# Discovering Emerging Businesses

**Arjun Mathur, Chris van Harmelen, Shubham Gupta**

## Abstract

In the last few years, extensive research has been done on user-item graphs in order to enable modern users to easily find interesting new items and tap into the so-called "long tail". We focused on the problem of restaurants in a network which have limited amount of reviews. We intend to understand the impact that initial reviewers have on predicting the success of a business, and using these insights to aid in developing a recommender system for restaurants with low review counts. Since traditional collaborative filtering and latent factor models perform poorly with little data, we explore how SimRank and the Status of initial reviewers can help predict how successful an emerging business will be.

## 1. Introduction

### 1.1. Motivation

Starting a new business, particularly a restaurant, is always a risky endeavor. With the help of business rating and recommending applications such as Yelp and Google Local, businesses are able to establish a public reputation as well as advertise their business through these reputable outlets.

There are individuals who's reviews are more influential than others', as users naturally have different reputations. Some users may write funnier reviews than others, or they might write more in depth reviews. A reviewer may also have more experience in reviewing restaurants, such as food critics, who generally have some influence over how well a restaurant does, solely based on their reputation and how positive the review they write is. Most reviewers on Yelp are not food critics, but rather regular individuals who are sharing their experiences. Emerging businesses have the potential to greatly benefit from a positive review from an influential user with many followers or simply from more total positive reviews.

This begs the question: how successful will an emerging business be given they have a handful of reviews on Yelp? With a large number of reviews, it is fairly straight forward to predict how well a business will do in the future as there is a lot of history. But when looking at businesses which have a small number of reviews, it is quite difficult to predict how well the business will do in the future. This problem extends into many other situations, such as drug development and testing. How well a drug performs on an individual patient can be thought of as the score of a review and can be used to help predict how successful a drug will be if further research is pursued. With drug development being a very expensive and time-consuming process, it is optimal to spend little to no time and money on developing a drug that turns out to be ineffective or harmful.

The rest of the report is organised as follows: In section 2 we discuss related literature and prior work done in this area. In section 3 we describe our graph formulation and general statistics of the network. We then explain the model and algorithms used for finding the potential customer set of a new emerging restaurant in section 4. Section 5 covers the evaluation criteria and results. We then conclude this work in section 6.

### 1.2. Dataset

We use the dataset provided by the Yelp Dataset Challenge. This contains data of various users, businesses and their activities from 5 cities: Phoenix, Las Vegas, Madison, Waterloo and Edinburgh. Table 1 shows the summary statistics from the dataset.

## 2. Prior Work

### 2.1. Effects of User Similarity on Social Media

In (Anderson et al., 2012), authors investigate the effects of similarity between two users and their status on evaluations that they give each other. The key idea

*Proceedings of the 30$^{th}$ International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

| | |
|---|---|
| Users | 252,898 |
| Businesses | 42,153 |
| Business Attributes | 320,002 |
| Reviews | 1,125,458 |
| Unique categories | 5945 |

*Table 1.* Summary statistics of Yelp dataset.

developed in this paper was to predict the outcome of group evaluations by only examining a few initial individual evaluations and their characteristics.

The basic idea behind this algorithm is that similarity between users and their relative status plays a big part in the decision making process. Thus they are good enough features to be able to predict the outcome of the evaluation process. This approach is tested on various datasets like Wikipedia admin election data, Stack overflow and Epinion and have showed favorable results towards incorporating the above mentioned characteristics in the evaluation prediction model.

This very idea of using the similarity and statuses of few initial users to predict the outcome in the future was very fascinating to us. We developed our model for predicting the potential customer set of a new business based on the similarity and status of the initial reviewers of the business and saw the effect of these 2 metrics on the business's performance.

### 2.2. SimRank: A Measure of Structural-Context Similarity

In (Jeh & Widom, 2002), the authors propose a method for measuring similarity between objects in logical graphs. The concept is based on the idea the intuition that two objects are similar if they are related to similar objects. The majority of similarity metrics such as vector-cosine similarity and the Pearson correlation (used for collaborative filtering, finding similar document, etc.) rely on immediate connections in a graph to compute similarity. On the other hand, SimRank uses a recursive algorithm to calculate similarity between two nodes based on the context of the graph structure. Formally similarity $s$ is calculated as:

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

Where $I_i(a)$ is the $i^{th}$ in node of node $a$ and $C$ is a decay constant. For the base case, the similarity of an object with itself $s(a,a) = 1$. These equations are subsequently generalized to apply to Bipartite graphs and calculation algorithms are described.

| | |
|---|---|
| No. of Components | 267 |
| Fraction of nodes in largest connected component | 0.997 |
| Fraction of edges in largest connected component | 0.999 |

*Table 2.* Connected Components Analysis

### 2.3. Scalable Similarity Search for SimRank

Due to the recursive nature of the SimRank algorithm, similarity calculation is extremely expensive. In (Kusumoto et al., 2014), the authors describe an approximation algorithm to computing the Single-Source similarity problem (finding the top k most similar neighbors for a certain node). This is done by linearization of a the recursive formula. In matrix notation, SimRank can be described as:

$$S = (cP^TSP) \vee I$$

Where $P$ is the transition matrix, $S$ is the similarity matrix, and $\vee$ denotes the entry-wise maximum function (Since $S_i i = 1$). Although the $\vee$ fucntion is not linear, this can be linearized by introducing a diagonal matrix D such that

$$S = cP^TSP + D$$

Using this linear reconstruction, a monte-carlo method to determine single-pair simrank score is combined with upper-bound pruning heuristics in order to very efficiently (and with high accuracy) calculate SimRank.

## 3. Dataset Analysis

### 3.1. Graph Formulation

As the initial step, we parsed the dataset to discover information on users, businesses and reviews. The Yelp dataset was then formulated as a user-business weighted bipartite graph. There exists a weighted edge from a user to business if the user has rated the restaurant in the past with the weight of the edge computed using the user's rating to that restaurant and user's overall rating. There are no user-user or business-business edges in the graph. We have done some initial analysis on this graph to understand the basic properties of it's structure.

### 3.2. Connected Components

It is clear from Table 2 that the user-business graph consists of a very large connected component which has around 99.7% of the total nodes of the graph. The rest of the components have very few users who have rated very few restaurants which are not voted bu any other users from the network. Thus, for further analysis we would be considering only the largest connected component of the graph as other components would not have any significant contribution to the final results.

### 3.3. Degree Distribution

As, the next step we analysed the degree distribution of both he user nodes and businesses nodes. For a user, it indicates the number of businesses he/she has rated and for a business, this means the number of users which have rated it. Interestingly, both the distributions seems to follow a power law distribution as shown in fig 1 and 2.

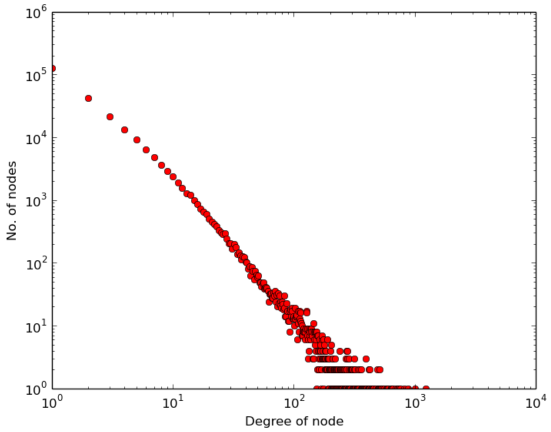

*Figure 1.* User Degree Distribution ($\alpha = 2.39$)

This indicates that the graph grows by preferential attachment, i.e., if a new business enters into the graph then it would probably be rated by a user who already have rated a large number of other businesses. Similarly if a new user enters the graph then he/she would most probably be rating the already popular business.

### 3.4. Rating Distribution

We also analyzed the distribution of average star rating of both the businesses and the users in the yelp dataset. As is the case with most of the other datasets (eg. Netflix, Movielens), the rating distribution is highly skewed towards the higher side of the rating scale. Majority of the businesses have an average rating of around 4 which is evident from the fig 3 and majority of the users have an average rating greater than or equal to 3.5 (fig 4). Due to this skew nature of the ratings, we have decided to consider the ratings greater than or equal to 4 as favourable and anything below 4 as unfavourable rating.

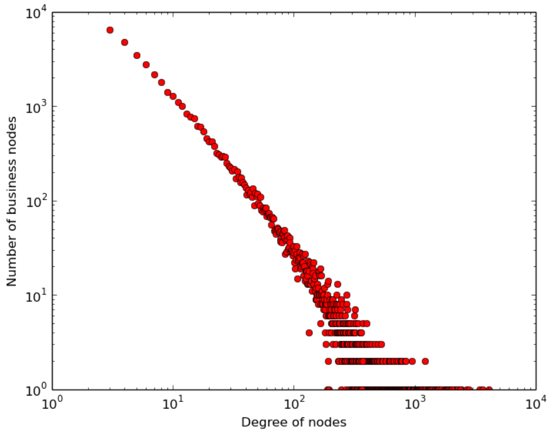

*Figure 2.* Business Degree Distribution ($\alpha = 1.79$)

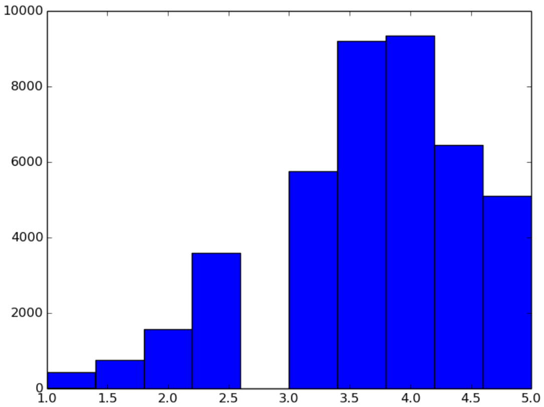

*Figure 3.* Average Star Rating Distribution for Businesses

## 4. Model and Algorithms

The main aim of our work is to predict the set of potential set of users who would be visiting a given restaurant in future based on the current graph structure. We do this through a multi-step procedure -

1. We first localize a restaurant R which currently have a low amount of reviews.

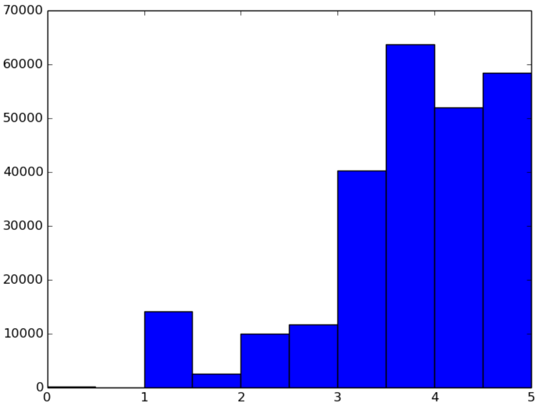

*Figure 4.* Average Star Rating Given by Users

2. We then use SimRank algorithm, as described below, to calculate a similarity score between the restaurant R and other restaurants.

3. We find a set, S, of top-k restaurants based on the similarity scores obtained in step 2.

4. We then find users who have rated S with positive significance (higher than their average rating), and add them to the predicted set.

The main reason to use SimRank as a similarity measure, instead of more general measures, is because of its ability to exploit the whole graph structure for finding the similarity score between 2 given nodes. As described by the authors in (Jeh & Widom, 2002), the SimRank algorithm is not limited only to the nodes with sufficient degree, but it is also capable of finding fairly accurate similarity scores for nodes with relatively lower degree. Herein we describe in detail our algorithm.

## 4.1. Weighted SimRank

SimRank as it is presented initially, does not support the use of weighted edges. For our network, if we consider only unweighted edges (whether a user has rated a restaurant), this gives us information on whether two users have *rated similar restaurants.* However, the information we are really searching for is whether two users have *rated restaurants similarly*, i.e. have similar tastes. In order to model this slightly more nuanced feature, we construct a weighted formulation of SimRank:

$$s(A,B) = \frac{C\sum_{i=1}^{|O(A)|}\sum_{j=1}^{|O(B)|} w_{(A,O_i(A))} w_{(B,O_j(B))} s(O_i(A),O_j(B))}{\sum_{k=1}^{O(A)} w_{(A,O_k(A))} \sum_{k=1}^{O(B)} w_{(B,O_k(B))}}$$

$$s(a,b) = \frac{C\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} w_{(a,I_i(a))} w_{(b,I_j(b))} s(I_i(a),I_j(b))}{\sum_{k=1}^{I(a)} w_{(I_k(a),a)} \sum_{k=1}^{I(b)} w_{(I_k(b),b)}} \tag{1}$$

Where A, B are users; a, b are restaurants, $O_i(A)$ denotes the $i_{th}$ out-edge from a user A and , $I_i(a)$ denotes the $i_{th}$ in-edge (a rating) of a restaurant a. 'w' is a function that gets the weight of an edge. We are using the star rating, normalized to (0,1], given by a user to a restaurant as the weight of the edge between them. In essence, unweighted version of SimRank defines similar users as users who have rated similar restaurants, with each restaurant weighted with equal importance. Our formulation essentially generalizes SimRank by defining weights for how strongly a certain restaurant is correlated with a user. It models the fact that how similar are 2 users in terms of their rating pattern.

### 4.1.1. Existence and Uniqueness of Scores

As mentioned in (Jeh & Widom, 2002), the basic properties that need to be satisfied by the simrank scores to converge and to be unique is that for every a, b, the sequence $S_k(a,b)$ is bounded and nondecreasing as k increases without bound. Here, $S_k(a,b)$ is the value of the simrank score between $a$ and $b$ for $k^{th}$ iteration. It was mentioned by the authors that the original(non-weighted) version of the simrank scores satisfy these properties. Let $S_{uw}$ be the unweighted simrank and $S_w$ be the weighted simrank. So, if $0 \geq S_{uw} \leq 1$ and the weights which we multiply to $S_uw$ for finding the weighted version, $S_w$ vary from 0 to 1, we can say that $0 \geq S_w \leq 1$. This shows that if we have the unweighted version converging to a unique similarity score, then the weighted version would also converge to a unique score. Thus it can be argued that our weighted formulation of SimRank would give us a unique and converging score between 2 nodes.

## 4.2. SimRank Optimization

Using the above mentioned formulation, we implemented a weighted-bipartite SimRank calculation. However, upon running the algorithm on the whole dataset, we quickly determined that it would take approximately 67 days to complete - slightly too long. This is due to the fact that the basic SimRank algorithm calculates the similarity between all pairs. So for a graph of size around 250k user nodes and 47k business nodes, it would be very slow to use a naive approach of calculating simrank. In an attempt to speed up the algorithm, we noted the fact that SimRank score is inversely related to the distance between nodes, so we limited the calculation to 5-hops. The computation was still extremely unweildy, so we decided to explore approximations and other alternatives.

The paper (Kusumoto et al., 2014) outlined in section 2.3 describes an algorithm published in June this year to accurately approximate SimRank score. While the implementation yielded a massive speedup (single-source top-k similarity queries for a node in under a minute), there are certain modifications that needed to be made.

In particular, (Kusumoto et al., 2014) do not describe a method to include edge weights for computation. The calculation of similarity score is based on a monte-carlo method of random walks. SimRank similarity is approximated by running random walks starting at two different nodes, and finding the overlap over multiple trials. Each random step chooses a neighboring node uniformly at random. In order to construct a weighted formulation, we modify the random step procedure by defining a probability distribution over edges proportional to the edge weights protruding from a node. Instead of a uniformly random step, we sample an edge from this defined probability distribution. This essentially ensures that if a user has given a higher rating to a restaurant, then he would be more probable to visit that restaurant as compared to another restaurant to which he has given a lower star rating.

Secondly, the algorithms indexing phase defines all potential nodes that can be similar to a certain source. However, this includes nodes that can be in the same class of the bipartite graph. Since we want to limit calculating only business-business or user-user similarity, we modified the indexing phase to only consider nodes in an even amount of edge hops from a node (thereby staying within one bipartite class). Finally, the greatest limitation of the algorithm, is that it only defines a relative SimRank score per source node. This means that the similarity score of each node will be proportional to the true similarity score, but the constant of proportionality between each source is different. Thus, it is impossible to compare the similarity between two disjoint user-restaurant pairs. Approximate scores are generally very low than the actual scores but, for a given node, we still are able to compute the similarity score based ranking of the nodes as the rankings are scaled down by an unknown yet constant factor. This is the reason why we chose to predict the potential set of customers for a given restaurant based on the similarity score based ranking instead of using the sscore values to actually predict the ratings of the user-restaurant pair.

## 5. Evaluation and Results

### 5.1. Evaluation Criteria

To evaluate our model, we first created a trimmed version of the graph by excluding reviews made after May 31 , 2013 and used this graph as our training set. We collected a set of around 30 businesses from our dataset as a set of emerging restaurants, We say a restaurant to be emerging if it had more than 60 reviews with a rating score of 4 or higher in the original full graph, and had between 10 and 20 reviews in the training set or the trimmed graph. For each business in this set, $B_i$, we found the potential set of users who are probable to visit the restaurant using the algorithm described in previous section. We got the original set of users who actually rated the restaurant highly from the untrimmed data/origial graph. We then computed the precision and recall of our algorithm from the set of predicted users that we generated.

In addition, we evaluated the accuracy of our model against commonly used models in practice: an alternating least-squares collaborative filtering algorithm and a Jaccard model. For the collaborative filtering method, we predicted the ratings every user in the trimmed graph would give each of the businesses in the emerging business set. For a given restaurant, $B_i$, in the emerging set, the potential set of customers are the users that have a predicted rating with positive significance (greater than 4) for $B_i$ and had not already rated the business. For the Jaccard model, we used the Jaccard similarity as a measure of similarity to find the top $k$ similar businesses of business $B_i$ and created a prediction set from the users that rated these businesses highly.

### 5.2. Results and Analysis

From various tests of these three models, we found that our model performed similarly, but better than the collaborative filtering algorithm, while the Jaccard model performed quite differently. Figures 5 and 6, shows the precision and recall values respectively for the set of emerging restaurants based on the 3 different evaluation algorithms. Each box in boxplot 5 corresponds to an evaluation algorithm (Simrank, Jaccard or Collaborative Filtering) and shows the variation of the precision values calculated over the different emerging restaurants. Similarly we have the distribution of recall values in plot 6. Both our model and the collaborative filtering model have a high recall and very low precision, while the Jaccard model has a very high precision with a very low recall. We experimented with different parameters in the SimRank algorithm, such as length of random walks, as well as different thresholds on the minimum rating in the trimmed graph. We also experimented with the number of most similar restaurants considered for predicting the set of users but all produced very similar results.

One of the reason for getting a low precision with the proposed algorithm is that we are predicting a very large potential set of users, which is an order of magnitude higher than the actual set of users who visited the restaurant. One possible way to improve this is to find some metric to rank the users and then trim the potential set based on the metric. Intuitively, status of user seems to be a potential candidate of this metric. To further test this intuition we performed experiments on how does the status of initial reviewers affect the success of a restaurant.

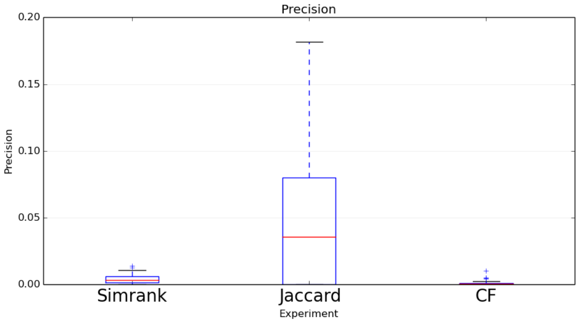

*Figure 5.* Precision For Each Model

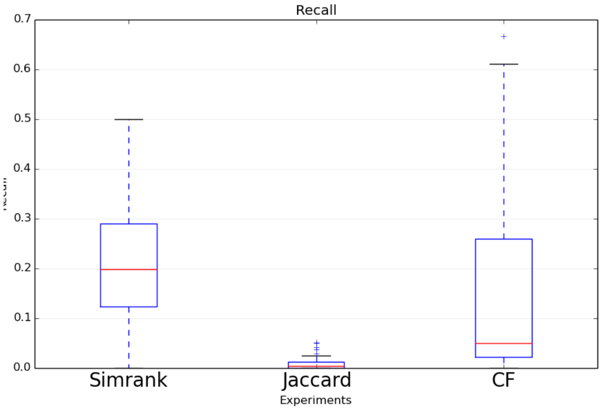

*Figure 6.* Recall For Each Model

We performed analysis on the status of each user to determine if there was any correlation with that and predictive power of an emerging restaurant. To do so, we defined the status of a user as a vector consisting of several features (total number of reviews, number of "helpful" upvotes on reviews, number of "funny" upvotes on reviews, etc.). We partitioned the resturants into 2 sets, emerging and non-emerging, based on their ratings. If there is a significant increase in the number of ratings in the trimmed and original version of the Yelp graph, then the restaurant is said to be emerging, else it is non-emerging. In order to delineate if status of a user had any impact on predictive power, we analyzed the distribution of high status users who rated emerging businesses highly (rating $\geq 4$) in the initial stages of the restaurant and the high status users who gave high ratings to non-emerging businesses initially (See figures 7-9).

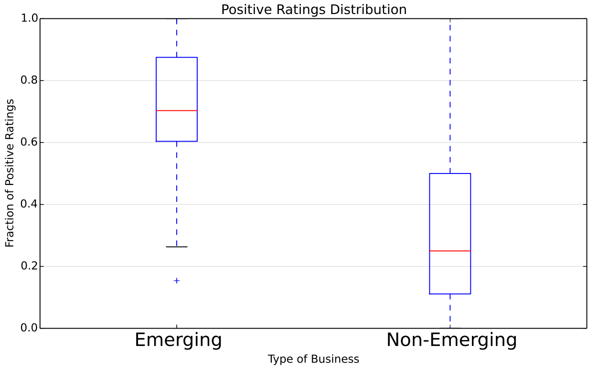

*Figure 7.* Fraction of Positive Users Versus Business Type

Figure 7 shows that out of the few initial ratings, the fraction of positive ratings(4-5 stars) is much higher for emerging restaurants than for non-emerging ones. This shows that there is a correlation between high average rating and emerging businesses. We broke these positive ratings on the basis of the status of the users who gave these ratings. We used 2 different metrics for defining the status: Total number of reviews given by the user and Total number of useful counts on the reviews given by the user.

It can be seen in Figures 8 and 9 that the status of a user is also indicative of whether or not a restaurant will become successful or not, as emerging businesses have more high status users rating them early on than non emerging businesses do. Both ways of measuring status, either through total number of reviews or number of useful votes, emphasize this result.

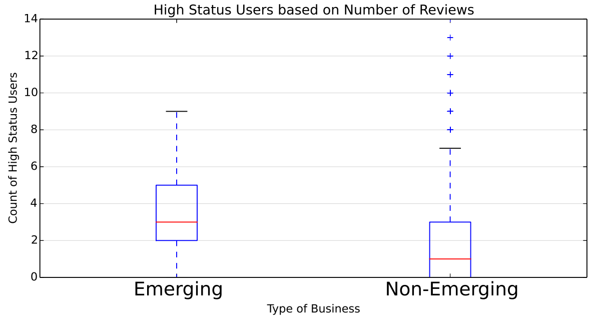

*Figure 8.* Number of High Status Users (Review Count)

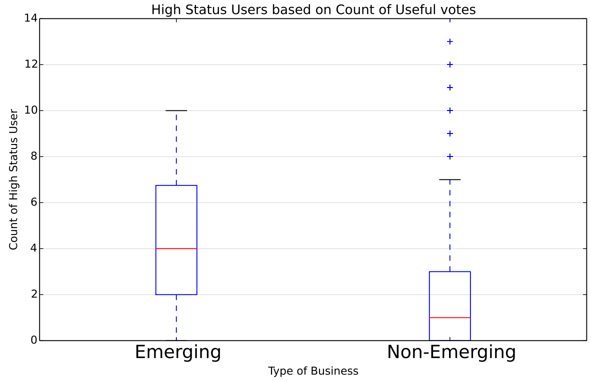

*Figure 9.* Number of High Status Users (Useful Votes Count)

## 6. Discussion and Conclusion

From the results, we can conclude that high status users, both when measured in terms of total review count and number of useful reviews, are markers for emerging businesses. It can also be seen that our model is inaccurate when predicting which users will eventually rate emerging restaurants, but does not perform worse than the commonly used collaborative filtering method.

Although results were not optimal, there are several takeaways for future work here. The following are a list of possible methods to improve perforance.

- Incorporate the status score metrics. As shown in the status data above, there was a clear correlation between user status score, and how influential their review is on predicting a popular restaurant. If a future model could add a weight onto each users status in order to calculate similarity (i.e. a status-based SimRank score), this would potentially yield more accurate results in finding the predicted rating for a particular user on the given business. We would be considering total number of reviews and total useful votes on the reviews of a user as a measure of his/her status. Given that our initial analysis shows that initial reviewers of a new business may play a role in the restaurant's success, including the status of a user in predicting their rating for a given business may provide some interesting insights. We plan to weight the ratings of the initial reviewers on the basis of their status and use this modified rating for further prediction. This essentially enables us to see how beneficial it is for a restaurant to have an influential user, that is a user with a high status score, give a positive review early on.

- Another option could be to further decompose user taste preferences by measuring similarity over each category. I.e. Alice and Bob may have the same type of taste preference for Italian food (like the same type of Italian restaurants similarly); however, Bob may also like Indian food while Alice may not. Instead of modeling a single similarity score through SimRank, we could decompose similarity into a set of similarity scores, one for each category.

- Furthermore, although there is a rich amount of information in textual reviews (McAuley & Leskovec, 2013), we only utilize the quantifiable ratings. Much research has been done recently to extract information from natural language in reviews. Adding these features could definitely have predictive power on emerging businesses.

## References

Anderson, Ashton, Huttenlocher, Daniel, Kleinberg, Jon, and Leskovec, Jure. Effects of user similarity in social media. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 703–712. ACM, 2012.

Jeh, Glen and Widom, Jennifer. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 538–543. ACM, 2002.

Kusumoto, Mitsuru, Maehara, Takanori, and Kawarabayashi, Ken-ichi. Scalable similarity search for simrank. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 325–336. ACM, 2014.

McAuley, Julian and Leskovec, Jure. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pp. 165–172, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2409-0. doi: 10.1145/2507157.2507163. URL http://doi.acm.org/10.1145/2507157.2507163.