CS 224W Information Networks
Angela Sy, Larry Liu, Blanca Villanueva
Oct. 16, 2014
Project Proposal

Finding Your Food Soulmate

I. Abstract

Individuals in social networks are often unaware of people who share similar tastes. From our own personal experience, we believe that people often bond over food, and so we want to detect communities of common food preferences and food-related activity. To do this, we aim to conduct an analysis of the Yelp network. Our approach takes into consideration the Yelp user base (consisting of users and businesses), as well as user activity: creation of user reviews, check-in's, and follows. The following proposal evaluates three algorithms relating to community discovery and cluster assignment. Advantages and disadvantages discussed range from the ability for an algorithm to capture belonging in multiple communities, to the computational viability of certain approaches for certain datasets.

Based on the evaluation of the three algorithms, we propose our own approach to finding food communities within the Yelp user base.

II. Research Papers

a) Summary

Fast algorithm for detecting community structure in networks
M.E.J. Newman

Newman proposes an algorithm that is faster than Girvan and Newman's algorithm (the GN algorithm), and operates in $O((m+n)n)$ time, where m is the number of edges, and n is the number of nodes. In a sparse graph, this is effectively $O(n^2)$ time. Their algorithm does not sacrifice significant gains in the qualitative accuracy of the communities it identifies.

Their algorithm is based on the idea of modularity. For any given division of a network into communities, they propose a quality index to evaluate how well that particular division actually identifies groups based on the density of in-group edges and between-group edges.

They also propose a greedy algorithm to actually find a near-optimal such division that starts by treating the graph as n communities, and joining communities until all nodes are connected. After each connection, they evaluate the quality index, and can thus identify the division that had the highest quality index throughout the process. This process of finding the optimal division is where Newman makes computational gains over the GN algorithm.

On Spectral Clustering: Analysis and an algorithm
Andrew Y. Ng, Michael I. Jordan, Yair Weiss

Spectral clustering methods have been used on a variety of data sets with relatively high success rates. However, the paper argues that these methods have issues. Firstly, there are many different underlying implementations of the spectral clustering algorithm where implementations are affected by the different eigenvectors which are used. Secondly, no definitive proof can be found that this clustering works accurately. Ng et. all propose an alternative

spectral clustering method which draws from matrix perturbation theory, analyze the given algorithm, and give experimental results where the algorithm was found to work well.

Spectral clustering is the technique of partitioning the rows of a matrix according to their components in the top few singular vectors of the matrix. These top components are eigenvectors of the matrices derived from the data. When applied to graphs, nodes in a graph represent the data which will be partitioned into clusters. The adjacency matrix of the graph can then be used to extract the eigenvectors, which allows us to highlight the key components of each group of node. Hence, we are able to obtain the same data represented in low-dimensional spaces which can then be easily clustered.

The algorithm that the authors suggest is to use an affinity matrix with row sums equal to one. It creates k singular vectors (the chosen eigenvectors of the matrix), forming a new matrix. This matrix is normalized and then, using each row as a data point, formed into clusters using K-means. These results are then used to group the original data points into clusters.

This algorithm was applied to seven different experimental cases, each of which gave good clustering results. The authors argue that although other methods could create similarly good results, none of them are as simple as this algorithm because of limitations (such as convex regions for k-means) or a lack of robustness. Hence, the authors argue that their spectral clustering algorithm performs both efficiently and accurately.

Clustering by fast search and find of density peaks
Alex Rodriguez and Alessandro Laio

Rodriguez and Laio propose a density-based clustering algorithm that overcomes the shortcomings of previous strategies: insufficient ability to locate non-spherical clusters (K-meas, K-medoids), reliance on data defined by a set of coordinates, and computational costliness (density-based spatial clustering of applications with noise, or DBSCAN). The approach relies on characterising distances between data points, and finding local maxima in a density of data points in a two-dimensional space.

The algorithm requires measuring the distance between all pairs of data points. Cluster centers are identified as local maxima in densities of data points. The approach assumes that cluster centers have neighbors that have a lower density and are at a relatively large distance away from any other nodes with a higher local density.

Two quantities are computed for each node. Local density (rho) and distance from points of higher density (delta). For a cluster center, delta is set to a large distance, such that delta is much larger than the typical nearest neighbor distance for local or global maxima. Thus, cluster centers can be found by searching for anomalously large delta values.

The authors argue for the efficacy of their approach in a variety of different cases (some necessitating slight modifications to the original approach).

b) Critique

Fast algorithm for detecting community structure in networks

Newman's algorithm seems to effectively categorize nodes into correct community structures when using both artificial networks for testing as well as real-world networks. It provides an algorithm that is less computationally expensive over algorithms proposed in the past, including Newman's own old algorithm that he made with Girvan. The change in quality index across the entire graph can be calculated in constant time, which means that the constraint is simply the number of configurations tried. The final result is that the algorithm finds a good community structure in $O((m+n)n)$ time.

Newman's algorithm is limited in that it can only work for unweighted graphs. It seems possible that a workaround can be produced by using a multigraph to account for weights, but even then the graph would require discrete, positive integer weights. As such, some of the nuance of real-world social networks could not be accounted for using this algorithm.

Furthermore, Newman's algorithm constrains nodes to single communities. While for some categorizations (e.g. undergraduate institution, hometown, etc.) having one unique membership makes sense, for other categorizations (e.g. food preferences, workplace history, ethnicity, etc.) nodes should be able to part of multiple clusters. In particular, individuals that bridge two communities in a real-world social network are coerced into one particular community.

Lastly, since the quality index in Newman's algorithm is based on a ratio of in-community and between-community edge counts, small data sets with poor resolution on the edge counts will produce community divisions that are very sensitive to few edges. The limited resolution leads to community structure that is very susceptible to error. While Newman discusses how the GN algorithm is more accurate than his new algorithm on small networks, he does not explore other algorithms that, even if it is proportionally more computationally expensive, is feasible on small networks where this paper's algorithm is inaccurate.

On Spectral Clustering: Analysis and an algorithm

The largest benefit of this spectral clustering algorithm is the efficiency of the algorithm without diminishing consistency or accuracy of results. The authors argue that the algorithm works well even for datasets that do not have convex regions or are not clearly separated, which they show with a range of experiments that they have performed.

This algorithm is an improvement over traditional spectral graph partitioning, which divides the graph into two sets of nodes and then applies these methods recursively to find k clusters. Instead, this algorithm directly computes k partitions by using eigenvectors.

However, a possible limitation of the algorithm occurs when clusters vary substantially in the degree to which they are connected. This situation may result in bad clusterings because of difficulties in finding clusters where nodes are closely connected in absolute terms but appear to be loosely connected relative to larger and more strongly connected clusters in the network.

Additionally, the authors did not define a method of evaluating the accuracy of the clusters formed by their algorithm. The evaluation consisted of find clusterings consistent with 'what a human would have chosen'. This subjective method of evaluation makes it difficult to determine the success rate of the algorithm compared to other spectral clustering algorithms.

Clustering by fast search and find of density peaks

A unique characteristic of this approach is the definition of a "border region". Nodes are in a border region if from within their own assigned cluster they still fall within the "cutoff distance" of other clusters. A node in the current cluster whose density is higher than that of the point of highest density within the border region is considered part of the cluster core. Otherwise it is considered noise. This avoidance of a noise cutoff allows for recognition of non-spherical clusters. It also allows for capturing the shape and probability peaks of clusters with very different densities.

The approach allows for nodes to be "split" between clusters, yet still allows for such clusters to remain pure (i.e. the node is still an unambiguous and valid member of both clusters). This is a direct advantage over other approaches such as Newman's, also discussed above.

The algorithm is less effective with datasets that contain a comparable number of nodes and clusters, as well as smaller datasets. One reason for such is that node density would be heavily affected by each node, and especially so outliers. Slight modifications to the original approach (e.g. introducing a cutoff distance) yielded comparable results to the unmodified algorithm.

III. Proposal

A. Problem:

Individuals in social networks are often unaware of people who share similar tastes, literally. From our own personal experience, we believe that people often bond over food, and so we want to detect communities of common food preferences. To do so, we want to use Yelp user, business, and review data and find similar ratings between users. Realistically speaking though, communities are much more likely to form when the members are in close physical proximity with each other, so our metric will also be sensitive to check-ins to the same restaurants, which gives us geographical data.

B. Data set:

This project will make use of a data set provided by Yelp to form our community network. The range of businesses is limited to locations in the area of Phoenix, AZ. Furthermore, the data set is limited to dates ranging from January 2011 to January 2013.

The Yelp data set provides the following information.

Object Type	Count
Users	70,817
Business	15,585
Reviews	335,022
Tips	113,993
Check-In's	11,434

We will be focusing specifically on the users as nodes and the reviews and check-in's as events in our network.

To evaluate our results (which are described further on this proposal), we can use the Business object which contains a list of categories that each restaurant is associated with. For example:

"categories":
["Bakeries", "Food", "Breakfast & Brunch", "Sandwiches", "Restaurants"]

Hence, we have sufficient data to represent a network of Yelp users and calculate food similarities between users. Furthermore, we have a method of evaluating the resulting community clusters of our project.

C. Model:

Our model will generate an undirected network of Yelp users where each user will be represented by a node. Edges representing the similarity in food taste between users will be created and weighted based on a composite score. This score (described below) roughly estimates similarity in food tastes. We can then establish a threshold

composite score that signifies compatibility between two users.We would then detect communities of food tastes based on this community network.

We have two options of defining our edges - either weighted or unweighted. If we choose to have unweighted edges, the threshold composite score would be used as an indicator variable. Composite scores higher than a given threshold would result in an edge in the community network (a separate network from our internal representation), while scores below the threshold would not. On the other hand, weighted edges would allow us to use the food taste compatibility scores and, after normalizing these scores, assign weights to any edge between users. This allows us to create edges between every pair of nodes in the network. We can make this choice.

The composite score representing the weight between two users is calculated based on reviews that users have made for the same restaurant. The similarity in the degree of happiness in the review (represented by the stars given) will create a more positive or more negative edge. Furthermore, the number of check-in's at the same restaurant by the two users will contribute positively to the score. This will also take into account a growth factor where increasing number of check-in's add exponentially to the score.

D. Algorithms:

The three research papers each proposed a different method and algorithm to cluster nodes. We can apply these algorithms in our Yelp user network to find communities of people based around their food taste. Ultimately, we will choose two of these algorithms to implement on our actual dataset and compare the accuracy of the two.

Newman's Algorithm

Newman's algorithm applies moderately well to our data set. Newman's algorithm uses an undirected, unweighted graph, and one of our unweighted network model could leverage the algorithm to produce clusters. The indicator variable in the unweighted model is essentially our guess as to whether or not two particular Yelp users would have compatible food tastes, and an edge would be drawn whenever the composite score exceeds a particular threshold. With a sufficiently high threshold, we know that the graph would be far from complete, and thus Newman's algorithm, which operates in $O(n^2)$ time on sparse graphs, would compute quickly.

However, Newman's algorithm does not adequately model a real-world social network in a variety of ways. First, the strength of a relationship between two individuals is an analog metric, and clustering based on the indicator variable coerces the metric into a binary dichotomy. As such, all relationships past the composite score threshold looks the same, while all relationships below the threshold look the same as well. It is unclear how much predictive power is lost (i.e. there might be a "tipping point" for degree of similarity for a relationship to form that, if we can successfully identify and pin the threshold to, would still produce accurate clusters) by using the binary dichotomy.

Furthermore, Newman's algorithm forces each node into a single community to which it best belongs. However, an individual's food preferences can be varied and diverse, so they might belong in multiple clusters. Newman's algorithm wouldn't be able to account for that nuance, and so many otherwise very similar food tastes between two individuals may be lost to the final clustering that the algorithm identifies.

Spectral Clustering

The spectral clustering algorithm would apply very well to our data set. First of all, spectral clustering relies on an undirected weighted graph with nodes corresponding to data points and edges showing relationships or distances between points. Our model uses an undirected weighted graph as well, taking different Yelp users as nodes and creating weighted edge based on users' food taste similarity. The similarity matrix that the spectral clustering algorithm is analogous to defining the similarity in food preferences of our range of users.

Furthermore, our project requires weighted edges. Food similarities between users require data to be preserved in the edge representations between nodes. The edges will be formed with a specific value weight that represents a closer or more distant relationship with other users.

Additionally, spectral clustering takes into account the possibility of users who have a strong presence in multiple communities, essentially bringing the idea of overlaps between communities. The nested and overlapping structure of real community networks are represented by the interwoven edges between members of the graph.

However, a challenge we will face in implementing this algorithm will be evaluating the accuracy of a certain clustering result. This evaluation is necessary to optimize what value of *k* clusters we use in our *k-means* algorithm. The difficulty of a consistent method of evaluating clustering accuracy is one that many researchers today are still attempting to solve. To mitigate this challenge, we can use two measures to consistently measure the success of our clusters - the conductance of clusters (representing the quality of clustering) and the ratio of the weight of inter cluster edges to the total weight of all edges (representing the cost of clustering).

Density-Based Clustering

The density-based clustering (DBC) algorithm applies well to our dataset. Allowing nodes to be members of multiple clusters (all the while maintaining individual clusters' purity) will ensure that users (as nodes) can be members of multiple food communities.

It is not unreasonable to assume that the Yelp dataset contains communities with large variance in density between them. These can still be captured by the DBC algorithm, so that both small (e.g. low population, small geographic area) and large but loosely tied communities (e.g. large total but low per-capita number of check-ins) can be captured.

The use of distances in the DBC approach can also be translated to the use of weighted undirected edges in our model, where larger distances could be modeled by larger negative values, and shorter distances could be modeled by larger positive values. A possible challenge is in the incorporation of negative edges, as the DBC method assumes only attractive relationships.

In terms of network structure itself, the Yelp dataset appears to have a significant enough difference between the number of nodes and possible clusters such that our network is not one that is difficult for the DBC to analyse (such cases are discussed in the Critique section).

E. Evaluation:

The evaluation of our clusters will be tricky as there are no real user communities grouped by food taste on Yelp. However, we can find the most loved and most hated categories of restaurants for each community (based on the intersection of restaurant categories .most loved and hated by the users in that community). Specifically, we will take the reviews of each user and count the types of restaurants that generated a positive review and the restaurant categories where the user provided a negative review and curate a list of restaurant categories that the user enjoys and dislikes the most.

We can then use these category listings to compare a user's tastes to the tastes of the community. The percentage of overlap between the user's list of loved and hated restaurant categories and the community's list will allow us to measure how closely aligned the food tastes of the user and the community are. The average of this quantity across all users in the community will then give us a score of how well our clustering performed.

F. Extensions:

We are using a static data set, so we cannot model how the network changes over time. With real-time data, we could use composite scores to inform optimization--since our internal representation is a large complete graph, it

would be computationally expensive to update the composite score frequently. Rather, we could revisit each edge (and recalculate the composite score between two users) at a frequency proportional to the composite score.

An interesting extension could be to label the clusters defined by the model. The clustering algorithms use edges to form clusters but do not give us reasons behind this clustering, hence we can provide specific names or 'labels' to each food taste cluster. We can determine the labelling of clusters through methods such as looking at the single most popular restaurant category loved by the users in the community.

Works Cited

Newman, M. "Fast Algorithm for Detecting Community Structure in Networks." *Physical Review E* 69.6 (2004): n. pag. Web. 12 Oct. 2014. <http://http://arxiv.org/pdf/cond-mat/0309508.pdf>.

Ng, Andrew Y., Michael I. Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an Algorithm." *NIPS* (2001): n. pag. Web. 11 Oct. 2014.

Rodriguez, A., and A. Laio. "Clustering by Fast Search and Find of Density Peaks." *Science* 344.6191 (2014): 1492-496. Web. 12 Oct. 2014.