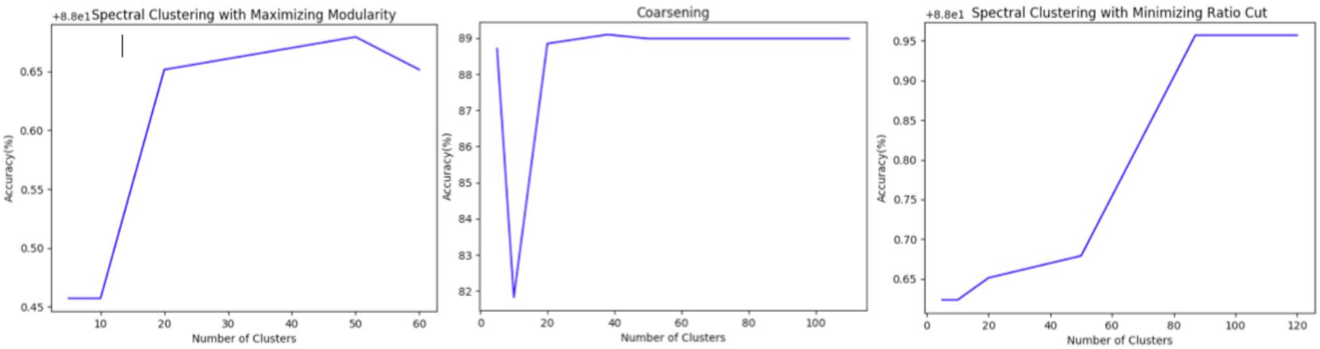

Figure 1. These graphs show the accuracy achieved by the predictor for a number of clusters for each of the three community detection algorithms. For each of the algorithms, six different cluster sizes were tested: 5, 10, 20, and 50 of the clusters with the most nodes, the maximal number of clusters that contained more than 1 node, and the optimal number of clusters for that algorithm, described in section 5.2.

User Type	Average Shortest Path Length
Successful	2.6995
Unsuccessful	2.7358
Giver	2.7234

Table 6. Average shortest path length by user type

A 2-sample t-test on the average length of shortest paths between successful requesters and givers as compared to unsuccessful requesters and givers returns a p value of 2.79×10^{-87}, which is far below the threshold $p = 0.05$. Then the average values are significantly different between the two groups, and successful requesters have significantly shorter paths to givers in the RAOP network, which is in line with our expectations that successful requesters are closer in proximity to a greater number of givers.

5.2. Community Detection

As described earlier, we used three different community detection algorithms to cluster the users, where the edges between a pair of users was weighted by the user similarity between the two. We then iterated these community detection algorithms on 42 distint numbers of clusters, ranging from 2 clusters to 1000 clusters. Since our statistic analysis of the clusterings showed that the features we were examining were significant across all cluster numbers and community detection algorithm types, we chose clusterings that minimized the p-value of the t-tests.

Algorithm	Successful Pairs	Expected Pairs
Min Ratio Cut	46	38.0071
Max Modularity	171	162.1272
Graph Coarsening	132	127.2704

Table 7. Actual and expected number of successful requester-giver pairs in same cluster

The table above shows that successful requesters and their corresponding givers appeared in the same cluster more frequently than would be expected if the distribution of these requesters and givers was simply random, suggesting that successful requesters and their givers had higher user similarity than a random pair of users. This difference is especially significant in the minimizing ratio cut clustering, where there are 21% more pairs than expected in a random network.

A 2-sample t-test on the average number of givers in requester cluster for successful requesters versus unsuccessful clusters returns a p-value of 3.53×10^{-71} for minimizing ratio cut, 1.03×10^{-75} for maximizing modularity, and 5.96×10^{-116} for graph coarsening, meaning there is a significant difference in the number of givers in clusters with successful requesters versus unsuccessful clusters and successful requesters tend to appear in clusters with more givers.

5.3. Prediction on the Success of Pizza Requests

Network analysis of the RAOP subreddit revealed a number of network features that differed significantly between successful and unsuccessful requesters, which we translated to features in our predictor. Our predictor aims to predict whether or not a request will be successful based on network features associated with the user making the request. The initial predictor was a random baseline that returned a score of successful or not successful with 50% probability. In subsequent versions of our predictor, we iterated on different combinations of features including the out degree, length of shortest paths, betweenness, and clustering assignments from each of the three clustering algorithms.

We were able to convert the clustering assignments to features in the predictor by taking the top n clusters (in terms of number of nodes assigned to the cluster) from the clustering assignment, and generating a binary vector for each node n_i, where a 0 meant the node was not assigned to that cluster, and a 1 meant n_i was assigned to that cluster. We also maximized predictor accuracy over the n number of top clusters (Figure 1). Accuracy was maximized by taking the top 87 clusters for minimizing ratio cut, top 50 for maximizing modularity, and top 34 for graph coarsening.

Feature	Predictor Accuracy
Random Baseline	0.4711
Out Degree	0.5763
+ Shortest Path	0.8860
+ Min Ratio Cut	0.8896
+ Max Modularity	0.8868
+ Graph Coarsening	0.8910

Table 8. Predictor accuracy with successive features

We found that betweenness centrality as a feature negatively affected our predictor accuracy (as expected, since our network analysis shows that betweenness was not significantly different between successful and unsuccessful requesters), so we removed the feature from our final predictor. Our three final predictors included out degree, shortest path, and one of the clustering assignments from the three algorithms. The most successful predictor, which included out degree, shortest path, and graph coarsening clustering assignments, had an accuracy of 89.1%, compared to our baseline accuracy of 47.11%.

6. Conclusion

A number of interesting statistics were shown as a result of our analysis on the RAOP community. Successful requesters tended to have shorter paths to other nodes than unsuccessful requesters or givers, which may indicate that successful requesters tend to cluster around many important members of the community or may simply know people that are more likely to fulfill a request. For each of the community detection algorithms, the predictor also showed that the maximal number of clusters with more than one member performed the best; this seems to indicate that more information about the clustering and more communities are needed to create a more accurate predictor, but only to a certain degree. Having outlier nodes did not help increase predictor accuracy; it, in fact, caused a decrease in accuracy in certain cases.

While the predictor's accuracy is quite high, it should be noted that our training data contains a high proportion of unsuccessful requests which may have skewed our predictor to mostly predict that a request was unsuccessful. However, it seems that within the RAOP community, most requests are not met with a giving of pizza, as our pre-processing in addition to the Jurafsky set also showed a skewing towards more unsuccessful requests.

Overall, our results show that using general network properties and community detection may help in increasing the accuracy of a binary predictor that is classifying whether or not a request will be met with a pizza.

7. Future Work

As mentioned in our literary analysis, previous work on the Random Acts of Pizza subreddit interactions focused primarily on linguistic features in the request texts themselves, while our work focuses solely on network features. In the future, we could combine our network features with linguistic features from previous studies to refine our predictor and observe changes to our predictor accuracy. Our current analysis was also confined to just the requests from the ADJ study, so we could expand our dataset by including requests (successful and unsuccessful) outside the time frame of the current RAOP data, as well as expend more effort identifying the givers of successful requests. Another area for future work would be to change the definition of predictor accuracy to focus more on identifying successful requests, rather than identifying whether or not a success was successful, since most of the RAOP data contains unsuccessful requests and the greater number of negative labels may slightly bias the predictor.

8. References

G. Karypis and V. Kumar, Analysis of multilevel graph partitioning, Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Supercomputing '95, 1995.

J. K. Maner, C. L. Luce, S. L. Neuberg, R. B. Cialdini, S. Brown, and B. J. Sagarin, The Effects of Perspective Taking on Motivations for Helping: Still No Evidence for Altruism, Personality and Social Psychology Bulletin, vol. 28, no. 11, pp. 16011610, Jan. 2002.

J. Skgeby, Gift-giving as a conceptual framework: framing social behavior in online networks, J Inf Technol Journal of Information Technology, vol. 25, no. 2, pp. 170177, Apr. 2010.

L. Hagen, A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. IEEE Transactions on Computer-

Aided Design of Integrated Circuits and Systems, 1992.

M. Chen, K. Kuzmin, and B. K. Szymanski, Community Detection via Maximization of Modularity and Its Variants, IEEE Transactions on Computational Social Systems, vol. 1, no. 1, pp. 4665, 2014.

M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E, vol. 69, p. 026113, Feb 2004.

O. Curry, S. G. B. Roberts, and R. I. M. Dunbar, Altruism in social networks: Evidence for a kinship premium, British Journal of Psychology, vol. 104, no. 2, pp. 283295, 2012.

R. Kannan, S. Vempala, A. Vetta. On clusterings: Good, bad and spectral. Journal of the ACM, 51(3):497-515, 2004.

T. Althoff, C. Danescu-Niculescu-Mizil, and D. Jurafsky. 2014. How to Ask for a Favor: A Case Study on the Success of Altruistic Requests. In Proceedings of the 8th International AAAI Conference on Weblogs and Social Media (ICWSM 2014).