Yelp Business Recommendation and Business Rate Prediction in Bipartite Networks– Final Report

Sidi Chang
schang03@email.wm.edu

May 11, 2016

Abstract

Yelp dataset from RecSys Challenge 2013 is used for prediction of future rates for businesses and recommendation for users. Both prediction and recommendation have been done in this project. In prediction part, random network prediction is used to serve as benchmark as well as regular Jaccard Coefficient. A new algorithm introduced by several Stanford student is implemented and evaluated. In that algorithm, a new parameter Threshold λ is used as the lower bound for similarity. Only if similarity is high enough, the prediction will be accepted. Otherwise prediction will be rejected. This method with appropriate λ can achieve 78.65% accuracy with 0.5 allowable error.

In recommendation part, random recommendation also serves as benchmark. A brand new algorithm prototyped by myself is introduced and implemented. A new weight parameter is considered when calculating score based on similarity. The accuracy of this new algorithm is highly improved compared with Netflix algorithm and random benchmark in both 0 allowable error and 1 allowable error.

1 Introduction

For the last five years, social network has become an important part of our daily life. As smart phones become increasingly popularized, Facebook, Google and Uber have changed our behaviors, as well as Yelp. Matching and predicting novel links has many applications in different areas, such as Ads, recommendation system and network analysis [4] [3]. Yelp Business rating prediction challenge [7] in 2013 gave me great datasets to analyze. In this paper, my motivation, prior related work, data collection and aggregation will be explained. Methodology and algorithms for this research plan will be discussed and concluded clearly and thoughtfully.

2 Motivation

Predicting rates for both business and customers are beneficial. For businesses, they could know their customer's review of their service and future rating, with the future rating, they can make changes to their services in advance. For customers, they will know whether the rate for a specific business is accurate which could help them choose the right business as they want. Yelp Data Challenge [7] gives me an opportunity to conduct research on this field, evaluating on Link Prediction in different areas and recommendation of businesses to each customer.

3 Problem Definition

A previous challenge called RecSys Challenge 2013 which focuses on Yelp business rating prediction [7] is used as my data source. To transform this into an easily analyzable form, definition is as following:

- We assume there exists a bipartite graph: set of users, set of businesses. Because two sets are bipartite, there does not exist any connection(edge) inside each set
- We say that there is an edge between customer i and business j if customer i gave business j a rating before

Then the problem is specified in two parts: Link prediction on a business rating, and business recommendation for users.

Link prediction on a business rating. Given a set of businesses B, a set of users C, we construct a B2C graph $G = (V, E)$ as following: Let V be the set of $|B| + |C|$. For each user i and business j, if customer has ever rated business before, then edge $value(i, j) =$ Star(1 to 5 Stars). For the graph G, we consider the following features: a) The number of customers who have rated to several businesses b) the number of customers who have interacted the same business. The link prediction task is then to predict the rating for each business in the future.

Recommendation Based on the previous graph, we need to make recommendations for each user with his/her highest potential business to go.

4 Prior Related Work

The big data and its techniques is discussed in the work of H.V. Jagadish et al. [3]. In this Review Article, the authors talked about key insights of big data. They states that most researchers only focus on part of creating value from Big Data rather than following all the steps. The methods surveyed show detailed steps on the big data life cycle, from data acquisition to interpretation. The conclusion shows that even we have entered an era of Big data with not only key technical challenges but also social and political challenges that need to be addressed. In my paper, I will follow the steps appeared in this paper to clean and

organize data and try to avoid social and political challenges such as making business name encrypted.

Previous work for a different Yelp Data Challenge Dataset have been done also as a final project in 2015 Stanford CS224W class [4]. Various methods such as preferential attachment, Jaccard Coefficient are used in this paper. The authors address this problem in detail and implemented several different methodology and algorithms. Then the authors also compare empirical data with Test Set and conclude their decision and future work. In my project, I will implement one of his algorithms and conduct analysis of his algorithm performance which he never mentioned. Also, I will compare his algorithm with random benchmark and direct Jaccard Coefficient method.

Above all, in my project the originality has two phases. The first phase is that I would follow some methodology and algorithms of those papers. [3] give me some basic insight of data analysis procedure. [4] give me methodology and algorithms I can use to make prediction. I will combine those ideas and choose the best one in my project. The second phase is that I would follow some paper such as [8] and come up with some new recommendation algorithm.

5 Data Collection and Processing

Yelp Dataset Challenge [7] is used to serve as base data for my project. Data is organized unstructured as JSON format.

5.1 Dataset Information

The Yelp data contains two main datasets: Training set and Test set and their elements are shown in Table [1].

Table 1: Elements in Training and Test set

	Training set	Test set
Businesses	11,537	1,205
Checkin sets	8,282	734
Users	43,873	5,105
Reviews	229,907	22,956

5.2 Business and user rate information

1. Yelp Data State Distribution
   99.97% of the businesses are located in Arizona(AZ) state from the dataset, so I cleaned

data outside the AZ state and all my analysis are only based on Arizona state businesses data.

2. Business Review count distribution
   It is very important to know whether most of the businesses are under reviewing or over reviewed. For example, if a business has been reviewed thousands of time, there is with high probability that a significant change of review star will not happen, which means the review have already been convinced. On the other hand, there is a strong need by users to know the future's rates for businesses with very few reviews. From Figure [1], we could guess the number of business rates follow power law, detailed analysis of this distribution will be shown in Section [7]. Similarly, the more a user reviews, the more his/her future review will be convincing. The number of users versus number of user reviews is shown in Figure [2]. Further analysis will also be done in Section [7].

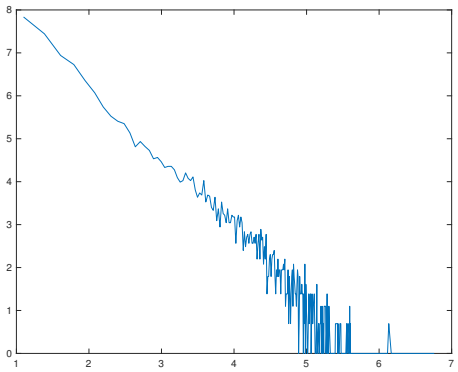


Figure 1: log-log distribution of number of businesses(y-axis) versus number of review times(x-axis)

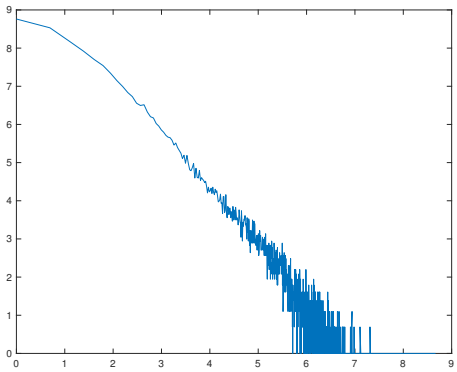


Figure 2: log-log distribution of number of users(y-axis) versus number of review times(x-axis)

6 Methodology and Algorithms for research

The paper proposed by Kevin (Junhui) Mao, Xiao Cai and Ya Wang [4] discussed prediction methodology and algorithms clearly and thoughtfully. Their algorithm will be implemented in my project and further analysis which they have never done will be presented. A new recommendation algorithm prototyped by myself will also be introduced and analyzed. Random network algorithms and several other popular algorithms talked in class will be used to serve as benchmark:

- scores based on proximity metrics using node neighbor features (Both prediction and recommendation)

- scores based on random network that serves as benchmark

Node Neighborhood Definition In a user-business bipartite network, if two users review the same business, we assume they are similar to each other. The more intersection they have, the closer they are.

- $\mathcal{N}(\mu)$:= set of users who are co-reviewers of a given user μ for a business b
- $\mathcal{N}(b)$:= set of users who reviewed business b
- $N(\mu)$:= number of reviews counted by user μ

Both sets only contain users data which will make algorithm computation easier.
Several existing algorithms will be applied and compared for each candidate new edge, The following algorithms would be considered:

- Random Benchmark
  This algorithm will simply create random edges between users and businesses, the number of users, businesses, total edges will be the same, and it will serve as a benchmark for evaluating other algorithms.

- Jaccard Coefficient [6]
$$score(\mu, b) = \frac{|\mathcal{N}(\mu) \cap \mathcal{N}(b)|}{|\mathcal{N}(\mu) \cup \mathcal{N}(b)|}$$
  Jaccard Coefficient is normalized into $[0, 1]$ and can measure the strength of relationship between two nodes when the existence of common neighbors is simply owing to the existence of lots of neighbors for each node. This methodology can be used in both prediction and recommendation. In prediction part, the rate for one business will be close to another business when users who rate them have a large Jaccard Coefficient. In matching, the system will recommend users to go to businesses where users with large Jaccard Coefficient usually go to.

- Preferential Attachment [6]
$$score(\mu, b) = |\mathcal{N}(\mu)| \cdot |\mathcal{N}(b)|$$
  Assume the chance that μ will review b is proportional to the number of other users who have already reviewed b, and it's correlated with the number of users who have similar behavior as μ.

6.1 Link Prediction

Two methods will be used to do link prediction: Random Benchmark and a method called **minimum-common-neighbors** introduced by Kevin (Junhui) Mao, Xiao Cai and Ya Wang [4]. [4] algorithm is shown below in Algorithm [6.1]. Random network prediction will be served as the benchmark to evaluate performance of this new algorithm.

Algorithm 1 Minimum common neighbors

Threshold(λ) is pre-set and adjusted based on performance, and for each user business pair (μ, b) :

(a) if $(\mu, b) \in$ edge,

(b) compute score (μ, b) : using Jaccard Coefficient or Preferential Attachment

(c) if $\frac{|\mathcal{N}(\mu) \cap \mathcal{N}(b)|}{|\mathcal{N}(\mu) \cup \mathcal{N}(b)|} < \lambda$: continue

(d) output $(\mu, b, score(\mu, b))$

Sort the (μ, b) pairs by decreasing order of score (μ, b)
Predict the top pairs with highest score (μ, b) as new nodes

In Algorithm [6.1], the parameter λ is defined as a lower bound for similarity, if the similarity is low which means for a specific business, none of the two users are similar to each other, then it makes no sense to predict through those users. In this situation, we will not allow prediction to happen. Once the similarity is higher than λ, we will update prediction based on those nodes.

6.2 Link Matching

Similarly, I will set a random network recommendation as the benchmark to evaluate the algorithm performance. Then I will apply Netflix recommendation algorithm talked in class to this problem. Also, I will introduce my improved methods in Algorithm [6.2]. This

Algorithm 2 Weighted common neighbors

For user μ_1 and business b,

(a) if (μ_1, μ_2) has common edges to b,

(b) compute similarity (μ_1, μ_2) : using Jaccard Coefficient

(c) update $score'(\mu_1, \mu_2)$ by $similarity(\mu_1, \mu_2) \times N(\mu)^{1/4}$

(d) output $(\mu_1, \mu_2, score'(\mu_1, \mu_2))$

Sort the (μ_1, μ_2) pairs by decreasing order of $score'(\mu_1, \mu_2)$
Predict the top pairs with highest $score'(\mu_1, \mu_2)$ and choose top rate in μ_2 to recommend

algorithm is a brand new algorithm created by myself. This algorithm insights from recommendation letter in academic application. Mostly, admission committee tend to believe or put higher weights recommendation letters written by well-known professors with great reputation in this area or people they have known before. Also situatoins may happen that

a strong recommendation letter from less famous professors be stronger than those mediocre ones from famous guy. Similarly, assume users who have reviewed thousands of times will have higher weights. But when the similarity between chosen user 1 and a highly-weighted user 2 is very low(close to 0), it is still less likely that user 2 will give the chosen user a proper recommendation business than others. But if their similarity is high, even not the highest, user 2 will have a better chance to give a better recommendation.

6.3 Allowable error

Allowable error is defined as if there is a range such that the difference between real and experimental data falls in, we will still consider the experiment is acceptable. Usually, if we only consider few parameters, the recommendation accuracy will be low. In my recommendation algorithm, all the recommendations made from the Training Set will be assigned a 5 star and then compared with the data in Test Set if there is an edge between specific user and specific business. Such recommendation will be biased because $N(\mu)^{1/4}$ will not yield a strong linear relation. So introducing allowable error will help make such recommendation be less biased.

7 Analysis

7.1 Input data analysis: Power Law

In this section, several methods will be applied to analyze whether the Yelp data of reviews for businesses and users follow Power-Law. In Figure [1] and Figure [2], we can guess that they are. So I use Curve distribution function in MATLAB to find the best curve for Businesses shown In Figure [3].

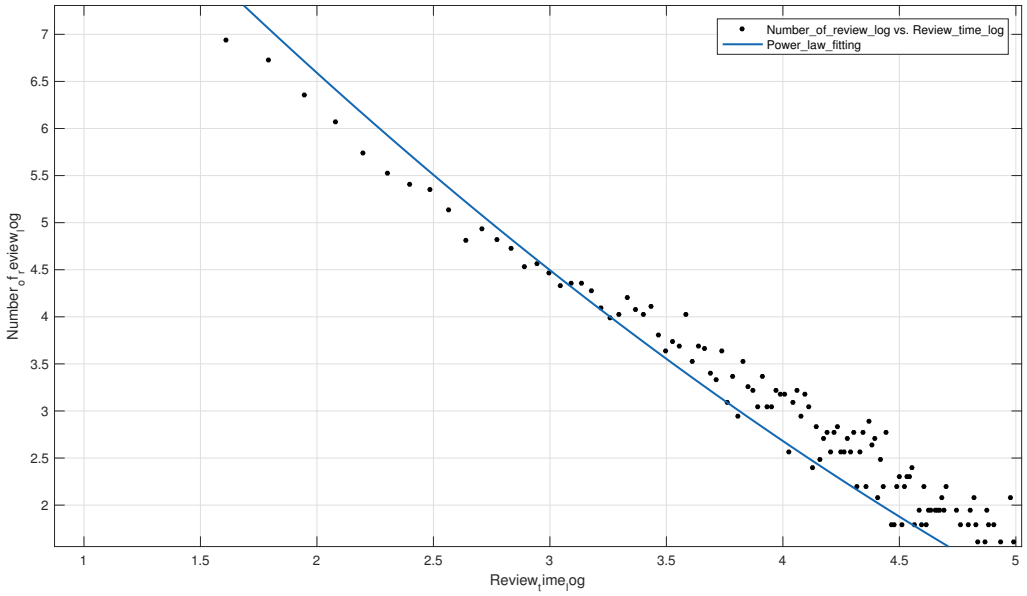


Figure 3: Linear regression for log-log distribution of number of businesses(y-axis) versus number of review times(x-axis)

The best fit is close to a linear equation for this log-log distribution. In other words, the original distribution for Number of businesses versus number of business reviews follows Power Law. Similarly, user review distribution also follows Power Law. This phenomenon again verifies the existence of Rich-Get-Richer Effects.

7.2 Input data analysis: Stars

In this section, analysis on user review(stars) will be done and some basic information of stars will be provided. In Figure [4], we can see the bar distribution of stars. I try to fit popular distribution such as Normal, Gamma, Weibull to this distribution but their MSE are all too high. From this bar figure, we can see it is very rare for people to rate 1 star to businesses which makes our prediction a little easier. Our previous range is $[1, 5]$, but now we can narrow it down to almost $[2, 4.5]$ by 80% confidence interval.

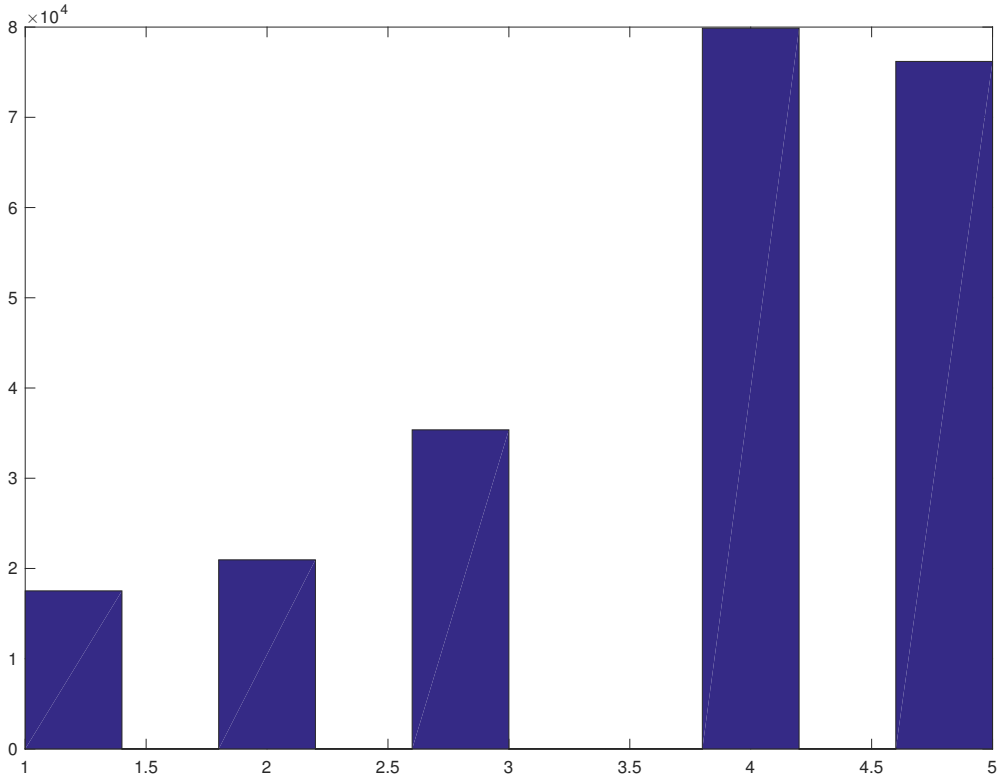
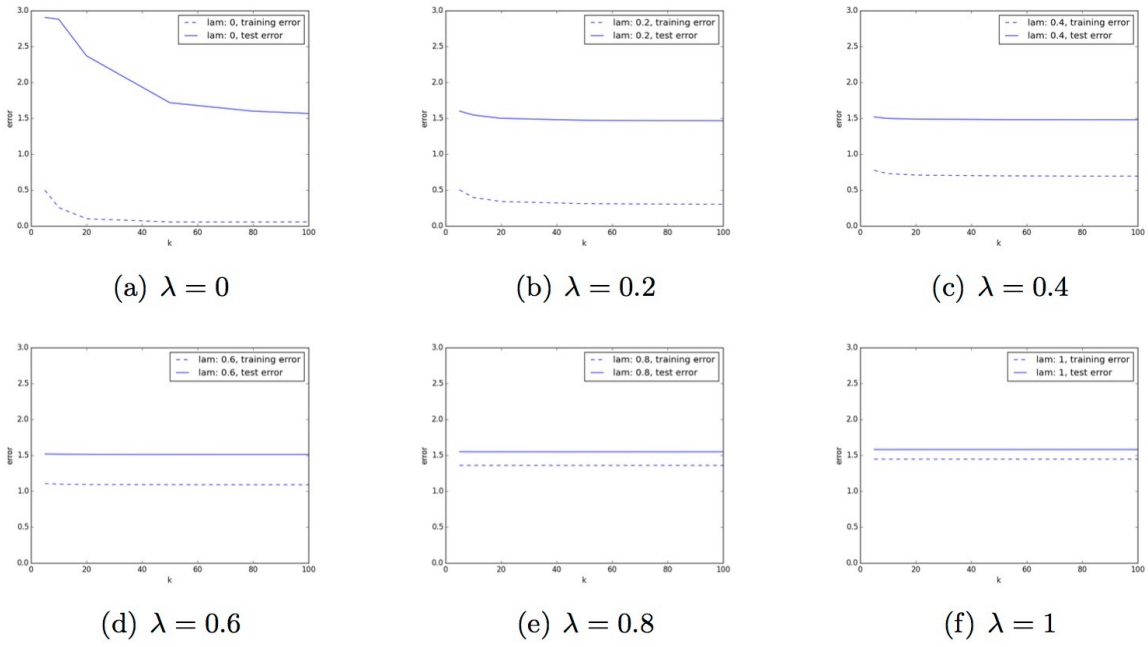


Figure 4: Stars of businesses by users

7.3 Prediction Threshold analysis: λ

In this section, I run several tests for different parameter λ from 0 to 1. When $\lambda = 0$, that means the algorithm is the same as regular Jaccard Coefficient because similarity will be at least 0, so $\lambda = 0$ is served as benchmark for regular performance. When $\lambda = 1$, the algorithm will reject all predictions, so it will perform as original data. In Figure [5], 6 different λ are presented.



(a) $\lambda = 0$ (b) $\lambda = 0.2$ (c) $\lambda = 0.4$

(d) $\lambda = 0.6$ (e) $\lambda = 0.8$ (f) $\lambda = 1$

Figure 5: MSE of λ test

In Figure [5](a), the MSE changes rapidly as the experiments went which shows the instability of prediction. When we increase the λ, the good thing is the predictions become more accurate, the bad aspect is we will not update a lot of the predictions. For example, when $\lambda = 0.8$, the update rate is only 39%, if we set λ too high, it is true that MSE will decrease but the prediction accuracy will also decrease. After several experiments, when $\lambda = 0.4$, the overall performance will be the best and prediction accuracy with allowable error 0.5 will be 78.65% which can rank top 10% in Kaggle.

7.4 Recommendation Analysis: $N(\mu)^{1/4}$

In previous section [6.2], we introduce my prototype algorithm for recommendation. In that algorithm, we introduce $N(\mu)$ and $N(\mu)^{1/4}$.

$N(\mu)$ is defined as number of reviews counted by user μ. With $N(\mu)$ as a parameter, those who have contributed a lot of reviews will have higher weight than others. But the distribution of $N(\mu)$ is shown in Figure [6]. The max and min value of $N(\mu)$ is shown in Table [2]. Obviously, $N(\mu)$ is not the right parameter as its range is from $[1, 6390]$ but similarity range is only from $[0, 1]$. Even if user 2 has similarity 1 with chosen user 1 with low review history, and user 3 has almost 0 similarity with chosen user 1 but with super high review history, $score'(\mu_1, \mu_3) > score'(\mu_1, \mu_2)$. Then we will recommend by user 3 which is not true at all. Then I tried several other parameters such as $\log(N(\mu))$, $\log(\sqrt{N(\mu)})$ and

$N(\mu)^{1/4}$ and figures are shown in Figure [9]. $\log(\sqrt{N(\mu)})$ and $N(\mu)^{1/4}$ are both close to linear equation, but the minimum number of $N(\mu)^{1/4}$ is 1 which is better compared to 0, so I choose $N(\mu)^{1/4}$ as the parameter. Better parameter expression could be found and I would assume that will yield a better matching result.

Table 2: Basic information for parameters

	Mean	Min	Max
$N(\mu)$	51.4338	1	6390
$\log(N(\mu))$	5.8537	0	8.6668
$\log(\sqrt{N(\mu)})$	2.9268	0	4.3334
$N(\mu)^{1/4}$	4.4672	1	8.7295

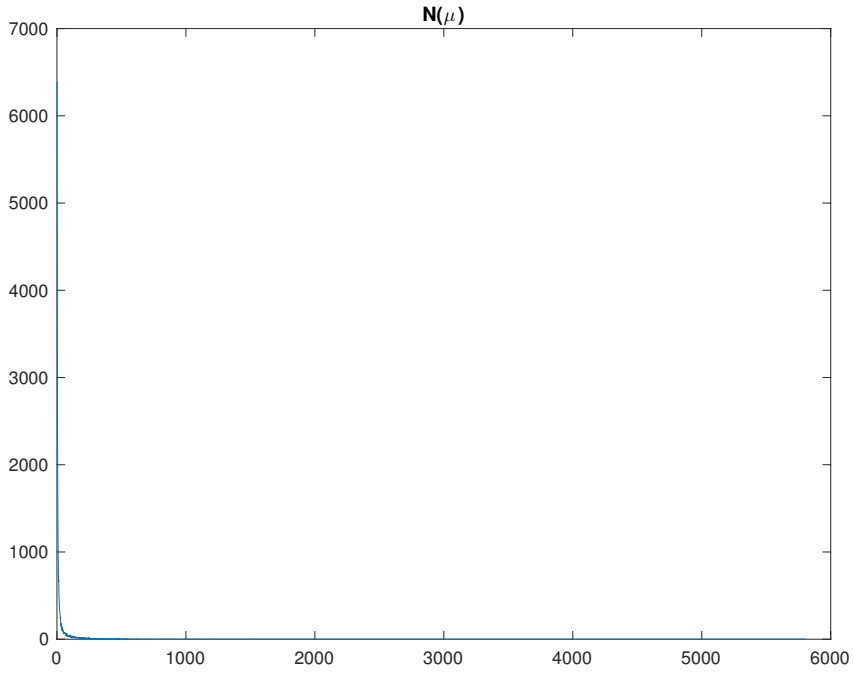


Figure 6: Distribution for $N(\mu)$

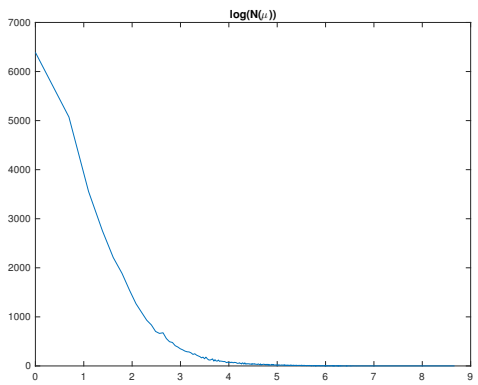


Figure 7: Distribution for $\log(N(\mu))$

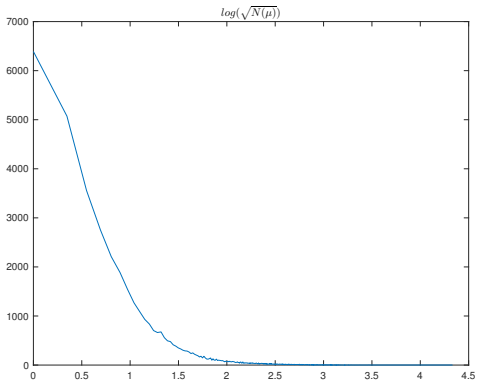


Figure 8: Distribution for $\log(\sqrt{N(\mu)})$

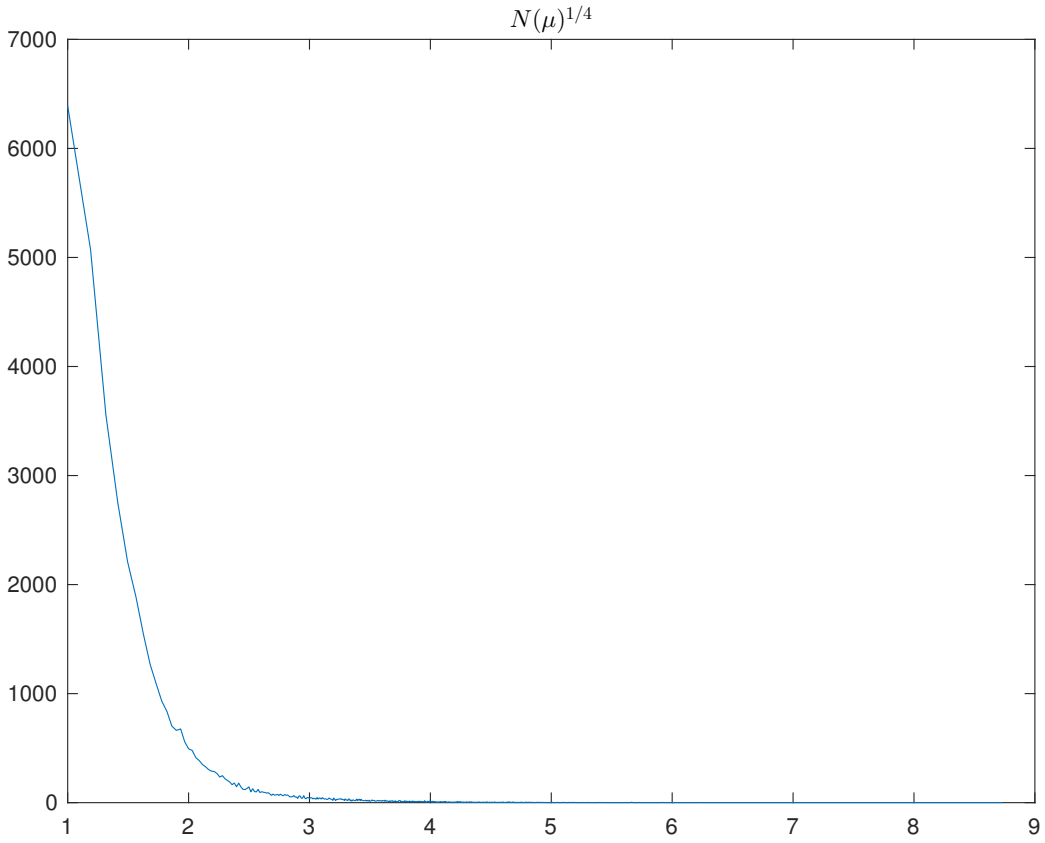


Figure 9: Distribution for $N(\mu)^{1/4}$

7.5 Recommendation Analysis: Results

In Table [3], recommendation accuracy results with different allowable error will be shown.

Table 3: Recommendation accuracy results comparison

allowable error	0	1
Random Benchmark	0.13	0.24
Netflix Recommendation algorithm	0.17	0.31
Weighted common neighbor	0.27	0.33

In Table [3], when allowable error is 0, that means in test set, the chosen user also rate chosen business 5 stars. When allowable error is 1, the chosen user will rate 4 or 5. Of course under the second situation, recommendation accuracy will be higher. My algorithm has a much better performance when allowable error is 0 but does not improve very much when we increase the allowable error. The reason for that is the size of Training and Test set. Though there are $5,105$ users and $1,205$ businesses in Test set, for a specific user, the recommendation shopping places from Training set only have very low response rate(user in Test Set rates that business), so our actual test cases are very limited which makes our results less accurate.

8 Future Work

There are a lot of works people can do in this problem. I will mention two of the most important parts. The first one is the location. Yelp Data Challenge dataset provides us with the location of each business, when we try to find similarity, location is also a very important issue: people in the same zip code area such as 23185(Williamsburg, VA) tend to go to the same place than visitors from Los Angeles. If we assign a weight to the location, I would believe the prediction and recommendation results will be better.

The second issue is about $N(\mu)^{1/4}$ parameter. No proof has been done about when the recommendation will be unbiased. Also, there must exist a better function for a linear equation than $N(\mu)^{1/4}$. If people can find a less biased parameter or a better linear equation, I would assume the accuracy will be better too with larger datasets.

References

[1] Nuno Ferreira, Jorge Poco, Huy T Vo, Juliana Freire, and Cláudio T Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2149–2158, 2013.

[2] Business Insider. Uber isn't making nyc traffic worse.

[3] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[4] Kevin (Junhui) Mao, Xiao Cai, and Ya Wang. Link prediction in bipartite networks - predicting yelp reviews.

[5] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[6] Anand Rajaraman, Jeffrey D Ullman, Jeffrey David Ullman, and Jeffrey David Ullman. *Mining of massive datasets*, volume 1. Cambridge University Press Cambridge, 2012.

[7] Yelp. Yelp business rating prediction.

[8] Wenliang Zhong, Rong Jin, Cheng Yang, Xiaowei Yan, Qi Zhang, and Qiang Li. Stock constrained recommendation in tmall. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2287–2296. ACM, 2015.