Predicting Yelp Reviews

Joseph Chang
chang100@stanford.edu

November 2017

1 Abstract

Yelp is a multinational company which publishes reviews about local businesses. In addition to business reviews, Yelp also has a social network in which users can befriend each other. This rich and unique network provides an excellent opportunity to apply network analysis techniques to solve real world problems. For this project, we attempt to predict the review a user gives to a business by analyzing the Yelp network. Recommender systems are an important to Yelp as many users utilize Yelp for business recommendations.

2 Prior Work

2.1 Effects of User Similarity in Social Media [1]

In this paper, Anderson, Huttenlocher, Kleinberg, and Leskovec study how the similarity characteristics of two users can affect the evaluation that one user provides of another. Previous work studies the effect that relative status between users affects the evaluations of the user. The paper experimented on Wikipedia and StackOverflow datasets and found that the more similar two users are, the less effect the difference in status has on the evaluation. The authors suggest that users rely on status only in the absence of similarity.

2.2 Matrix Factorization Techniques for Recommender Systems [2]

In this paper, Koren, Bell, and Volinsky illustrate how matrix factorization techniques can be used to create successful recommender systems. The authors cite the results of the Netflix Prize competition indicate the superiority of matrix factorization models over nearest-neighbor techniques in producing product recommendations. The recommender system strategies mentioned in this paper are content filtering and collaborative filtering. Content filtering creates representations for the user or products which allows the recommendation system to recommend products to the users. Collaborative filtering incorporates the relationships between users to create recommendations. Collaborative filtering is generally more accurate compared to content filtering but suffers from the cold start problem.

Koren et al. focus on two types of collaborative filtering: neighborhood methods and latent factor models. Neighbor methods are based on the assumption that similar products will get similar reviews. Latent factor models attempt to explain ratings by characterizing items and users based on 20 to 100 factors inferred from ratings patterns. Latent factor models are based on matrix factorization. The most simple matrix factorization model maps users and items to a joint latent factor space in which ”user-item interactions are modeled as inner products in the factor space”. If each item has a representation q_i and each user has a representation p_u, the user’s rating of the item is $q_i^\top p_u$.

2.3 Evaluating Collaborative Filtering Systems [3]

In this paper, Herlocker, Konstan, Terveen, and Riedl assess key properties of collaborative filtering recommender systems. The authors emphasize the importance of keeping the goal/task of the recommender system in mind when evaluating the recommender system. For example, two major tasks are annotation in context (filtering through discussions to decide which ones are worth reading) and finding good items (giving users a ranked list of recommended items).

2.4 Critique

These papers layer a foundation for us to build upon. The first paper, Effects of User Similarity in Social Media, shows the importance of similarity between users relative to the status difference between users. While the majority of our work uses similarity, we also incorporate the idea of using status when similarity is low to weight the review. The original paper focuses on the reviews one user gives to another. We expand on this idea in a collaborative filtering context by weighting using similarity score when similarity is high, but weight using status when similarity is low. In the second paper, Matrix Factorization Techniques for Recommender Systems, Koren et al. give an excellent overview of content-based filtering, collaborative filtering, and latent factor models. We incorporate the basic ideas of content-based and collaborative filtering in our approach. The final paper, Evaluating Collaborative Filtering Systems lists key properties of collaborative filtering recommender systems to keep in mind when training and evaluating recommender systems. We extensively build on these ideas by utilizing content-based recommendation systems, collaborative filtering recommendation systems, and status based recommendation systems.

3 Dataset

For this project, we utilize data for Yelp Dataset Challenge [4] which contains 6 datasets: Users, Businesses, Reviews, Tips, Check-ins, and Photos. There are 4,700,000 reviews, 156,000 businesses, and 1,200,000 users in the dataset.

3.1 Yelp User Network

Our project centers around the user friendship network on the yelp dataset. The user friendship network is an undirected Graph in which the nodes represent the users and the edges represent the existence of friendship between users.

Number of Nodes	1183362
Number of Edges	4402329
Alpha of Power Law	1.62

Table 1: User Network Statistics

Of the 1183362 nodes, approximately 655749 have no edges. Of the remaining nodes with edges, the vast majority of them are part of a large strongly connected component. This strongly connected component consists of 514709 nodes which makes up approximately 43.5% of the entire graph.

The plot below illustrates the degree distribution of the user network. From this plot, we find that the degree distribution follows a power law quite well. This makes sense as the yelp user network is a social network and exhibits preferential attachment behavior. Using the maximum likelihood estimate method, we find that the alpha value of the power law is 1.62.

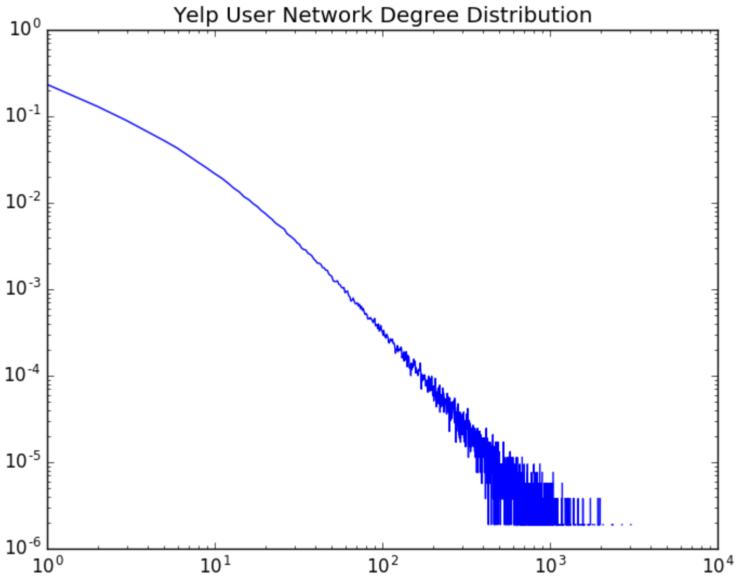


Figure 1: User Degree Distribution

3.2 Cleveland Dataset

For some of our methods, operating upon the entire Yelp dataset is computationally infeasible. For example, generating embeddings for over 1 million users via node2vec is not feasible. Hence, we investigate a subnetwork of the entire Yelp dataset: all users and businesses in Cleveland. There are 2979 businesses and 31946 Yelp users in Cleveland.

4 Methods

4.1 Introduction to Collaborative Filtering

Collaborative filtering models are based on the assumption that similar users will give the same business similar ratings. Thus, we propose the following prediction system.

$$\hat{r}(u_i, b_j) = \frac{\sum_{u \in U} S(u_i, u) \times r(u, b_j)}{\sum_{u \in U} S(u_i, u)} \tag{1}$$

where U is the set of users, S is a similarity scoring function where $S(u_i, u)$ denotes the similarity between users u_i and u, and $r(u, b_j)$ denotes the rating user u gives business b_j, and $\hat{r}(u_i, b_j)$ is our model's prediction of what user i gives business j.

4.2 Friendship Similarity

We propose a baseline similarity scoring function based on friendship. The simplest version is to set the similarity of two users to be 1 if they are friends on yelp and 0 if they are not friends.

4.3 Jaccard Similarity

The Jaccard Similarity coefficient is a statistic for measuring the similarity of sets. In this case, we utilize the similarity of the sets of friends two users have to determine the similarity of two users. In other words, if user u_i has friends F_i and user u_j has friends F_j,

$$S(u_i, u_j) = \frac{|F_i \cap F_j|}{|F_i \cup F_j|} \tag{2}$$

4.4 Pagerank

We stray from our previous collaborative filtering approaches by utilizing a status based approach via Pagerank.

Pagerank was originally designed to rank pages on the web. Good webpages are endorsed (linked to) by other good webpages [9]. Formally, the Pagerank score of node j, r_j, is:

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

We compute the Pagerank scores for all nodes in the user-business graph. If two users are friends, there is an edge between the two nodes. If a user rates a business, an edge will exist between the user node and the business node. This means that a user with a lot of friends and/or a lot of reviews will have a high Pagerank score. This user is also likely an expert Yelp user and thus deserves the high Pagerank score.

For our prediction, we predict the score a user gives to a business as a weighted sum that other users give to the business, weighted by the Pagerank score of the other user. This means that the rating an "expert" gives to a business is weighted more relative to nonexperts. Note that this is not a collaborative filtering approach.

4.5 Personalized Pagerank Similarity

As demonstrated by Leskovec [10], personalized Pagerank can be utilized to compute the similarities between two nodes. For personalized Pagerank, we perform a random walk, but at each step, with some probability $(1 - \beta)$, we teleport back to our original node. The similarity between the original node and any other node is proportional to the number of times we visit the second node during our random walk.

Formally, to compute the similarity between a user j to user s, we have

$$\begin{aligned} r_j &= \sum_{i \to j} \beta \frac{r_i}{d_i} \\ r_s &= \sum_{i \to s} \beta \frac{r_i}{d_i} + (1 - \beta) \end{aligned}$$

4.6 Representation Learning with Node2vec

For this method, we want to create low dimensional representations for nodes in a graph. We hope that a distributed representation for each node will preserve more information than our previous methods. To generate low dimensional representations, we utilize node2vec [6]. Node2vec utilizes stochastic gradient descent to maximize the following quantity:

$$\sum_{u \in V} \sum_{n \in N_s(u)} \log \frac{\exp(f(n) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))} \tag{3}$$

where f is the encoder function which maps nodes to their embeddings and $N_s(u)$ denotes the set of nodes that are in the neighborhood of u.

After training our embeddings using node2vec, compute the similarity between nodes u and v by the cosine similarity of $f(u)$ and $f(v)$. We utilize these embeddings in two ways.

4.6.1 Collaborative Filtering with node2vec

Similar to previous collaborative filtering approaches, we compute the similarity between two users by the cosine similarity between the embedding vectors of the two users.

4.6.2 Content-based Filtering with node2vec

We base this method off the assumption that similar businesses will have similar embeddings. To predict the rating a user, u, gives to a business, b, we compute the similarity between the business b and all other businesses that the user has reviewed. We then output a weighted sum of scores the user has given to other businesses where the weights are proportional to the similarity scores.

4.7 Combining Similarity and Status

In Effects of User Similarity in Social Media [4], the authors found that "we find that evaluations are less status-driven when users are more similar to each other". As such, we attempt to combine both similarity and status when assigning the weights to the reviews other users given to the target business. For the similarity score, we utilize the node2vec approach mentioned in the previous section as that provides the best results. For the status score, we utilize the Pagerank scoring system mentioned previously. If the similarity score is high, we weight the review according to its similarity. When the similarity is low, we weight the review according to the status of the user.

5 Results and Analysis

5.1 Evaluation Metrics

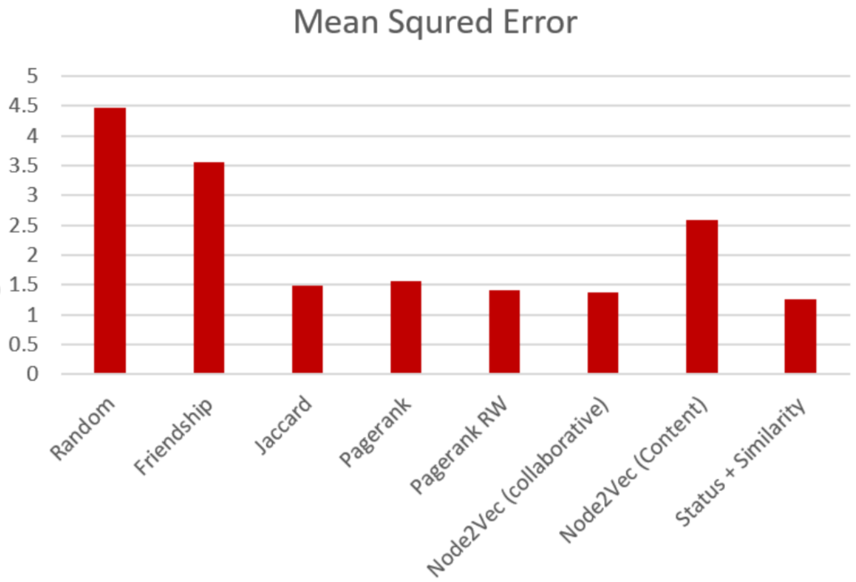
The yelp dataset allows the users to rate businesses on a five star scale. We divide our dataset of almost 5 million reviews into a training and validation set. The training set has 80% of the data and the validation set has 20% of the data. We utilize a couple metrics for evaluating our predictions such as mean squared error, mean error, and standard deviation. In particular, if the ratings are normally distributed, the mean squared error is proportional to the negative log probability of our predicted rating. This conforms with basic tenets of information theory in which the self-information of an event is the negative log probability. If we have rating predictions $\hat{y}$, ground truth predictions y, and n evaluation examples,

$$\texttt{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}-y)^2 \tag{4}$$

$$\texttt{Mean Error} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}-y| \tag{5}$$

5.2 Results

Model	MSE	ME	STD
Random	4.46	1.69	1.96
Friend-Baseline	3.56	1.44	1.82
Jaccard	1.48	0.95	1.22
Pagerank	1.57	0.97	1.25
Pagerank Similarity	1.41	0.91	1.18
Node2vec (Collaborative)	1.37	0.92	1.17
Node2vec (Content)	2.59	1.19	1.6
Combining Status and Similarity	1.26	0.89	1.13



5.3 Model Analysis

We run our models over the Cleveland dataset. As a reference value, we compute the mean squared error, the mean error, and standard deviation for a model in which we randomly output a rating between 1 (the minimum review score) and 5 (the maximum review score). The random model has a mean squared error of 4.46, a mean error of 1.69, and a standard deviation of 1.96.

Our first model, the simple friend baseline, is a moderate improvement over the random model with a mean squared error of 3.56. It is apparent that this model is much too simple to be a good collaborative filtering model. A major flaw with this model is that for an isolationist user (a user with relatively few friends), if none of the user's friends review the target business, it is impossible for the model to make a good prediction for the target business. This is extremely important as new users joining Yelp will have zero of very few friends, but good recommendations must be made for these users in order to retain them.

The next model, the Jaccard similarity model, shows significant improvements over our previous models with a mean squared error of 1.48. The mean squared error is approximately one third of the mean squared error of the random model. Additionally, it is important to note that this model is computationally easier compared to our higher performance models. The Jaccard similarity model alleviates, but does not completely dodge the "isolationist user" problem brought up previously in the baseline.

The Pagerank model strays from our previous collaborative filtering models. The performance of this model is slightly worse compared to the Jaccard model with a mean squared error of 1.57. This confirms the belief that recommender systems are extremely personal and are highly dependent on the user in question. Because Pagerank only computes the status of users in the network and predictions are made based on the status of other users, this model does not perform very well.

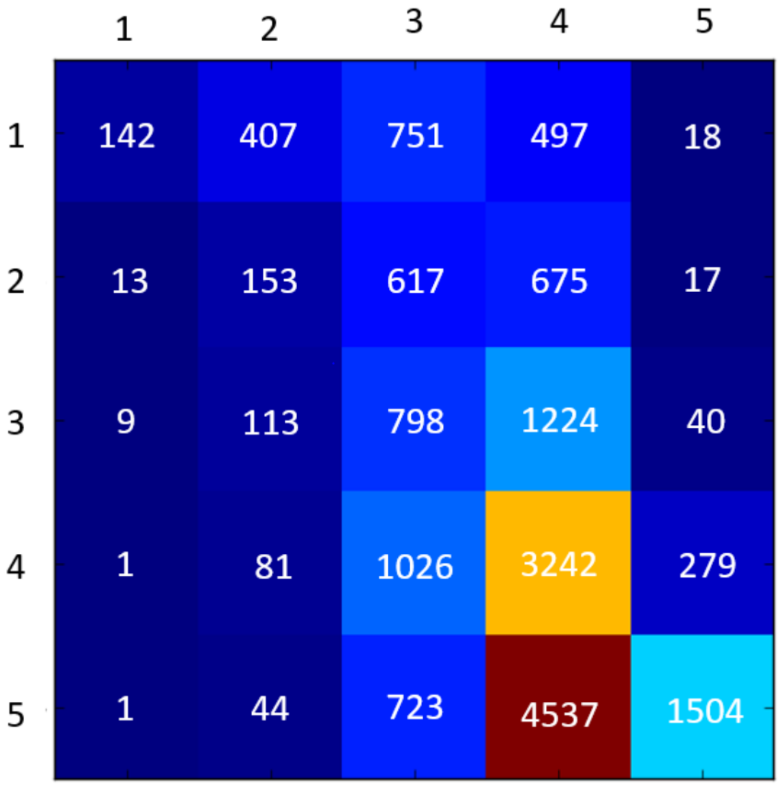
The Personalized Pagerank Random Walk model produces better results compared to both the Jaccard and Pagerank models with a mean squared error of 1.41. This model, compared to the Jaccard model, does an even better job of circumventing the isolationist user problem. As long as the user has at least one friend, the personalized pagerank algorithm can compute reasonable similarity scores. The improvements between this model and the original Pagerank model further provides evidence that reccomender systems are extremely personal and depend heavily on the user in question.

The collaborative filtering with node2vec model demonstrates small gains compared to the personalized pagerank random walk model. The model produces a mean squared error of 1.37. The node2vec model, similar to the personalized pagerank model circumvents the isolationist user problem as it also utilizes a random walk to find nodes in the neighborhood of the target user. It appears that the cosine similarity between two embeddings is a better similarity metric compared to our other models. This confirms our belief that low dimensional representations preserve more information than the metrics we use in previous models. Unfortunately, this model is computationally expensive as we must generate embeddings for each user.

The content filtering with node2vec model, with a mean squared error of 2.59, performs much worse compared to our more advanced collaborative filtering models. It is important to note that the embeddings generated via node2vec for businesses may not be very good as no businesses ever share an edge in the user business network. Hence, the only way a business appears in another business' "neighbor set" is if that node is reached during a random walk. This assumption appears to be faulty and the similarity scores between two businesses may not be entirely accurate. Note that both node2vec approaches are extremely computationally expensive as embeddings must be generated for every node in the graph.

Our final model, combining status and similarity, produces the best results with a mean squared error of 1.26. This model can completely circumvent the ”low friends” problem by relying on the reviews of expert Yelp users for predictions. Previous models such as jaccard, personalized pagerank, and node2vec still require users to have friends to make predictions, whereas this model does not require users to have any friends. It is possible, and even quite likely, that new Yelp users will have no friends. It is still important to give good predictions for these users so that these users will continue using Yelp. Additionally, the results of this model conform with previous theories about status and similarity.

5.4 Confusion Matrix



	1	2	3	4	5
1	142	407	751	497	18
2	13	153	617	675	17
3	9	113	798	1224	40
4	1	81	1026	3242	279
5	1	44	723	4537	1504

The confusion matrix above is for the model in which we combine status and similarity. The horizontal direction is the model’s prediction and the vertical direction is the ground truth prediction. The biggest mistake our model makes is predicting 4 stars when the review was actually 5 stars. Both of these reviews are considered positive and are hard to differentiate between.

6 Conclusion

In conclusion, we see promising results for our models as the majority of our models far exceed the random model and simple baseline. We explore this problem from a variety of different approaches, including collaborative filtering, content-based filtering, and status. Our results show that collaborative filtering outperforms status and content-based filtering models. By combining status and collaborative filtering, we achieve our best results.

7 Distribution of work

Joseph did all the work.

References

[1] Anderson, Ashton, et al. “Effects of User Similarity in Social Media.” www.cs.cornell.edu/home/kleinber/wsdm12-sim.pdf.

[2] Koren, Yeshuda, et al. ”Matrix Factorization Techniques for Recommender Systems.”

[3] Herlocker, Jonathan, et al. ”Evaluating Collaborative Filtering Recommender Systems.”

[4] Yelp. Yelp Dataset Challenge. https://www.yelp.com/dataset/challenge, 2017.

[5] Hamilton, William, et al. ”Inductive Representation Learning on Large Graphs.”

[6] Grover, Aditya, et al. ”node2vec: Scalable Feature Learning for Networks.”

[7] Schelter, Sebastian, et al. ”AIM3 - Scalable Data Analysis and Data Mining.”

[8] Goodfellow, Ian et al. ”Deep Learning.”

[9] Su, Jessica. ”Pagerank.” web.stanford.edu/class/cs224w/slides/handout-page_rank.pdf

[10] Leskovec, Jure. ”Pagerank.” web.stanford.edu/class/cs224w/slides/14-pagerank.pdf