

Predicting Yelp Ratings Using User Friendship Network Information

Wenqing Yang (wenqing), Yuan Yuan (yuan125), Nan Zhang (nanz)

December 7, 2015

1 Introduction

With the widespread of B2C businesses, many products and service providers need both evaluation and prediction of customers' feedback. For example, Yelp has a five-star quality rating system of restaurants as well as review text, which generates a big volume of explicit and implicit user data. Consequently, a lot of meaningful research questions can be answered using Yelp's datasets.

In this project, we attempt to predict the rating a user will give to a restaurant listed on Yelp using Yelp's Challenge Dataset. Being able to predict the rating a user assigns to a restaurant is helpful when trying to build better recommendation systems on Yelp. We approach the problem from a social network analysis perspective by incorporating Yelp user-user friendship networks in our predictions, and we attempt to test if the additional network information enhances the accuracy of the rating predictions.

2 Literature Review

With the vast amount of information on products and businesses available to users online nowadays, there is increasing interest in developing recommender systems that provide users with personalized recommendations on items. Usually these systems work by predicting numeric ratings users give products or businesses, and in general they belong to one of two types: content-based methods or collaborative filtering based methods.

Content-based methods compare how similar a target item is to items that the user has rated before and gives a predicted rating based on the user's previous ratings. Mooney and Roy determine the similarity between books by mining the text in book descriptions on Amazon.com and then recommend similar books to users [5]. Sarwar, Konstan and Riedl compare different methods of computing item similarity and different methods of producing predictions from the computed similarities [8]. Pazzani and Billsus allow users to provide a profile of webpages that they find "interesting" and then revise this profile by comparing the similarity between text on webpages [6].

On the other hand, collaborative filtering methods rely on the assumption that users similar to each other tend to like the same items or tend to give similar ratings. Koren, Bell and Volinsky, the winners of the Netflix Prize Contest, summarize the application and flexibility of matrix factorization techniques used in recommender systems, and they introduce how to use singular value decomposition (SVD), regularization, stochastic gradient descent and alternating least squares to tackle missing data problems [3]. McAuley and Leskovec use latent factor models to uncover hidden dimensions in review ratings and Latent Dirichlet Allocation to uncover the hidden dimensions in review text [4]. Yu *et al.* develop an algorithm to recommend web communities to users, and they solve the sparsity problem in traditional collaborative filtering methods by generating the latent link between communities and members using latent topic associations [7].

There have also been attempts to improve traditional recommender systems by taking into consideration the social relations among users. He and Chu present a social network-based recommender system (SNRS) that incorporates the influence from both immediate friends and distant friends of a user [1]. They test their recommender system on Yelp’s dataset, and they find that SNRS performs better than other traditional methods. Using users’ contact information on Flickr, Zheng and Bao are able to prove the usefulness of users’ social network structure when recommending Flickr groups to users [10]. Yang *et al.* focus on matching users to Yahoo services using users’ contacts on Yahoo! Pulse [9]. They propose a hybrid model that combines a factor-based random walk model to explain friendship connections and a coupled latent factor model to uncover interest interactions.

Taking inspiration from the previous work done, we use a latent factor model with bias terms as our baseline method for predicting user ratings of restaurants. Since previous studies have shown that user social relations are effective at improving rating predictions, we improve our baseline model by adding users’ friends’ information into the model. Intuitively, it is reasonable to add user-user interaction because people often go to restaurants with friends, so their friends’ preferences will influence their own preferences to some extent. However, not all friends’ opinions are equal, and depending on how friends are involved in the Yelp friendship network their opinions may be thought of as more or less reliable. Taking this fact into consideration, we further weight friends’ ratings by their degree centrality.

3 Data Summary

3.1 Description

The dataset we choose to work with is the Yelp Challenge Dataset. Compiled for researchers and students to explore a wide variety of topics on Yelp, the Challenge Dataset includes 1.6 million reviews and ratings, 481,000 business attributes, a social network of 366,000 users for a total of 2.9 million social edges, and aggregated check-ins over time for each of the 61,000 businesses. The businesses included in the dataset are located in the U.K., Germany, Canada and the U.S. This dataset is particularly suitable for our purposes, since in addition to user ratings of businesses, it also provides information on which users are friends with each other on Yelp. The data is available for download via the Yelp Dataset

Challenge website in the form of .json files (http://www.yelp.com/dataset_challenge).

Number of users	269231
Number of restaurants	21892
Number of reviews	990627
Average review rating	3.0

Table 1: Data Statistics

Since Yelp is best known for its reviews on restaurants, we only explore restaurants in the U.S. and leave out the other business types for our project. After applying these filters, we end up with 21892 businesses that are identified as restaurants and 269231 users that have posted a total of 990627 reviews at these restaurants (Table 1).

3.2 Network Properties and Visualization

To construct the Yelp user-user friendship network, we let each user be a node, and add an undirected edge between two users if they are friends with each other on Yelp. Summary statistics of the Yelp friendship graph are shown in Table 2 and the connected components information is shown in Table 3. From the connected components information, we can see that the connections of the network are very sparse. Approximately 50% of the users do not have friends. This can also be seen from the degree distribution plotted in Figure 1. The degree distribution of nodes is extremely right-skewed, with most nodes having degree less than 120 and 1.06% of nodes having degree more than 120. It approximately follows a power-law distribution with $\alpha = 1.44$.

Number of nodes	269231
Number of edges	986864
Alpha of power-law	1.44

Table 2: Graph Statistics

Size of CCs	Number of CCs
1	135648
2	1763
3	162
4	24
5	7
6	2
7	2
129414	1

Table 3: Connected Components (CC) Info

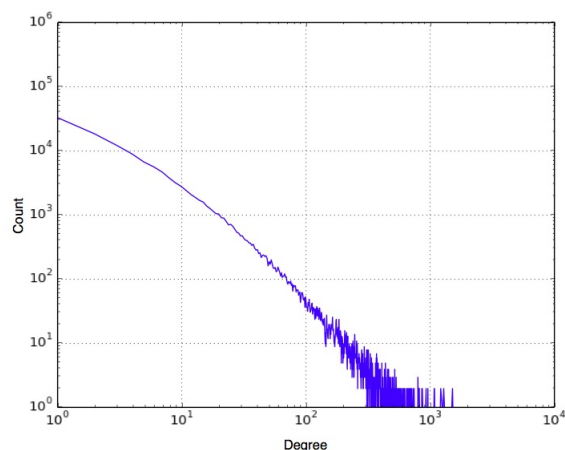


Figure 1: Power Law Distribution of Yelp Dataset

To create the visualization of the network, we filter out nodes with degree more than 120, and take a random sample of 10% of the remaining nodes. We plot the Yelp friendship network using these sampled nodes in Gephi and apply the Force Atlas 2 layout. After looking at user attributes such as the average rating users give, number of reviews posted, number of years being a Yelp user, restaurant locations most reviewed, restaurant categories most reviewed etc., we observe that the network shows clustering pattern by most reviewed restaurant locations. Intuitively, this makes sense. Since people go to restaurants together with friends, we would expect friendship clustering to show separation pattern by location.

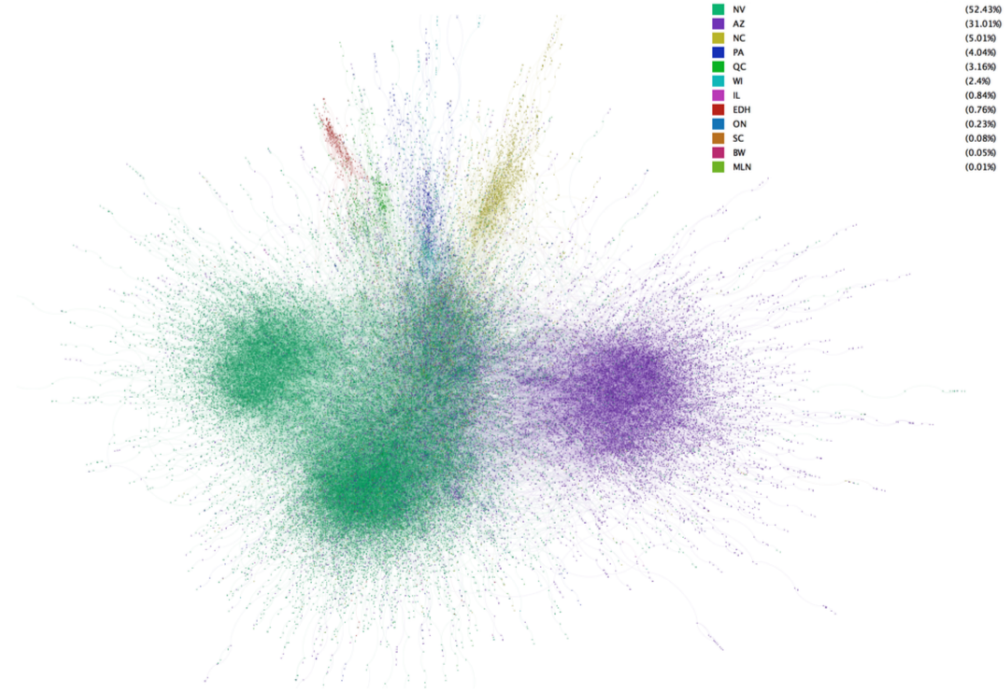


Figure 2: Yelp friendship network with nodes colored by location

4 Baseline Model and Results

4.1 Model

The basic model we use to predict ratings is the “standard” latent-factor model.

$$\hat{r}_{u,i} = \mu + a_u + b_i + q_i^T p_u$$

Here, $\hat{r}_{u,i}$ is the prediction of the rating for item i by user u . μ is a global offset parameter. a_u and b_i are user and item biases respectively. p_u and q_i are user and item factors. The

system learns by minimizing the Error Sum of Squares (SSE) combined with regularization.

$$\min \sum_{(u,i) \in \tau} [(r_{u,i} - \hat{r}_{u,i})^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + a_u^2 + b_i^2)]$$

- Initialization: μ is given by averaging ratings and we will not update it during iterations. a_u and b_i are initialized by averaging ratings and residuals. We would like p_u and q_i for all users u and items i to be such that $q_i^T p_u \in [0, 5]$. So we initialize all elements of p and q to random values in $[0, \sqrt{5/k}]$, where k is the number of latent factors.
- Update: We use stochastic gradient descent to perform updates according to the update equations shown below.

$$\begin{aligned} \epsilon_{u,i} &\leftarrow r_{u,i} - \hat{r}_{u,i} \\ a_u &\leftarrow a_u + \eta(\epsilon_{u,i} - \lambda a_u) \\ b_i &\leftarrow b_i + \eta(\epsilon_{u,i} - \lambda b_i) \\ p_u &\leftarrow p_u + \eta(\epsilon_{u,i} q_i - \lambda p_u) \\ q_i &\leftarrow q_i + \eta(\epsilon_{u,i} p_u - \lambda q_i) \end{aligned} \tag{1}$$

- Parameters: We read each line of the rating file from disk and update the parameters for each line. Each iteration of the algorithm will read the whole file. We set the number of iterations to be 100, and the step size η to be 0.1. We then try out different values for the number of latent factors k and the regularization parameter λ .

4.2 Results

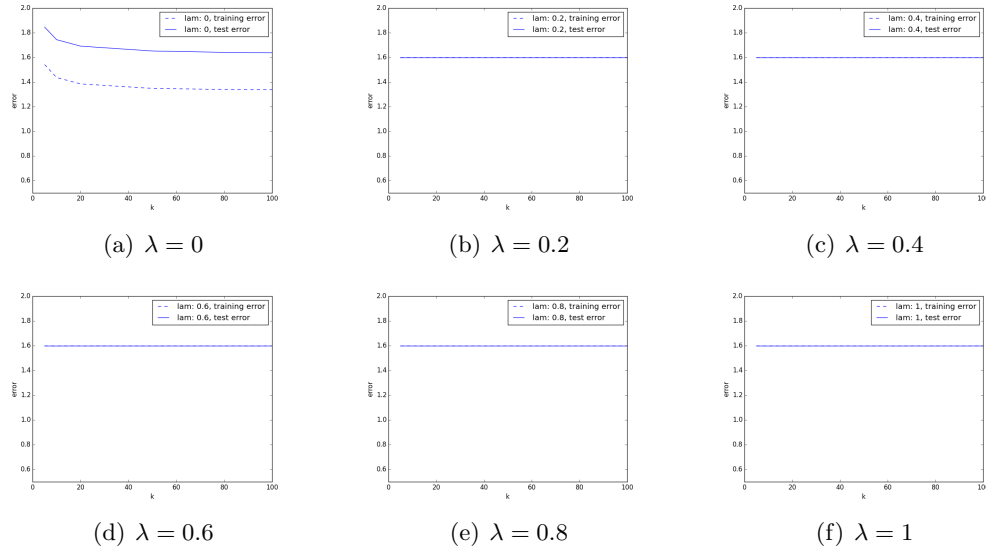


Figure 3: Baseline model training and test MSE for different k 's and λ 's

We randomly split 20% of the reviews into a test set and 80% into a training set, and we investigate the performance of $k \in [5, 10, 20, 50, 80, 100]$ and $\lambda \in [0, 0.2, 0.4, 0.6, 0.8, 1]$ by comparing their mean-squared error (MSE) on the training and test sets. The results are shown in Figure 3. We can see when $\lambda > 0$, the training errors and test errors are nearly the same under different iterations. We believe there may be underfitting problems, and it is reasonable to propose a more complicated model as our improved model. The smallest MSE of 1.5974 is given by $k = 50$ and $\lambda = 0.4$.

5 Improved Model and Results

5.1 Model

For our baseline model, we consider the overall average rating, the bias term for user and the bias term for restaurant. Considering the fact that people will go to restaurants together with friends and they will evaluate the restaurant similarly since they tend to have similar tastes and obtain similar services, we can extract further useful information from the friendship network of users.

Inspired by the SVD++ model in [2], we propose our improved model which takes into account the influence of friendship on users' ratings. A user will demonstrate implicit preference for restaurants that his or her friends have visited and rated. Therefore we add an additional "friends" term to the original free user factor p_u . The estimation of the rating given to restaurant i by user u using our improved model is given as follows:

$$\hat{r}_{u,i} = \mu + a_u + b_i + q_i^T (p_u + |F(u)|^{-0.5} \sum_{j \in F(u)} y_j)$$

Here $F(u)$ represents user u 's friends who have rated restaurant i before. $|F(u)|$ is the size of this set and it works as a normalization constant. For each user, we add an additional k dimensional vector y_j . Thus the user factors are now composed of two parts: one is the free user factor p_u as in the baseline model and the other one is the "friend term" $|F(u)|^{-0.5} \sum_{j \in F(u)} y_j$.

The cost function of the improved model is given by

$$\min \sum_{(u,i) \in \tau} [(r_{u,i} - \hat{r}_{u,i})^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + a_u^2 + b_i^2 + \sum_{j \in F(u)} \|y_j\|^2)]$$

- Initialization: μ is given by averaging ratings and we will not update it during iterations. a_u and b_i are initialized by averaging ratings and residuals. We would like p_u and q_i for all users u and items i to be such that $q_i^T p_u \in [0, 5]$. So we initialize all elements of p and q to random values in $[0, \sqrt{5/k}]$, where k is the number of latent factors. We initialize all elements of y to be 0.
- Update: In each update, we update new values of parameters using the old values. We use stochastic gradient descent to get the update equations as following.

$$\begin{aligned}
\epsilon_{u,i} &\leftarrow r_{u,i} - \hat{r}_{u,i} \\
a_u &\leftarrow a_u + \eta(\epsilon_{u,i} - \lambda a_u) \\
b_i &\leftarrow b_i + \eta(\epsilon_{u,i} - \lambda b_i) \\
p_u &\leftarrow p_u + \eta(\epsilon_{u,i} q_i - \lambda p_u) \\
q_i &\leftarrow q_i + \eta[\epsilon_{u,i}(p_u + |F(u)|^{-0.5} \sum_{j \in F(u)} y_j) - \lambda q_i] \\
\forall y \in F(u) : y_j &\leftarrow y_j + \eta(\epsilon_{u,i} |F(u)|^{-0.5} q_i - \lambda y_j)
\end{aligned} \tag{2}$$

- **Parameters:** We read each line of the rating file from disk and update the parameters for each line. Each iteration of the algorithm will read the whole file. We set the number of iterations to be 100, and the step size η to be 0.1. Similar to before, we then try out different values for the number of latent factors k and the regularization parameter λ .

In addition, we also consider using degree centrality to weight the new user factor y_j . The weighted “friend term” is given by

$$(\sum_{j \in F(u)} D_j)^{-0.5} \sum_{j \in F(u)} D_j y_j$$

Our experiments find that weighting by degree centrality leads to small difference in prediction accuracy compared with the improved model without weighting. Therefore we omit the detailed description of the model with weighting by degree centrality here.

5.2 Results

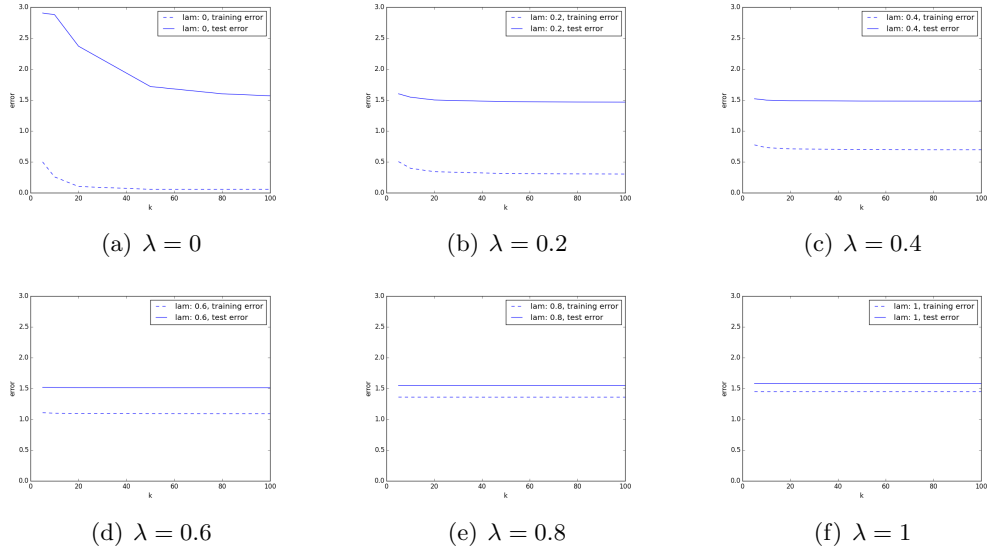


Figure 4: Improved model training and test MSE for different k 's and λ 's

With this improved model, we observe that the friends’ influence term helps to improve the accuracy of rating prediction significantly. Again, we randomly split 20% of the reviews into a test set and 80% into a training set, and we investigate the performance of $k \in [5, 10, 20, 50, 80, 100]$ and $\lambda \in [0, 0.2, 0.4, 0.6, 0.8, 1]$ by comparing their MSE on the training and test sets. The results are shown in Figure 4. We can see when $\lambda = 0$, the training errors are very small but the test errors are large, so there are overfitting problems. With the increase of λ , the gap between training errors and test errors becomes smaller. The smallest MSE of 1.4663 is given by $k = 100$ and $\lambda = 0.2$.

6 Conclusions

By comparing the results of the baseline model and the improved model, we observe that the friend term introduced in user factors improves the prediction accuracy significantly. The free user factor p_u represents explicit ratings given by user u . The friend term represents user’s implicit preference for restaurants. People will have similar tastes with their friends and friends can also recommend or comment on restaurants, which influences ratings given to restaurants by users. The two terms combined together therefore give us more information of a user’s rating behavior. We also try to incorporate the centrality measures in the friend term, but our results show that weighting friends’ ratings by degree centrality does not produce a noticeable improvement on the prediction performance. This result is reasonable for the Yelp Dataset because most of the users do not have friends information and only very few users have a lot friends. Therefore we conclude that friendship network information allows us to predict user restaurant ratings more accurately, although further differentiating between friends through weighting by degree centrality does not offer much improvement in prediction accuracy.

References

- [1] He, J., & Chu, W. W. (2010). A social network-based recommender system (SNRS) (pp. 47-74). Springer US.
- [2] Koren, Y. (2008, August). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 426-434). ACM.
- [3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8), 30-37.
- [4] McAuley, J., & Leskovec, J. (2013, October). Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems (pp. 165-172). ACM.
- [5] Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. In Proceedings of the fifth ACM conference on Digital libraries (pp. 195-204). ACM.

- [6] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3), 313-331.
- [7] Qian, Y., Zhiyong, P., Liang, H., Ming, Y., & Dawen, J. (2012, November). A latent topic based collaborative filtering recommendation algorithm for web communities. In *Web Information Systems and Applications Conference (WISA)*, 2012 Ninth (pp. 241-246). IEEE.
- [8] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.
- [9] Yang, S. H., Long, B., Smola, A., Sadagopan, N., Zheng, Z., & Zha, H. (2011, March). Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web* (pp. 537-546). ACM.
- [10] Zheng, N., & Bao, H. (2013). Flickr group recommendation based on user-generated tags and social relations via topic model. In *Advances in Neural Networks-ISNN 2013* (pp. 514-523). Springer Berlin Heidelberg.