Exposing commercial value in social networks: matching online communities and businesses

Murali Narasimhan muralina
Camelia Simoiu csimoiu
Anthony Ward tonyward

December 13, 2014

Abstract

This paper explores the problem of user recommendations in group settings. Research in networked social behavior has found evidence that individuals tend to form connections to those having similar interests to their own. Known as homophily, this tendency has been shown to significantly inform the structure of social networks. Collaborative filtering (CF) techniques, however, typically do not take into account explicit social relationships among users in predicting user preferences, although the importance of peer influence in user consumption and marketing has long been recognized. We test for the existence of homophily on the Yelp! dataset and find ample evidence that social ties are indicative of similar preferences. Based on this insight, experimental results show we are able to improve on traditional CF methods by 1) using clustering algorithms to detect communities of users with similar tastes, and 2) incorporating information about social structure in CF techniques. We find that leveraging social structure information improves the performance of CF and reveals relevant and valuable matches between businesses and the members of the group. These results have important implications for targeted marketing.

Keywords — Collaborative Filtering, Homophily, Social Network, Community Structure, Clustering, Recommendation System, Gaussian Mixture Model, K-means

1 Introduction

A growing trend in the movement of e-commerce is the integration of online social network information in websites with user generated content (UGC). With rich data on user purchase history and preferences, recommender systems have become a key tool for many businesses in providing users with personalized recommendations on products such as music, movies and books. Most user recommendation systems to date, however, have been typically ego-centric, focusing on recommendations for individuals based on preferences of 'similar' users in the system. We propose to add a socialized dimension to recommendation systems that leverages the structure of a user's social network and the preferences of their connections. This is based on two insights: firstly, consumption is often social (restaurant-going, fitness activities, sports, etc.); secondly, a user's connections often provides valuable information about their preferences. In such cases, marketing efforts may benefit from targeting particular groups of users with similar interests to increase the likelihood of purchase. For instance, a business may offer Groupon-type deals or discounts to such groups for collective-purchase.

We incorporate these insights into analyzing the social network and business dataset provided through the Yelp! Data Challenge. Our goal is to detect communities of users with similar preferences and predict which businesses (restaurants) the group as a whole is likely to prefer, and determine whether incorporating user network information improves accuracy. In the context of Yelp!, two users are thought to have similar preferences if they have both reviewed the same or similar restaurants and have submitted a similar star rating.

We first test the ability of various clustering algorithms to detect communities of users with similar restaurant preferences. Secondly, we establish a baseline CF model that predicts star rating for a user-business pair, and measure the change in prediction performance after incorporating social network information. Section 2 includes background information and previous work, Section 3 provides an overview of the exploratory data analysis, Section 4 outlines the models and methodology used, Section 5 presents results, Section 6 outlines the conclusion and future work.

2 Previous Work

A growing body of research gives evidence for the principle that similarity breeds connection. Known as homophily, this principle is hypothesized to structure network ties of every type (marriage, friendship, work), so that individuals' personal networks are often homogeneous with regard to many behavioral and sociodemographic characteristics (McPherson *et al.* (2001)). More intuitively, we often turn to friends for recommendations on products unknown to us, or new restaurants.

We apply this insight to adapt traditional collaborative filtering (CF) algorithms by leveraging information about a user's social structure. CF is a technique that makes automatic predictions about the interests of a user based by finding other users with tastes that are similar to the target users, based on their preference history. More specifically, classical CF methods incorporate a version of the K-nearest neighbors (kNN) Song *et al.* (2007) in order to find the k most similar users according to a particular measure of similarity (inverse Euclidean, cosine similarity, etc.). A key assumption of CF is that individuals who agree in the past tend to agree again in the future. As a result, CF first finds users with similar preference to the target user's and makes recommendations to the target user aggregating the ratings of their top-K similar usersKoren (2009), Gross *et al.* (n.d.).

There are a number of challenges in traditional CF systems. Sparsity is an important issue, as even if a user is very active, the number of items largely exceeds the number of products a user purchases or reviews, leading to a very sparse user-business matrix. Since predictions are based on similarity measures computed over the co-rated set of items, large levels of sparsity can lead to poor accuracy Jameson (2004). Other challenges that arise in designing recommendations for groups are: defining a similarity metric to identify and aggregate groups of users with similar preferences, and designing an algorithm to generate recommendations based on the aggregated preference profile of the group (Gong (2010), Jameson (2004), Terveen & McDonald (2005)). Moreover, We construct a feature matrix of business attributes and use this to define a similarity score that captures the affinity between users tastes and opinions. In order to overcome the data sparsity problem, we use PCA to reduce dimensionality. PCA also makes our feature space smaller and computationally allows us to explore a number of clustering algorithms in order to detect groups of users with similar preferences.

3 Data and Exploratory Analysis

We use data from the Yelp! Dataset Challenge, which contains the 2013 crowd-sourced reviews for businesses in four cities, businesses and their attributes, and social network of users (friend lists of users). We limit our analysis to Madison, as this city had a good balance of a sufficiently large number of users business pairs and one of the least-sparse graphs. We further limit our analysis to the `restaurant` category, as this contains the largest set of reviews, based on the intuition that someone's preferences in other categories (Clinics, sports clubs, etc.) will not be a good indicator of their restaurant tastes.

We define the social network for a given city as a graph $G = (U, E)$ where U is the set of users and E is the set of edges representing an explicit friendship between two users displayed publicly on Yelp!, much like the connections existent other social networks such as Facebook. Let $B = \{b_1, b_2, ...b_n\}$ be the set of businesses in the given city, and A_{b_i} be the set of attributes for the business i. These include features like `noise_level`, `alcohol`, `patio`, `good_for_families`, `divy`,`ambiance` etc.

Basic statistics of the data set is presented in Table 1. We observe that the social graph is sparse, with over 50% of users having zero degree (no friends within the network) and low clustering. The degree distribution of users approximates a power law distribution. We observe a heavy tail for high-degree nodes meaning that most users have very few connections, while a few users dominate the connectivity and have a large number of connections. (Figure 1). This graph is dominated by a single large component comprising of approximately of 29% of users for Madison. In order to appropriately test the effect of network structure, we further limit our analysis to the users in this connected component, ie. those users who have at least one connection within the component.

To test our hypothesis that explicit user connectivity is better indicator of restaurant preference, we compute the percentage of categories for which users have submitted reviews in common. The baseline randomly sampled 200 nodes from the graph and calculated the percentage of pairs that had reviewed at least one common category. The dyad set is defined as the set of pairs of users that are connected by an edge. A triad is a set of three nodes such that there exists a tie between every pair of nodes. For all sets, we only consider users that have degree greater than

Metric	Madison
Number of Users	8,729
Undirected Edges	6,962
Zero Deg Nodes	5,907
NonZero In-Out Deg Nodes	2,822
Connected component size	0.291328
Closed triangles	9,723
Frac. of closed triads	0.040044
90% effective diameter	5.382648
Clustering coefficient	0.054752

Table 1: Network Statistics

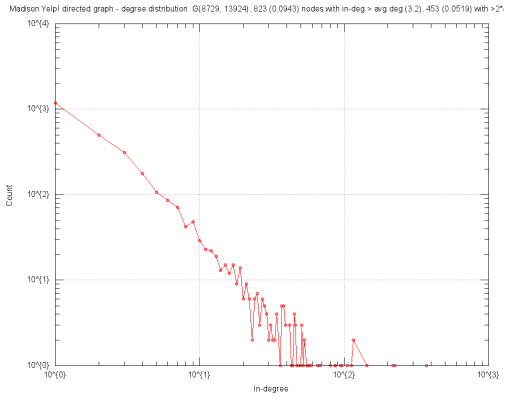

Figure 1: Degree distribution of Yelp Users for users in Madison.

zero (ie. those in the weakly connected component). Despite the sparsity of the graphs, we observe almost a 100% increase in the percentage of users having reviewed a common category from the baseline to the dyads, and a 300% increase from baseline to triads, confirming our hypothesis [1].

Baseline	Dyad Set	Triad Set
0.0565	0.1047	0.1950

Figure 2: Percentage of Users reviewing at least one restaurant category in common.

4 Methodology

4.1 Feature Extraction

For each user u_i, we construct a 450-dimensional binary feature vector from the attributes of the businesses reviewed by u_i. In the case of categorical variables with more than two levels such as `price_range`, we define a binary variable for each level. The vector is constructed by computing the weighted mean of business attributes from all the user's reviews:

$$score(u_i) = \frac{1}{5R} \sum_{r=1}^{R} s_{ri} * b_r \quad (1)$$

where R is the number of reviews user u_i has submitted for the given city, s_r is the star rating gave to the business in review r, and b_r is the $1x405$ binary attribute vector for the business being reviewed. This quantity is normalized by 5R, as 5 is the maximum star rating possible. The intuition is that a user's preferences as captured through their aggregate reviews will provide a strong signal for the archetypal restaurant the user prefers. For example, if a user consistently gives high ratings to restaurants with live music, smoking and outdoor patio, and a low rating to restaurant with attributes: smoking, high noise level, formal, high price range, the aggregated feature vector will capture this ranking of preferences. Higher importance will be placed on the features that were present for highly-scored businesses, and lower importance for features that were present for poorly scored businesses. We can then begin to ask which businesses are most alike those preferred by the user (ie. which have the attributes most highly ranked).

For every existing connection in the network between two users u_i and u_j, we also construct 5 different social features:

- `percentage of friends in common`
- `average degree of` u_i `and` u_j
- `maximum degree of` u_i `and` u_j
- `average degree centrality of` u_i `and` u_j
- `maximum degree centrality of` u_i `and` u_j

A high percentage of friends in common is likely to indicate the existence of a strong tie between users u_i and u_j, an increased likelihood of having similar tastes and being friends in reality. Degree and centrality are both measures of the popularity of the users. If a pair of users is connected to a large number of people or is very central to the network, it may be an indication that they are influences, trend-setters and likely to be a source of influence to their friends' opinions and tastes.

[1] This test was also used as part of Camelia Simoiu's CS224 project.

4.2 Community Detection

Subsequent to normalizing the feature matrix, we apply PCA dimensionality reduction in order to speed up computation and correct for possible correlation among the features. In order to discover communities of users with similar preferences, we implement the K-means algorithm as a baseline. We determine the optimal K to be equal to the number of principal components that account for 95 % of the variance **??**. We represent each of the K communities identified by its centroid, and use the inverse euclidean distance to capture the similarity, or 'affinity' of the group's preference for each business and between users for community detection. Specifically, if u and v are two vectors in this feature space, then the similarity $Sim(u, v)$ between them is computed as

$$Sim(u, v) = \frac{1}{1 + norm(u - v)}$$

where $norm(u - v)$ represents the Euclidean distance between u and v.

We have split the dataset into train and test sets so that approximately 80% of reviews are in the train set and 20% in the test set. The cutoff point was temporal, so that reviews before October 2013 were in the train set, and reviews after October 2013 were in the test set. This allowed the same users to be in the train and test sets in order to test the accuracy of the clustering algorithms.

To evaluate how well the clustering methods perform, we use the traditional evaluation metric of *pairwise accuracy* for ranked results in Information Retrieval. We compare how well each cluster predicts relative preferences of the users within the cluster. And by assessing pairwise preferences, we are comparing the global ranking of preferences for each user with those predicted by the cluster that the user is assigned to. More specifically, we define the accuracy to be the fraction of business pair preferences correctly predicted for all users.

We further try to improve on this baseline with various other clustering techniques including: Minkowski-Weighted K-means clustering (Amorim & Mirkin (2012)) and a Gaussian Mixture Model (GMM). The Minkowski Weighted K-Means (MWK) has the advantage that it incorporates feature weighting in K-Means and has been shown to achieve better accuracy, particularly when dealing with noisy datasets. The objective function is as follows:

$$J = \sum_{j=1}^{k} \sum_{i \in S_k} \sum_{v=1}^{V} w_{kv}^{p} \, |y_{iv} - c_{kv}|^{p} \qquad (2)$$

where w_{kv} is the weight of feature v in cluster k, and p is a user-defined parameter tuned to achieve better results.

We contrast this to the model-based approach of the GMM model, which uses the iterative expectation-maximization algorithm to fit a probabilistic model to the data. We expect the GMM to improve on K-means if the data in the 'K' component distributions is densely distributed around its centroid, and the mixture model covers the data well (ie. the component (normal) distributions are able to capture the dominant patterns in the data well). GMM have the added advantage in that they offer flexibility in choosing the component distribution and we are able to obtain a density estimation for each cluster. Our other motivation in trying a Gaussian mixture model is that it can be thought of as a soft clustering method, since the posterior probabilities for each point indicate that each data point has some probability of belonging to each cluster. All models used user and business data projected onto the hyperspace defined by the principal components.

4.3 Collaborative Filtering: Predicting User Ratings

We contrast the classical collaborative filtering technique as the baseline, as well as modified CF algorithm that focuses on the structure of a user's social network. Our goal is to predict the star rating given by a user u_i to a particular business b_i. We implement the N-nearest neighbors algorithm (kNN) in Song *et al.* (2007) for the CF baseline in order to find the top 'k' similar users having also reviewed b_i. Similarity is defined as the inverse Euclidean distance between the u_i's business attribute vector, and those of the other reviewers for b_i. The predicted star rating is then the average rating of the k nearest neighbors. We contrast the performance of this algorithm to a modified version where we use average of u_i's connections to predict the star rating, weighted by the social features as described in Section 3, and use linear regression to learn the optimal weights of each of the features. For example, if a u_j is a less popular connection (ie. lower degree or less central user) their star rating will factor less than another user with higher degree or centrality. Similarely, the opinion of a connection having more friends in common

with u_i will weigh more than one with less friends in common.

5 Results

5.1 Community Detection

After running PCA on the feature matrix, we find that 34 principal components explain almost 99% of the data. We construct a histogram of the percentage variance explained versus principal components (also known as a Pareto plot) and visually inspected the incremental variance explained by each additional component (not included as this is one of the standard approaches to determining K) Ding & He (2004). We then project the train, test and business data onto this reduced feature space.

Results for the three clustering algorithms can be found in Figure 3. We observe that K-means with PCA dimensionality reduction achieves the highest accuracy with a score of 70.50%. That is, we are able to correctly predict with 70.50% the rankings of businesses preferences for each user from the businesses rankings predicted by the cluster centroids. MWK-Means and GMM achieve similar and slightly lower accuracy scores (68.75%) with the chosen parameters. We explored sensitivity to parameters for all three algorithms, however it was computationally expensive to do an exhaustive grid search of the parameter space to find the best parameters within the timeline of the project. This may explain why we see lower accuracy scores for the MWK-Means and GMM models.

All three algorithms are sensitive to centroid parameter initialization, K must be known in advance. In addition, K-means is known to perform non-optimally for sparse, high dimensional data. This is because k-means tends to be sensitive to outliers, which is especially the case in high dimensional data sets. Since it uses squared deviations, any extreme value will have a large effect on the least squares metric. In addition, MWK-Means has additional parameters that must be tuned, which becomes increasingly difficult to do with noisy, high dimensional data. This may be because there are insufficient data points per mixture, resulting in a poor estimator of the covariance matrices. We suspect that outlier ratings may be preventing us from achieving higher accuracy scores. For example, businesses with low-popularity, or those with features that are rarely seen in the rest of the data may be influencing the centroid.

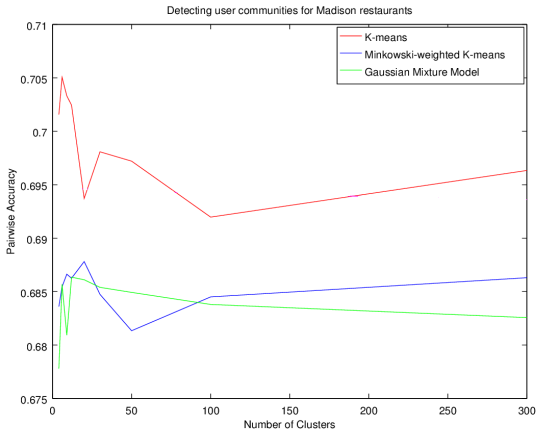

Figure 3: Clustering results for Madison.

As a check, Figure 3 show plots of K versus accuracy scores. We find that all three clustering algorithms achieve the highest accuracy for K 10. Increasing K does not lead to further improvements. We suspect that there is a lot of noise in the data, especially due to low activity users (those with few reviews submitted), businesses with few reviewers or rarely-seen attributes, which are lowering the accuracy values. Time permitting, a possible solution to this would be to restrict the set of businesses to the top 50-100 or set a lower-bound on the number of reviews. Another possible explanation is that there may not be as much heterogeneity in the data as we expected. While PCA has the advantage of speeding up computation, we may be loosing a lot of the heterogeneity encoded in the original business attributes.

5.2 Adding Social Features

Using the optimal number of clusters, we find that including social affinities between users into the clusterin algorithm improves the pairwise accuracy score by approximately 3%, which is a much smaller increase than expected. We suspect that the improvement was not even higher because of data sparsity, as we find there are many busineses with few reviewers. As we increase the number of K, even users that have given dissimilar scores will be included in the 'nearest neighbors' set, resulting in a possible bias to the prediction. The same holds for CF augmented with social features. The limit of common reviewers for a business is small, and we find that there are few edges in the network reviewing the same businesses. However, the small number of such reviewers that are found seem to be contributing to increasing the accuracy score.

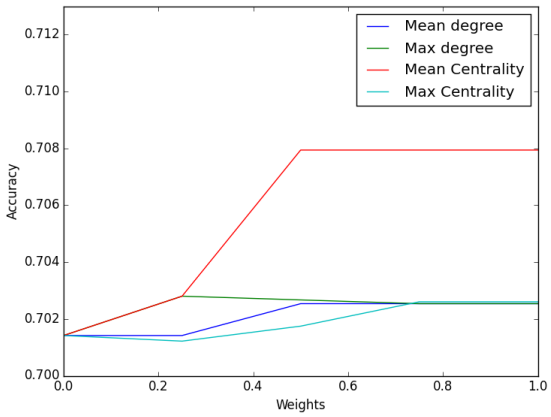

Figure 4: Sensitivity Analysis of Feature weights.

We update our similarity scores between user vectors u and v by adding a value corresponding to the social affinity between u and v as given below

$$Sim(u, v) := Sim(u, v) + W' * Social(u, v)$$

where W is the weight vector corresponding to the social features and $Social(u, v)$ is the social feature vector representing the connection between u and v. To find appropriate values for the weights, we analyzed how sensitive the accuracy was to each social feature with various values for the weights. The results of our sensitivity analysis is presented in 4.

5.3 Collaborative Filtering Algorithms

For our analysis of the collaborative filtering algorithm, we compare how well kNN performs with and without social features for various training set sizes varying from 500 to 3000. We fix k=10 and we keep the test size set constant at 500 for all our runs. The results of our analysis using mean-squared error (MSE) is shown in 5. We find that for all training set sizes, the training error and test error are both lower with social features than without. This clearly indicates the value of adding appropriately weighted social features to a traditional collaborative filtering algorithm like kNN. More specifically, we find that incorporating social features into the CF algorithm reduces the MSE by approximately 1% from 1.1740 to 1.1640. Further, as we add more training data, the training error and test error decrease for kNN with social features.

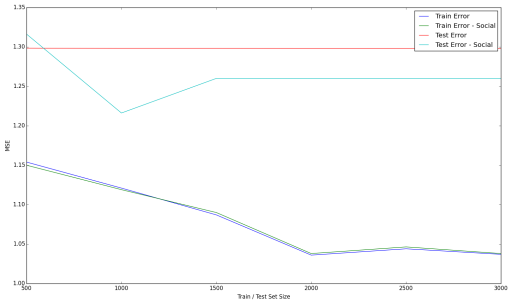

Figure 5: MSE as a function of train and test size.

6 Conclusions and Future Work

Despite the various challenges presented in the data set (sparse review data and social network graph), we find evidence that prediction accuracy of community detection and star ratings are improved by exploiting information about a user's localized social network structure. We are able to slightly improve MSE of the CF algorithm by exploting social features, as well as the pairwise accuracy of the K-means clustering algorithm. As a proof of concept, this results implies that similar improvements to CF recommendation systems may be a promising area of future research.

There is room for improvement to our methodology and data processing. Incorporating the content of reviews will definitely add an additional dimension of information that could be used to determine the similarity of the users. For instance, negative versus positive comments as related to topic models might be one way to gain more insight into user preferences relating to service, ambiance, food quality, etc. Experimenting with different distance and similarity functions (eg. cosine similarity) might lead to small improvements in results, however we feel that it would not drastically improve our methodology. Secondly, since our objective is to predict the most well-liked business for a community of users, our test function could be modified to exclude businesses that are, not in the top X% most rated businesses. We are currently learning the weights using linear regression, however softmax regression might be better suited, as the star rating can only take on integer values between 1-5. In terms of clustering algorithms, it may be worth exploring soft k-means and hierarchical models. Hierarchical models would have the advantage of outputting a dendrogram, which is more informative than the flat, unstructured set of clusters

returned by k-means and does not require the prespecification of the number of clusters as K-means does. These methods may also be an effective means of removing outlier reviews and obtaining more cohesive clusters.

References

Amorim, R.C., & Mirkin, B. 2012. Minkowski Metric, Feature Weighting and Anomalous Cluster Initialisation in K-Means Clustering. *Pattern Recognition*.

Ding, Chris, & He, Xiaofeng. 2004. K-means clustering via principal component analysis. *Page 29 of: Proceedings of the twenty-first international conference on Machine learning*. ACM.

Gong, Songjie. 2010. A collaborative filtering recommendation algorithm based on user clustering and item clustering. *Journal of Software*, **5**(7), 745–752.

Gross, Tom, Masthoff, Judith, & Beckmann, Christoph. Workshop on Group Recommender Systems: Concepts, Technology, Evaluation (GroupRS).

Jameson, Anthony. 2004. More than the sum of its members: challenges for group recommender systems. *Pages 48–54 of: Proceedings of the working conference on Advanced visual interfaces*. ACM.

Koren, Yehuda. 2009. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, **81**.

McPherson, Miller, Smith-Lovin, Lynn, & Cook, James M. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, **27**(1), 415–444.

Song, Yang, Huang, Jian, Zhou, Ding, Zha, Hongyuan, & Giles, C Lee. 2007. Iknn: Informative k-nearest neighbor pattern classification. *Pages 248–264 of: Knowledge Discovery in Databases: PKDD 2007*. Springer.

Terveen, Loren, & McDonald, David W. 2005. Social matching: A framework and research agenda. *ACM transactions on computer-human interaction (TOCHI)*, **12**(3), 401–434.