# Forecasting Ratings and Review Counts for Yelp Businesses

Alexandros Anemogiannis
Stanford University
anemogiannis@stanford.edu

Vighnesh Sachidananda
Stanford University
vighnesh@stanford.edu

## 1 INTRODUCTION

When assessing the value of content reviewed by others, it's important to take into account both its average rating as well as the number of ratings that it's received. There is a potential for bias in ratings given how many ratings and the number of reviews a business has. In this paper consider the problem of estimating the score and number of reviews that a business receives using information contained in the early votes. After examining the Yelp dataset, we've found that a business' rating converges to within $\epsilon = 0.01$ in its first $k = 10$ reviews, on average; however, it takes an average of roughly 1000 days for a business to receive 10 reviews. Additionally we show that early ratings to Yelp are biased favorably towards businesses. By correcting for this relationship we show that we can more accurately predict future reviews. To show this we build a recommender system with up to 25% less mean squared error from this observation we have made. Furthermore, we also implement a combination of statistical and graphical network analysis techniques to predict the number of reviews a business will receive in a specified number of days. We outline the models therein, achieving excellent accuracy in fitting preferential attachment models and polynomial models for time series prediction.

## 2 RELATED WORK

### 2.1 Collaborative Filtering

Amazon's Item-to-Item Collaborative Filtering paper provides insight into how traditional recommendation systems work, what qualities are essential for production systems at Web scale and how the personalization team at Amazon approached the recommendation problem. The paper begins by outlining and evaluating three traditional approaches to providing recommendations: (1) "Traditional" or User-User Collaborative Filtering, (2) Cluster Models, and (3) Search.

Traditional Collaborative Filtering works by generating recommendations based on finding similar users to the queried user and highly ranking the items that they bought or rated. From these highly ranked items, the ones the queried user has already bought or purchased are removed from consideration. Typically, the cosine distance is used to evaluate how close a user is to another user. This metric balances two key considerations: (1) the intersection or agreement of the two users and (2) the number of items the users have rated/bought in total. Although this algorithm generally performs well in practice, it is computationally very expensive and often time practitioners must relax the problem in various ways that degrade performance. Typical relaxations include user sampling, dimensionality reduction and discarding the most popular/unpopular items.

Clustering methods operate by clustering the customer base into segments and generating independent recommendation systems for each cluster. These models have better online scalability then the User-User Collaborative Filtering case however clustering is quite expensive with the scale of data points and dimensions (which scales with number of items) that most Web companies experience and must be done offline. The authors claim from their experience that the similar users found are not the most similar users possible (i.e. the ones found in Collaborative Filtering) and thus the quality is not as good as it can be.

Lastly, the authors examine search based methods. Such methods work by extracting features from items already purchased or rated by users and looking up other items with similar keywords or subjects. For example, if a customer buys a Godfather DVD other crime drama titles could be recommended by a search algorithm. This algorithm seems to work well when a user has interacted with a small number of items which is traditionally where Collaborative Filtering does not do well. However, once users start interacting with more items it becomes very hard to prioritize information and produce a summary of these interactions to the search algorithm. Quality degrades as a consequence and such search algorithms do not exactly facilitate the goal of finding new items the customer could not find without the help of the algorithm.

### 2.2 Simrank Bipartite Matching

Much of the work done with online reviews has focused on building recommender systems that use collaborative filtering to determine what items a user will likely enjoy[5]. These approaches use existing ratings to determine groups of similar users and similar objects, which can then be used to recommend content that aligns with a user's preference. Rather than make specific recommendations for each user, we hope to recover a global representation of an object, namely its average rating the number of ratings it receives, using only a few of its initial reviews.

In order to do so, we use the notions of user-content similarity and user status presented in Anderson et al. [1]. user similarity specifies the similarity between two users based on the actions they take, and user status specifies how a user is perceived by the rest of the community.

### 2.3 Preferential Attachment & Fitness Attachment

The preferential attachment model was first proposed by Barabási as an explanation for the power-law distribution of degrees that commonly appears in networked systems (eg. World wide web, academic citations)[2]. The intuition behind this formulation is that as the degree of a node increases, so does the probability of

assigning an edge to the node. More specifically, the dynamics underlying the edge attachment process are such that a node $v$ with degree $k$ has a probability $p(v_k) \propto k^{\alpha}$ of receiving a new link. This gives rise to a power law distribution of degrees in the network characterized by $P(k) \propto k^{-\alpha}$, where $P(k)$ indicates the probability that a node has degree $k$. Note that $\alpha$ is assumed to be invariant to time.

### 2.4 Fitness Attachment

Other researchers have suggested that power law distributions in networks may arise not necessarily from the degree of a node but some value intrinsic to the node. To take this into account, Bianconi specifies a simple augmentation to the underlying edge attachment dynamics [3]. Under the proposed approach, the probability of an entering edge attaching to a node $v$ with degree $k$ is

$$p(v_k \text{receives edge}) \propto \eta_v \cdot k^{\alpha},$$

where $\eta_v$ is the fitness score of node $v_k$. In the rest of the paper, Bianconi derives the resulting distribution of degrees in the network, which is a power law

$$P(k) \propto \frac{k^{-(1+c)}}{\log k},$$

where $c$ which is a constant related to the distribution of fitness scores $\rho(\eta)$ and $\alpha$.

## 3 DATASET

We've decided to narrow our focus to the Yelp dataset due to its convenient formatting and quality. Yelp is a platform that allows users to post reviews of (predominantly restaurant) businesses. The reviews include a 1 to 5 star rating, possibly accompanied by text and pictures that provide additional context for the rating. The dataset contains 4,700,000 reviews about 156,000 businesses spread across 12 metropolitan areas.

In order to reduce the computational overhead of running our algorithms on the dataset, we restricted our focus to the businesses in Las Vegas, which contains 24,768 businesses and 1,500,000 reviews. This geographic restriction has the added bonus that the users submitting ratings ***.

## 4 METHODS & ALGORITHMS

### 4.1 Filtering Rating Biases

Since our project focuses on early detection and denoising of ratings, we observe the rating bias inherent to the Yelp Dataset. We find that ratings are biased and specifically the first few ratings for a restaurant are favorable for a restaurant. These findings, which were also seen in [Groupon Yelp paper], are shown in the below figure.

This figure displays the difference between the average rating $E[\bar{r}]$ after k reviews and the final rating a business will have $E[r_{true}]$. On the y axis we show this difference and on the x axis, we show how this difference appears at different values of k. We range k from 0 to 10 and include results averaged over 10,000 businesses that have at least 1 review.

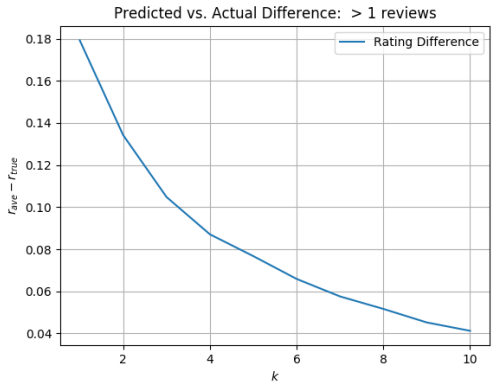


**Figure 1: Difference between average rating at index k and steady state rating of a business**

This result likely has social implications which our project aims to uncover. In subsequent sections of this milestone, this bias can be leveraged to make more accurate predictions of the future or steady state rating value that a business will achieve.

Drawing on the previous section we show that the implications of the social bias include the ability to make better predictions about how good a restaurant is given only an early set of ratings.

We empirically show evidence of this claim by comparing the mean squared error between the average rating and true rating as defined in the previous section with a linear estimator. We use the Lasso estimator to produce the improved estimate [Lasso Reference here]. These results are shown in the plot below.

From the plot we see that for $k \leq 4$ the improvement in rating prediction is especially pronounced. These results further reveal the unbiased nature of early ratings as seen in Figure 1.

Although the results from the previous section demonstrate it is feasible to remove bias in rating predictions, we wanted to understand whether or not this problem is of consequence. Considering that it only takes about 6 reviews to achieve a Mean Squared Error of .1, this appeared to be worth addressing. If these 6 reviews occurred over one weekend, this prediction problem is not really much of a problem at all.

Once we analyzed the dataset, we understood quickly that the arrival rate of early reviews is actual quite long. We computed the time to reach k reviews across 10,000 businesses and report these numbers in the figure below.

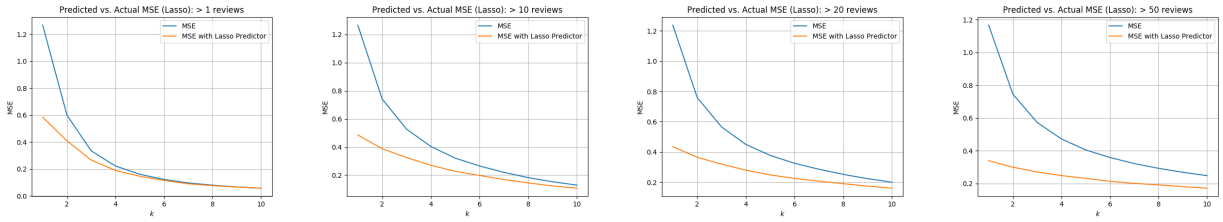


**Figure 2: Predicting with Mean vs. Lasso Unbiased Estimator with Varying Business Popularities**

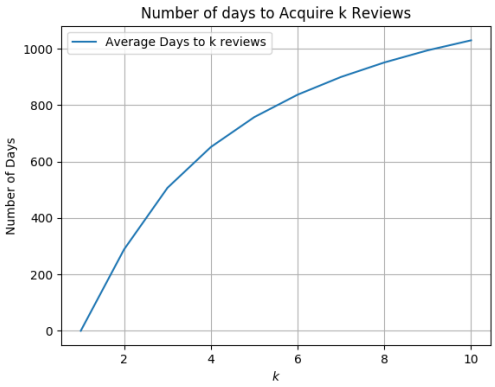


**Figure 3: Average number of days to reach k reviews**

As seen in the above figure, it takes more than 2 years on average for a business to obtain its first 6 reviews. In this time the biased average rating is displayed to users. Consequently, we find the initial results promising and hope to use insights from the course to further our analysis.

*4.1.1 Improving Recommendations with Temporal Information.* Noting the temporal bias apparent in the Yelp Dataset, we modidfy the rating/link prediction algorithm in [5].

This formulation crowdsources reviews from the most similar users to the queried user. Simularity is often defined as the cosine similarity in this setting:

$$s(u,b) = \frac{u_a \cdot b_a}{||u_a|| \cdot ||b_a||}$$

From these similarities, we have many available algorithms to make predictions and here we will focus on using a Nearest Neighbors approach. This procedure has been analyzed in many texts, including [4] For a user $u$ and business $b$ and its $k$ closest neighbors, we denote the predicted review $\hat{r}$ as follows:

$$\hat{r} = \frac{\sum_i^k r(u_i,b) \cdot d(u_i,u)}{\sum_i^k d(u_i,u)}$$

Notice that here, we take a weighted average of the neighbors scores for the queries business $b$.

In order to make the computation on this large dataset feasible we make computational considerations.

- We consider only the reviews in Las Vegas, the city with the most reviews in the Yelp Academic Dataset
- Our analysis covers only the top 500 businesses and the 243,212 users that visited them
- Instead of exact similarity we use an approximate nearest neighbor algorithm. Specifically, we use the annoy nearest neighbors package by Spotify.

## 4.2 Review Arrival Process

In order to predict the number of yelp reviews made each day we fit a simple quadratic model. We train the model on data from 2004 up to late 2008 and predict from 2008 to 2017. Based on the model and data seen in Figure 4, we can see the growth of the Yelp network and the accuracy of the predicted trend.

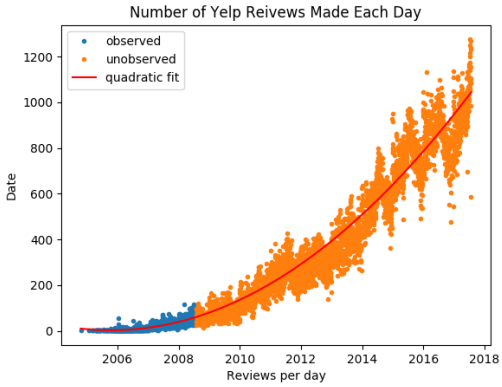


**Figure 4: Number of Yelp reviews made each day**

With an accurate model for predicting the arrival process of we now focus on building a preferential model to analyze the evolution of the network and the impact of degree on this phenomenon.

## 4.3 Preferential Attachment Model

The power law distribution of degrees in the Yelp dataset 5 (restricted to Las Vegas) motivated our exploration of using the preferential attachment model to understand the dynamics of edge attachment and use this understanding to predict review counts of businesses. Given observations of the network up to some time $t_1$, our goal is to predict how edges (ie. reviews) that arrive between $t_1$ and $t_2$ are distributed among the businesses. Let $E_t$ be the set

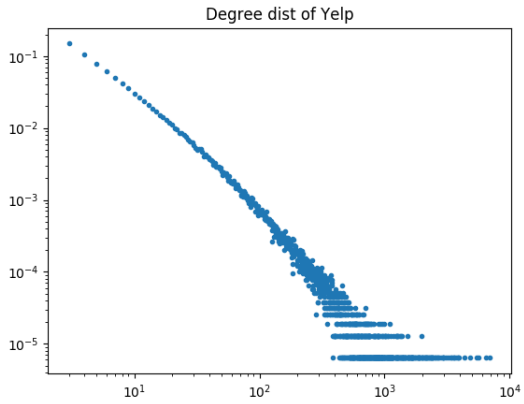


**Figure 5: Estimated probability of attachment**

of edges that arrive at time $t$ where $e_b \in E_t$ indicates that edge $e$ attaches to business $b$, and let $\rho_a(d)$ be the probability that an arriving edge attaches to a business with degree $d$. From the data collected up to time $t_0$, we obtain the empirical PDF by the following

$$\rho_a(d) = \frac{\sum_{t=0}^{t_1} \sum_{e_b \in E_t} \mathbf{1}\{\deg(b,t) = d\}}{\sum_{t=0}^{t_1} \sum_{b \in \mathcal{B}} \mathbf{1}\{\deg(b,t) = d\}},$$

where $\deg(b,t)$ is the degree of business $b$ at time $t$. The intuition behind this expression is that we consider, across time, the number of attachments to a node with degree $d$ normalized by the number of nodes with degree $d$ that the edge could have attached to. Without this normalization, $\rho(d)$ would be heavily skewed towards nodes with low degree since there are many more of them than those with larger degrees. We plot $\rho_a(d)$ in 6, for which we fit a least-squares line. The slope of the line corresponds to the relationship $\rho_a(d) \propto d^{0}.82$ (since the plot is log-log), which is slightly sublinear preferential attachment.

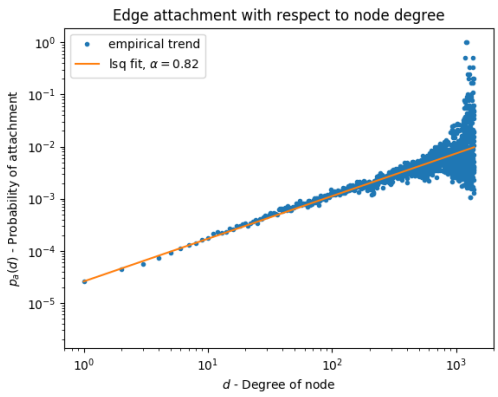


**Figure 6: Estimated probability of attachment**

Given $\rho_a(d)$, we predict how the edges will be distributed among the businesses between times $t_1$ and $t_2$ using the Monte Carlo method (see Algorithm 1). Upon the arrival of each edge, we draw $d^*$ from $\rho_a(d)$ to determine the degree of the business that receives the edge. Let $B(d^*)$ denote the set of such businesses. Then $b \in B(d^*)$ is chosen proportional to it's current average rating, ie.

$$p(b_i) = \frac{\text{rating}(b_i)}{\sum_j \text{rating}(b_j)},$$

so that, among businesses with equivalent degrees, the ones with higher ratings are more likely to be visited by a user.

**Algorithm 1** Predicting degree of businesses at time $t_2$

**Input**:
$\rho_a(d)$ : empirical PDF of attachments up to $t_1$
$D$ : dictionary mapping degrees to businesses
$B$ : dictionary mapping business to its degrees
$\hat{m}$ : estimated number of edges added between $t_1, t_2$
$N$ : the number of Monte Carlo iterations

**for** $i = 1$ to $N$ **do**
  **for** $k = 1$ to $\hat{m}$ **do**
    Randomly draw $d^*$ from $\rho_a(d)$
    Randomly draw business $b^*$ from $D[d^*]$
    Assign $m$th edge to business $b^*$
    Update $B$ and $D$ to reflect increment in $\deg(b)$
    **if** $\deg(b^*)$ outgrows domain of $\rho_a(d)$ **then**
      Extrapolate $\rho_a(d)$ to be defined for $\deg(b^*)$ using a least-squares fit to $\rho_a(d)$ (as done in 6)
      Renormalize $\rho_a(d)$ over its new domain.
    **end if**
  **end for**
**end for**
Average over iterations: $B(b) /= N, \ \forall b \in B$

**Output:**
$\hat{\deg}(b, t_2)$ for all $b \in \mathcal{B}$

# 5 EXPERIMENTAL RESULTS

## 5.1 Predicting Ratings

Here we outline key differences in our rating prediction algorithm from traditional methods and showcase the results of this algorithm.

*5.1.1 Imputing the Ratings Matrix.* The main alteration we make to predicting interactions in the bipartite graph is imputing the ratings matrix with values that are cognizant of the index of the review. This works to produce a more accurate estimate on the training set of data we offer the algorithm. We learn the relationship between features and the steady state score of yelp ratings. The goal of minimizing error with these weighted feature predictions is counterbalanced with minimizing the magnitude of the weights and encouraging sparsity (feature selection).

$$\min_{\beta} \{||y - X\beta||_2^2 + \lambda||B||\}$$

Through calculation of the above learned coefficients, we produce an imputed matrix $\tilde{X}$ that we run collaborative filtering on.

$$\tilde{X} = X\beta$$

*5.1.2 Feature Analysis.* We analyze the feature weights that our algorithm learns and detail them below:

| k | Average User Age (Days) | Current Business Rating |
|---|---|---|
| 0 | 0.000072 | 0.421116 |
| 1 | 0.000069 | 0.616808 |
| 2 | 0.000058 | 0.743212 |
| 3 | 0.000060 | 0.813980 |
| 4 | 0.000062 | 0.853868 |
| 5 | 0.000057 | 0.882040 |
| 6 | 0.000053 | 0.899969 |
| 7 | 0.000046 | 0.914242 |
| 8 | 0.000043 | 0.925571 |
| 9 | 0.000044 | 0.933943 |

**Figure 7: Lasso Feature Weights**

From the feature weights, we see that accounting for the user age, a proxy for expertise is more important as a business has few reviews. Conversely we see that the current business rating (mean) is more faithful as a business gets more reviews as we had seen previously.

*5.1.3 Accuracy.* When applying the temporal modifications to the ratings matrix, we notice that in most cases, this helps improve recommendation quality.

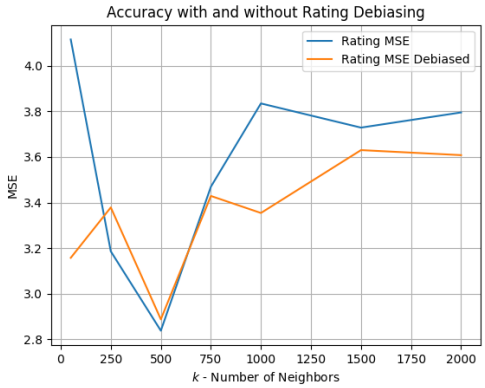


**Figure 8: Recommendation Accuracy with and without accounting for temporal effects**

## 5.2 Predicting Review Counts

Given a window of prediction $(t_1, t_1 + \Delta)$, we use the Monte Carlo method described in 4.3 to generate an estimate of the degrees of the businesses at time $t_2$. These estimates $\hat{\deg}(b, t_1 + \Delta)$ are compared with $\deg(b, t_1 + \Delta)$ using average percent error

$$\text{APE} = \frac{\hat{\deg}(t_1 + \Delta) - \deg(t_1 + \Delta)}{\deg(t_1 + \Delta)}.$$

A plot of the prediction error is shown in 9. To generate this plot, we varied the starting time $t_1$, which correspond to the number of observations of the edge attachment process, and the prediction gap $\Delta$, which corresponds to the amount of time that we must.

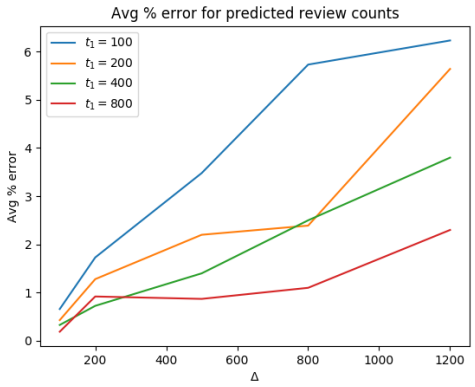


**Figure 9: Average percent error of review counts**

Though the performance of this approach is poor (due to the large errors), there are, there are several intuitions that we can draw from 9. Given more observations of the edge attachment process (ie. larger $t_1$), the accuracy of $\rho_a(d)$ improves, so the dynamics modeled in the Monte Carlo better represent what's occurring in real-life, which obtains a lower error. Given a larger window $\Delta$, we also observe a drastic drop in performance which we believe is due to the massive state space that the Monte Carlo simulations need to explore. As $\Delta$ increases,

*5.2.1 Fitness-Augmented Preferential attachment.* We also attempted to form predictions based off a preferential attachment model augmented with fitness. A natural choice for a node's fitness score is it's average rating since, in general, users are more likely to eat at restaurants with higher ratings. A plot of the empirical PDF of the probability of edge attachment $p_a(d)$ against the degree of a node is shown below

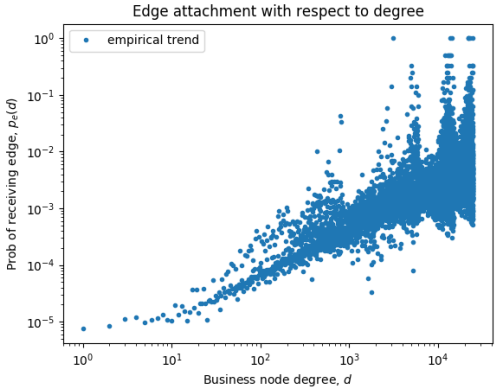


**Figure 10: Sample PDF using fitness model**

Since the scores increased the variance of the support of $p_a(d)$ (since it now occurs over the range (0, 5 $\times$ the max node degree) rather than (0, 5 $\times$ the max node degree), the variance of $p_a(d)$ increases, making it risky to use for estimation.

## 6 CONCLUSIONS

*6.0.1 Forecasting Ratings.* In this paper we demonstrate the temporal bias in Yelp ratings. Earlier reviews will be rated higher than what the average rating will converge too. This bias has negative implications for forecasting unseen ratings in a recommender system.

We show how we can incorporate knowledge of temporal biases in predicting ratings and also show that doing so can yield favorable results for prediction. In some cases of our simulations, we improve the Mean Squared Error in rating prediction by more than 25%. Modifying the ratings matrix does not appear to pose negative implications in the few cases where this modification does not help we are no farther than 10% MSE away from the original collaborative filtering solution.

Predicting the review counts was an ambitious task, and the poor results stem from cascading errors arising from several sources. The Monte Carlo approach failed to explore the large state spaces arising from large values of $\Delta$ and any error from the predicted number of edges given a time window cascaded with this error, giving extremely poor resuluts.

## REFERENCES


[1] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Effects of User Similarity in Social Media. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM '12)*. ACM, New York, NY, USA, 703–712. https://doi.org/10.1145/2124295.2124378

[2] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[3] Ginestra Bianconi and A-L Barabási. 2001. Competition and multiscaling in evolving networks. *EPL (Europhysics Letters)* 54, 4 (2001), 436.

[4] Gérard Biau, Benoît Cadre, Laurent Rouviere, et al. 2010. Statistical analysis of k-nearest neighbor collaborative recommendation. *The Annals of Statistics* 38, 3 (2010), 1568–1592.

[5] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (Jan 2003), 76–80. https://doi.org/10.1109/MIC.2003.1167344