Reviews and Neighbors Influence on Performance of Business

Mohit Kothari

Computer Science and Engineering
University of California, San Diego
mkothari@ucsd.edu

Sandy Wiraatmadja

Computer Science and Engineering
University of California, San Diego
swiraatm@ucsd.edu

***Abstract*—The task of rating prediction has been gaining popularity, especially after several companies come up with competitions, such as the Netflix Challenge[1] and the Yelp dataset challenge[12]. In this paper, we aim to modify and incorporate two methods for rating prediction of businesses, one utilizes the semantics of the review text, while the other uses the influence of the surrounding businesses. For the review text semantics, we combine unsupervised learning technique, Latent Dirichlet Allocation, with supervised learning method, such as SGDRegressor. On the other hand, we use K-D Tree implementation to find the nearest neighbors of a business and incorporate them into the feature vector representation of the business for predicting its rating. For experimentation, we use two datasets, Yelp and Google Local, and compare the performance of the different models on the two datasets. We find that semantics representation of the review text performs quite well, achieving an RMSE of 0.5985 for Yelp dataset and 0.5643 for Google dataset. Adding the neighbors influence only improves the performance by 0.3%, with an RMSE of 0.5970 for Yelp dataset and 0.5626 for Google dataset, which shows that the neighbors do not seem to influence the business' rating as much. We also discuss the challenges faced while working with Google Local dataset because of its sheer size and incompleteness.**

***Keywords*—Yelp, Google Local, Rating prediction, LDA, K-D Tree, SGD Regressor, SVR**

I. Introduction

In the last few years, there has been an increase in the interests on rating prediction researches, both in academia and industry. This prediction is often used in the recommender systems. Products, either items, businesses, or services, that are predicted to have high ratings are then recommended to users. In this paper, we are specifically interested in rating prediction of businesses. One key difference between a business and an item is that it has a physical location. We believe that this is one factor that can have an impact on the businesses, either positively or negatively. Whenever a person visits a business, there is a possibility that the same person will also visit other places that are around the area. Higher foot traffic can usually be observed in neighborhoods where there are several good businesses in the region. Therefore, a new business owner can get the benefits of getting new customers more easily in those neighborhoods. We refer to this neighbor effect as the extrinsic factor that influences a business. On the other hand, there are still some intrinsic factors that are more influential in the success of a business, such as the quality of the service or the item itself. The question that we wish to explore in this paper is how much effect the extrinsic factor has compared to the intrinsic factors, in predicting the rating of a business.

To study the business rating prediction that utilizes geographical location on top of its intrinsic characteristics, we perform our analysis on Yelp and Google Local business rating data. There are several factors that can be included in the intrinsic factors of a business, such as its reviews, number of check-ins, hours of operations, etc. Among these, one of the most important factors seems to be the reviews written by the users about the business itself as it presents a rich context about the quality of the business and the type of the business itself. Therefore, we decide to use only the review text to represent the intrinsic characteristics of the business.

The remaining paper is organized as follows. In Section II, we discuss other studies that are related to our work. We also explain some background on the techniques that we use in our experiments in Section III. The datasets used are explained in Section IV, followed by some details of the exploratory analysis done on the datasets in Section V. In Section VI, we discuss the features that we choose for our feature vectors, and we proceed to explain further the models that are being compared, including the baseline model, in Section VII. The results of our experiments can be found in Section VIII. We conclude this paper by analyzing the results of our experiments in Section IX.

II. Related Work

A similar research related to geographical location effect on a business' rating has been done by Hu, et. al. [5], which was presented in Special Interest Group on Information Renewal (SIGIR) Conference in 2014. In their paper, they adopt the latent factor model using Matrix Factorization, which incorporates the geographical neighborhood influence, along with other intrinsic influences. Their best model achieves an RMSE of 1.0072 on the old Yelp dataset, which utilizes neighborhood combined with business category, user review content, and business popularity influences. The geographical neighborhood is modeled as the linear combination of the latent factors for the extrinsic characteristics of its neighbors. There are several observations mentioned in the paper that

[1] http://www.netflixprize.com/leaderboard

we explore and discuss further in Section V. This general idea, suggested by Hu, becomes our motivation for this paper. However, instead of using Matrix Factorization, we decide to represent each business as a feature vector and run linear regression on the vectors.

There have also been a lot of research that are previously done which incorporates review text in a rating prediction task. Wang et. al. [11] analysed review text to create word clouds and get more semantic information about the reviews and help users with review reading. Hood et. al. [4] try using sentiment analysis (using wordnet) [8] on the user reviews in combination of user clustering to predict the success of the business in the future. To generate the review text features, they add Part-of-Speech tag on each token of the text and take only the adjectives and nouns, as a measure of how positive or negative the reviews are. A class project by Fan et. al. [3] use reviews text alone to predict business' rating. They use a naive word frequency model along with Part-of-Speech tagging of the review text as their feature vectors. They achieve an RMSE of 0.6014 on the old Yelp dataset by using Linear Regression model. A very interesting research done by McAuley et. al. [6] tries to come up with statistical models by fusing latent review topics along with latent dimensions that are present in rating data. By doing this, they try to get more natural interpretations of users' review scores and higher prediction accuracy of the ratings themselves. They achieve an improvement of around 4.53% using their Hidden Factors as Topics (HFT) model as compared to Latent Dirichlet Allocation (LDA), and an improvement of 3.78% as compared to latent recommender system. This general idea, suggested by Fan and McAuley, becomes our motivation to use the latent topics present in user generated review text and use those topics as the feature vector to predict a business' ratings.

III. Background

There are two main tools that we use to help us define our model.

A. K-D Tree

In order to get the nearest neighbors of a business, we utilize a multidimensional binary search trees, which is commonly called k-d tree [1], where k is the dimensionality of the search tree. K-d tree was invented in the $1970s$ by Jon Bentley. It is a data structure for storing multikey records, that can be used as an efficient way to store information that is retrieved by associative searches. Each node in the tree is a k-dimensional point, and they are organized in such a way that the non-leaf nodes can be thought of as partitions of the hyperplane space, splitting it into different regions. An example of a 3-dimensional k-d tree can be seen in Figure 1, where the different partitions, created from the nodes, are highlighted. This organization makes k-d tree ideal for handling multiple different types of queries, using multidimensional search keys, efficiently.

One such query is for the nearest neighbor searches, which is exactly what we need for this experiment. In order to do this query, we need to specify a distance function D_f that defines how far two points are. An example of such a function is the most commonly used Euclidean distance function. However, for our purpose, since the Earth is a sphere and the geographical location is represented in latitude/longitude coordinates, we cannot simply use the Euclidean distance function. In Section V, we discuss further on how to handle the spherical coordinates of the geographical locations. K-d tree nearest neighbor queries has an average running time of $O(\log n)$, with the worst case running time of $O(n)$ when it has to visit all the nodes in the tree.

[2]Source: http://en.wikipedia.org/wiki/K-d_tree

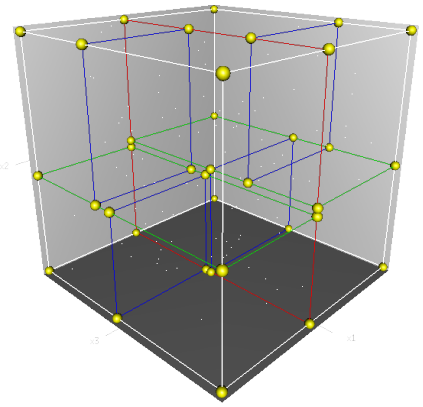


Fig. 1: A 3-dimensional k-d tree[2].

B. Latent Dirichlet Allocation (LDA)

In order to get the latent topics present in the reviews text, we utilize an unsupervised learning algorithm called Latent Dirichlet Allocation (LDA) [2]. Given a document list, LDA can uncover hidden topics. One of the major assumption of LDA is that, given a document, the words occurring in that document are independent of each other, i.e. it has a bag-of-word model assumption. LDA associates each document $d \in D$ with a K-dimensional topic distribution, θ_d, which is a normalized weighted vector specifying the fraction of each topic presents in the document. In short, each element of the vector, i.e.$\theta_{d,k}$ depicts the probability of the occurrence of topic k in document d.

On the other side, each topic $k \in K$ has an associated word distribution, ϕ_k, which is a normalized weighted vector specifying the probability distribution of words in that topic. Each ϕ_k is a V-dimesional vector where V is the vocabulary size. Finally, the topic distribution vectors themselves, θ_d, are assumed to be drawn from a Dirichlet distribution and have a Dirichlet priors α, β associated with them.

The final model includes word distributions for each topic ϕ_k, topic distribution for each document θ_d, and topic assignments for each word $z_{d,j}$. Parameters $\Phi = \{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ and topic assignments z are traditionally updated via Gibbs sampling [2]. The likelihood [6] of a given set of documents D, with the word distribution and topic assignments for each word, is given by Equation1. It is a multiplication across all the documents and across all the words in each document. The two terms in the product are the likelihood of seeing these particular topics ($\theta_{z_{d,j}}$), and the likelihood of seeing these particular words for

this topc ($\phi_{z_{d,j},w_{d,j}}$).

$$p(N|\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{z}) = \prod_{d\in D}\prod_{j=1}^{N_d}\theta_{z_{d,j}}\phi_{z_{d,j},w_{d,j}} \quad (1)$$

IV. Data-sets

For this paper, we use two independent datasets to train and test our model using the features that are further discussed in Section VI.

A. Yelp Dataset

For this exercise, we use the latest Yelp dataset that is provided for the 2015 Yelp Dataset Challenge [12]. The dataset includes information about local businesses, reviews and users in 10 cities across 4 countries, the cities covered by Yelp dataset are as follows:

1) Phoneix, Arizona, USA
2) Las Vegas, Nevada, USA
3) Charlotte, North Carolina, USA
4) Urbana-Champaign, Illinois, USA
5) Pittsburg, Pennsilvenya, USA
6) Madison, Washington, USA
7) Montreal, Canada
8) Waterloo, Canada
9) Edinburg, U.K.
10) Karlsruhe, Germany

Overall, it contains $1,569,264$ reviews, $366,715$ users and $61,184$ businesses. There is also check-in information for each business, tips given by users, and a social graph of users consisting of approximately $2.9M$ edges. For our exploratory and analytical purposes, we only look at the users, reviews, and business information. As a first step, we look at the heatmap of the business distribution based on their geographical locations. As shown in Figure 2, most of the businesses are concentrated in Las Vegas and Los Angeles, as compared to other cities. Next, we describe how the data is formatted and provided by Yelp in the following subsections. This dataset is quite dense and structured, as opposed to Google dataset that is described in Section IV-B.

1) Users: Yelp dataset contains information of about $366K$ users in JSON format and for each user, we are given the following information:

```
{
    'type': 'user',
    'user_id': (encrypted user id),
    'name': (first name),
    'review_count': (review count),
    'average_stars':
    (floating point average, like 4.31),
    'votes': {(vote type): (count)},
    'friends': [(friend user_ids)],
    'elite': [(years_elite)],
    'yelping_since':
    (date, formatted like '2012-03'),
    'compliments': {
        (compliment_type):
        (num_compliments_of_this_type),
        ...
    },
    'fans': (num_fans),
}
```

2) Reviews: Each of the $1.5M$ reviews have the following properties associated with them:

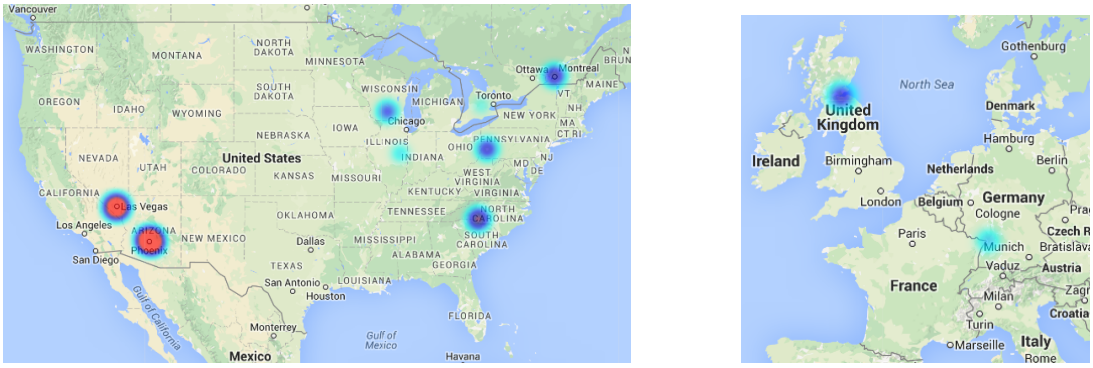
```
{
    'type': 'review',
    'business_id': (encrypted business id),
    'user_id': (encrypted user id),
    'stars': (star rating),
    'text': (review text),
    'date': (date),
    'votes': {(vote type): (count)}
}
```

3) Business: For each of the $61K$ businesses, we are provided with following information:

```
{
    'type': 'business',
    'business_id': (encrypted business id),
    'name': (business name),
    'neighborhoods': [(hood names)],
    'full_address': (localized address),
    'city': (city),
    'state': (state),
    'latitude': latitude,
    'longitude': longitude,
    'stars':
    (star rating, rounded to half-stars),
    'review_count': review count,
    'categories':
    [(localized category names)]
    'open': True / False ,
    'hours': {
        (day_of_week): {
            'open': (HH:MM),
            'close': (HH:MM)
        },
        ...
    },
    'attributes': {
        (attribute_name):
        (attribute_value),
        ...
    },
}
```

B. Google Local Dataset

Google Local dataset is provided to us by Prof. Julian McCauley. It mainly consists of information on businesses, users and users' reviews of businesses. The dataset contains a total of $3,114,353$ places, $11,453,846$ reviews and $3,747,939$ users. There are several challenges that we encounter while working with such a big dataset. Our main problem is the limitation in the computing resources we have. Another big issue that we encounter while working with this dataset is its sparsity and inconsistencies. To make it comparable to Yelp dataset and easier to work with, we heavily prune the dataset. Looking at the spread of the data, since there is a good number of businesses listed in California, USA, we decide to include only businesses that are located there, that have at least 3 reviews. Figure 3 shows the heatmap of the distribution of places that are used in this paper. As is evident from the figure,



(a) US distribution

(b) Europe distribution

Fig. 2: Heatmap of Yelp dataset

the majority of the places are concentrated near Los Angeles and San Francisco Bay area.

A lot of pre-processing needs to be done before the Google Local dataset can be used. First of all, the businesses' data does not contain the ratings. For each business, we read the Reviews data to find all reviews for that business, and use the average of them as the business' rating. Furthermore, unlike Yelp dataset, Google's users can give a rating score of one of the following: 0, 1000, 2000, 3000, 4000, or 5000. Therefore, we normalize it by dividing the rating by 1000, so that each rating is now in the range of 0 to 5, similar to Yelp dataset, and we treat 0 ratings as missing rating while computing the business ratings. Next we describe the format of the data.

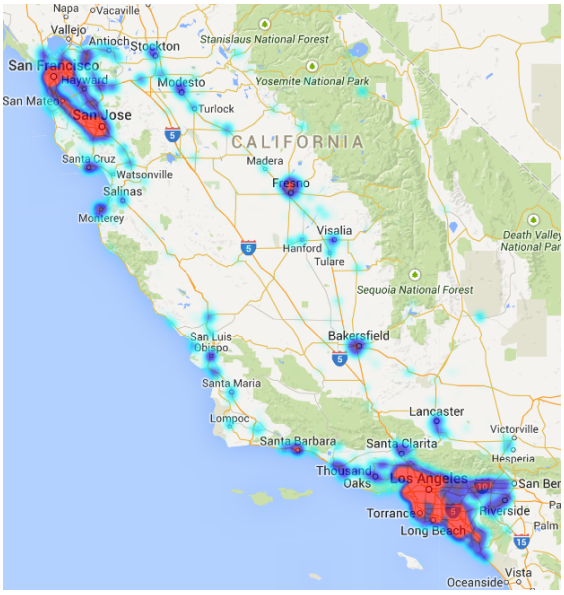


Fig. 3: Heatmap of Google Local dataset

1) Users: Every user has the following information:

```
id : {
     'userName': (username),
     'currentPlace': (Address),
     'education': (Mostly empty),
     'jobs': (Positions (user filled)),
     'previousPlaces': (previous locations)
}
```

One of the main limitations of this dataset is that most of the entries are very sparse. A lot of the fields are left blank. Apart from userId and userName, there is no other useful information that can be extracted out of this dataset.

2) Reviews: Each of the $11M$ reviews has following attributes:

```
<placeid>, <userid> : {
     'username' : (username),
     'rating' : (rating),
     'review' : (review text),
     'categories' : (categories),
     'gPlusPlaceID' : (placeid),
     'gPlusUserId' : (userid),
     'texttime' : (date of post),
     'utime' : (unix time)
}
```

An interesting difference from the Yelp dataset is the categories field. Yelp provides categories for each business, whereas Google Local provides it for each review. This means whenever a user writes a review for a business, the user has to also fill in the categories of the business itself. Therefore, it is harder to extract the true businesses' categories information from Google Local dataset, as opposed to Yelp where the business owners can specify the categories themselves when opening up the Yelp account.

3) Places: Places have following attributes associated with them,

```
<placeid> : {
     'name' : (place name)
     'hours' : (hours of operation),
     'phone' : (contact info),
     'closed' : (boolean place is closed),
     'address' : (place address),
     'id' : (placeid),
     'gps' : (lat, long)
}
```

V. Exploratory Analysis

Now that we have the raw datasets, we proceed to look at certain properties inherent in them. Since we are working with two different datasets, we only utilize the fields that are common to the both of them. First, we look at the distribution of the review counts of each business. Figure 4 shows the review count distribution for both Yelp and Google dataset. We can see that all businesses have at least 1 review, with the majority having less than 10 reviews. Secondly, we look at the rating distribution of the businesses as a histogram since the rating for both datasets lies in the range of 1.0 to 5.0. Figure 5 shows the histogram of the ratings across all businesses. We observe that Google users tend to give higher ratings compared to Yelp users. And we also notice that the number of reviews in Google dataset is much smaller than Yelp. This might affect the topic model since the data for training the model comes from the review text, and therefore having more reviews is usually preferable.

As mentioned previously, we want to look into the neighborhood influence on a business. Before we can do that, we want to check the observations that Hu et. al. make in their paper [5] and see if the same applies to our dataset. The first one is that most businesses have neighbors within a short geographical distance from their locations. Figure 6 proves the claim. It shows the fraction of businesses with at least 1, 3, 6, and 10 neighbors that are within a geographical distance threshold of 20, 50, 100, 200, 500, 1000, or 2000 meters. We can see that within 2000 meters, the majority of businesses have at least 1 neighbor. We use k-d tree implementation to easily query for the nearest neighbors of a given business. One thing to be aware of is that when talking about geographical location of a point on Earth, which is a sphere, we need to take their great circle distance. Ideally, we would like to use k-d tree that uses the Haversine formula[3] as its distance function. However, Python's scipy library already provides a k-d tree package[4] that uses Euclidean distance. Therefore, we use an approximation to simplify the k-d tree implementation, which we have verified to perform just as well as using Haversine formula. This approximation is done by projecting the latitude (lat) and longitude (lon) point into its corresponding 3D-Cartesian coordinate, as shown in Equation 2, where R represents the radius of the Earth in meter:

$$\begin{aligned} x &= R * \cos(\texttt{lat}) * \cos(\texttt{lon}) \\ y &= R * \cos(\texttt{lat}) * \sin(\texttt{lon}) \\ z &= R * \sin(\texttt{lat}) \end{aligned} \tag{2}$$

Another observation that Hu et. al. make is that there is a weakly positive correlation between the rating of a business and the average rating of its neighbors. They make this observation due to the phenomenon of "things of one kind come together", where people often associate certain regions to be good or bad based on the majority of the businesses in the area. Figure 7 plots the Pearson's correlation coefficient[5] between a business' rating and the average rating of its nearest neighbors of a certain distance threshold. If a business does not have any neighbors within the specified distance, then it is not considered in the correlation computation. As a comparison, we also compute the correlation coefficient between the business' rating and the rating of a randomly sampled business from the dataset. This produces a really small, near zero, correlation coefficient. On the other hand, the Pearson's correlation is higher when the distance threshold is low, around 20 to 50 meters. After 100 meter threshold, the correlation seems to be levelling off.

This weak positive correlation can be attributed to the fact that the majority influence actually comes from the intrinsic characteristics of the business itself. Even if a restaurant is in a great food district, if the food quality is not as high, then people will prefer to eat elsewhere. However, this still shows that there is some dependence between a business and its neighbors. We explore this further to see how strong the extrinsic characteristics (neighbors) can influence the business' rating prediction.

Since we use topic modeling to represent the intrinsic characteristics, we also observe the word count distribution of the reviews to get an idea on the length of each review text. Figure 8 shows the box plot of the distribution. Even though Yelp reviewers tend to give longer reviews compared to Google Local users, both dataset have a relatively good spread, with an average of 100 words for Yelp and 50 words for Google Local, per review text. Apart from this we also look at the top categories for businesses from Yelp dataset to get a sense of intuition what kind of topics should a topic model reveal, Table I shows top 10 categories and they align which a normal person's intuition about what popular businesses are being reviewed.

Top Categories
Restaurants
Shopping
Food
Beauty & Spas
Nightlife
Bars
Health & Medical
Automotive
Home Services
Active Life

TABLE I: Top business categories for Yelp dataset

VI. Feature selection

Based on our exploratory analysis of the dataset we decide to use a combination of the following features as our feature vector for training the model. For training LDA model we use a well know toolkit called Mallet [7] written by Prof. Andrew McCallum from University of Massachussettes, Amherst. We experiment with 4 different values of K (10, 25, 50, 100) for training our LDA model, Tables II and III show top 20 words of the selected 5 topics after training a model with 50 topics. We also tag these 5 topics with intuitive labels by observing the top words.

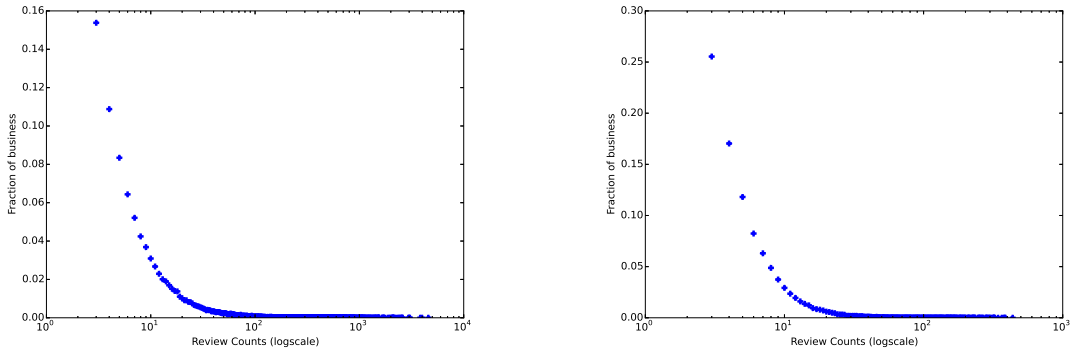
A. Business' Self Topics Vector

For our first feature, we use topic vectors of the business itself. After running LDA on the user generated reviews, we get

[3] http://en.wikipedia.org/wiki/Haversine_formula

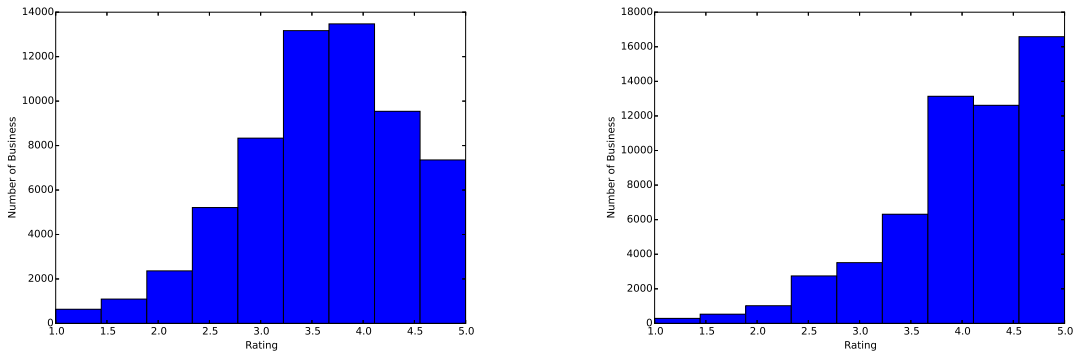
[4] http://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html

[5] http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient



(a) Yelp (b) Google Local

Fig. 4: Review count distribution over businesses



(a) Yelp (b) Google Local

Fig. 5: Review rating distribution over businesses

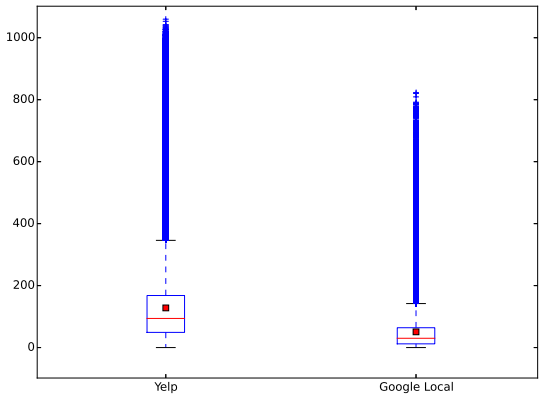


Fig. 8: Word count distribution for Google Local and Yelp dataset

a K-dimensional topic distribution vector $\boldsymbol{\theta}_t$, $\forall t \in \boldsymbol{T}$ where T is set of all review texts in the dataset. To compute the topics vector ($\boldsymbol{\theta}_b^{'}$) of the business ($b \in \boldsymbol{B}$) we take the average of topic vectors for all the review texts of that business ($\boldsymbol{T}_b$).

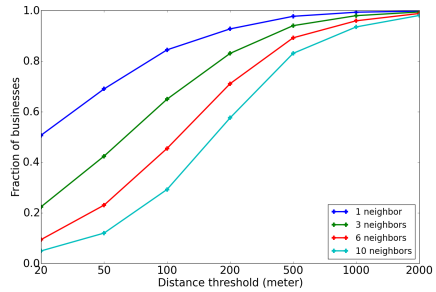
$$\boldsymbol{\theta}_k^{'} = \frac{1}{|\boldsymbol{T}_b|} \sum_{t \in \boldsymbol{T}_b} \boldsymbol{\theta}_t \tag{3}$$

The features, for each business b, are then the K-dimensional topics vector $\boldsymbol{f} = \boldsymbol{\theta}_b^{'}$.

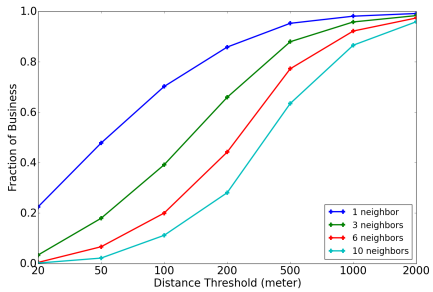
B. Average of Neighbors' Topics Vector

Another feature vector that we try to experiment with is the topic distribution of the nearest neighbors, instead of using the business' own topic vector. We obtain this distribution by averaging out the K-dimensional topics vectors of the neighbors. Given a business b and its nearest neighbors $\boldsymbol{N}$ queried from the k-d tree, assuming the topic vectors per business is already calculated using Equation 3 above, the feature vector for each business b is calculated as follows:

$$\boldsymbol{f} = \frac{1}{|\boldsymbol{N}|} \sum_{b^* \in \boldsymbol{N}} \boldsymbol{\theta}_{b^*}^{'} \tag{4}$$

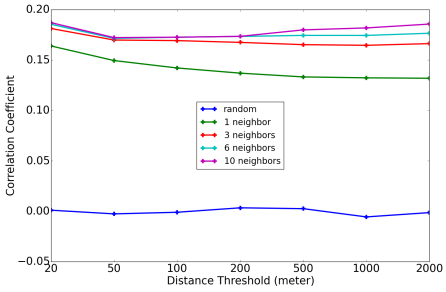


(a) Yelp

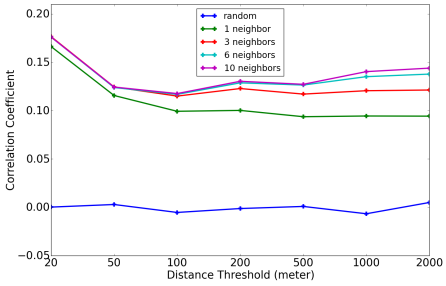


(b) Google Local

Fig. 6: Fraction of businesses with the specified neighbor count within some distance threshold



(a) Yelp



(b) Google Local

Fig. 7: Pearson's correlation between a business' rating and the average rating of its neighbors

C. Average of Neighbors' Rating

For our third feature, we use the average rating of the nearest neighbors of the business. From the k-d tree constructed, we can query the N nearest neighbors of the business, and calculate the average of their ratings as the business' rating prediction. Therefore, given a business b and its nearest neighbors $\boldsymbol{N}$, the rating prediction of the business (r_b) can be calculated as follows:

$$r_b = \frac{1}{|\boldsymbol{N}|} \sum_{b^* \in \boldsymbol{N}} r_{b^*} \tag{5}$$

VII. Models

In this section, we discuss the model that we pick, as well as the baseline model for comparison.

A. Baseline Model

For our baseline, we decide to use a simple prediction which uses global mean as the prediction. The global mean is calculated from the training set, which consists of 80% of the dataset that we randomly select. After we calculate the mean, we use this as the prediction on the test set, which consists of the remaining 20% of the dataset, and calculate the RMSE. Let R_t be the average ratings of all the sample in training set. Then, the predicted ratings for all businesses are R_t.

B. Stochastic Gradient Descent Regressor (SGD-R) Model

We explore 4 different parametric model based on the feature vectors selection that is mentioned in the previous section. Because this is a regression problem, we need a regressor that can train the models efficiently, taking into account that this problem has a relatively large number of training samples. We compare two regression solutions, namely the SGD-R and the Support Vector Regression (SVR). We compare both results on the model that utilizes the business' topics vector only, trained on Yelp dataset. We notice a small improvement in SVR, achieving an RMSE of 0.5868, compared to SGD-R's RMSE of 0.5985. However, this improvement of approximately 2% comes at a cost. The runtime before reaching convergence increases by a significant amount. SGD-R only takes around 2 minutes, but it takes roughly 15 minutes before SVR converges. Therefore, we decide to train the following 4 parametric models, on both Google and Yelp dataset, using only the SGD-R, since it is more scalable for bigger dataset. Furthermore, to avoid overfitting, we add an L_2 regularization parameter on our SGD-R models.

1) SGD-R Business' Self Topics Vector: Using only the business' topics vector, we train an SGD-R model on both Yelp and Google Local dataset. We try 4 different values for the number of topics, $K = [10, 25, 50, 100]$. In order to choose the best value for K, we run a 10-fold cross validation on the training set, and choose the K that gives the smallest average RMSE on the validation set. The resulting average RMSE can

"Arts & Entertainment"	*"Bars"*	*"Beauty & Spa"*	*"Nightlife"*	*"Fitness"*
music	bar	hair	club	kids
shows	drinks	cut	night	gym
stage	night	salon	dance	class
tickets	drink	color	music	park
great	place	time	vegas	classes
cirque	music	haircut	people	fun
vegas	bartender	great	girls	great
seats	fun	stylist	floor	yoga
amazing	people	job	drinks	water
fun	friends	dress	line	area
time	cool	appointment	party	play
venue	bartenders	wanted	pool	day
love	live	work	free	workout
audience	crowd	shop	fun	fitness
theater	loud	back	guys	people
good	cocktails	years	dj	year
watch	cocktail	style	crowd	studio
funny	beer	amazing	clubs	golf
show	atmosphere	recommend	table	equipment

TABLE II: Top words for selected 5 topics from Yelp dataset. The topic model was run with 50 topics. Due to space constraint we are only showing 5 topics

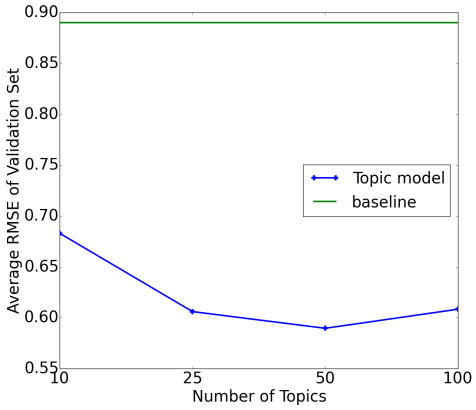
"Fitness"	*"Italian Restaurants"*	*"Airport & Rentals"*	*"Computer Repairs"*	*"Mexican"*
gym	food	san	computer	food
training	restaurant	francisco	store	mexican
fitness	wine	car	phone	tacos
classes	menu	airport	system	burrito
equipment	great	jose	buy	good
class	delicious	time	laptop	salsa
life	service	rental	apple	taco
great	dinner	driver	repair	chips
workout	dishes	service	problem	burritos
weight	excellent	bus	back	fish
ve	dining	shuttle	fixed	chicken
work	meal	taxi	pc	place
body	italian	trip	drive	delicious
yoga	experience	city	price	love
trainers	amazing	cab	data	fresh
people	wonderful	lax	fix	great
years	atmosphere	area	iphone	beans
feel	small	experience	screen	restaurant
instructors	decor	company	bought	asada

TABLE III: Top words for selected 5 topics from Google Local dataset. The topic model was run with 50 topics. Due to space constraint we are only showing 5 topics

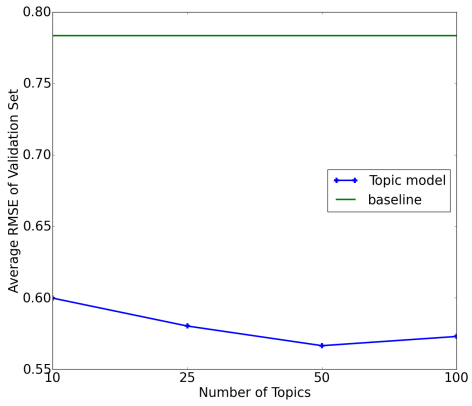
be seen in Figure 9.

2) SGD-R Average Neighbors' Topics Vector: We now want to see the effect of the extrinsic characteristics of a business. Using only the average of the business' neighbors' topics vector, we train an SGD-R model on both dataset. Since we mostly want to see the neighboring effects, we decide to use only 1 value for K, the number of topics. As can be seen in Figure 9, we choose $K = 50$ since it achieves the lowest RMSE for both dataset. On the other hand, we try different values for $N(1, 3, 6, 10)$ and $D(20, 50, 100, 200, 500, 1000, 2000)$, which are the number of nearest neighbors and the distance threshold in meters, respectively. Similarly as before, in order to choose the best value for N and D, we run a grid search on the different values, and run a 10-fold cross validation on the training set for each different combination value. We choose the N and D that achieve the smallest average RMSE on the validation set. The resulting average RMSE can be seen in Figure 10.

3) SGD-R Average Neighbors' Rating: Another model that we wish to explore is using only the business' average neighbors' ratings as its prediction. For this model, we also decide to pick $K = 50$ while varying the other parameters, $N(1, 3, 6, 10)$ and $D(20, 50, 100, 200, 500, 1000, 2000)$. The resulting average RMSE on the validation set, after running a grid search and doing the 10-fold cross validation, is shown in Figure 11

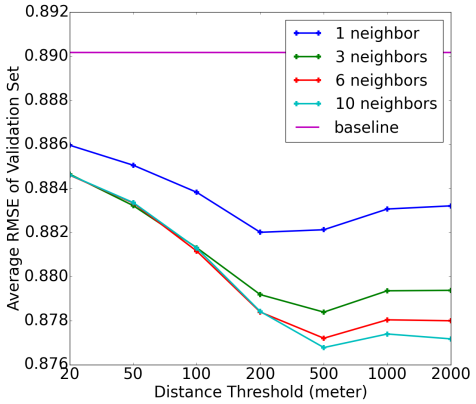


(a) Yelp

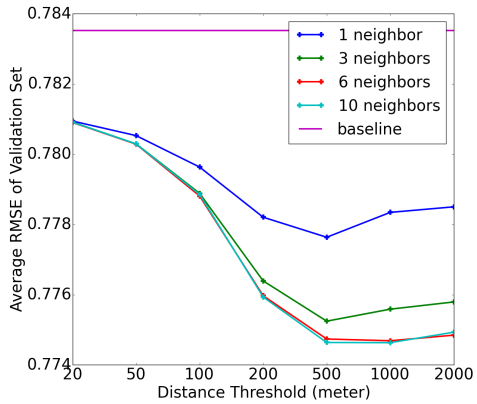


(b) Google Local

Fig. 9: Average RMSE on 10-fold cross validation, using the business' topics vector only.



(a) Yelp



(b) Google Local

Fig. 10: Average RMSE on 10-fold cross validation, using the average neighbors' topics vector only.

4) SGD-R Combination: Finally, we do a linear regression on the combination of the 3 different features vectors in our last model. By doing so, we hope that the features used in this model manage to incorporate both the intrinsic and extrinsic characteristics of a business. So for every business ($b \in \boldsymbol{B}$) given it's nearest neighbors $\boldsymbol{N}$, the final feature vector is given by Equation 6. It is a concatenation of self-topic vector, average of neighbor's topic vectors and average rating of the neighbors, for a total dimension of of $2K+1$, where K is the number of topics used.

$$\boldsymbol{f} = (\boldsymbol{\theta_b'}, \frac{1}{|\boldsymbol{N}|}\sum_{b^* \in \boldsymbol{N}} \boldsymbol{\theta}_{b^*}', \frac{1}{|\boldsymbol{N}|}\sum_{b^* \in \boldsymbol{N}} r_{b^*}) \tag{6}$$

VIII. Experimental Results

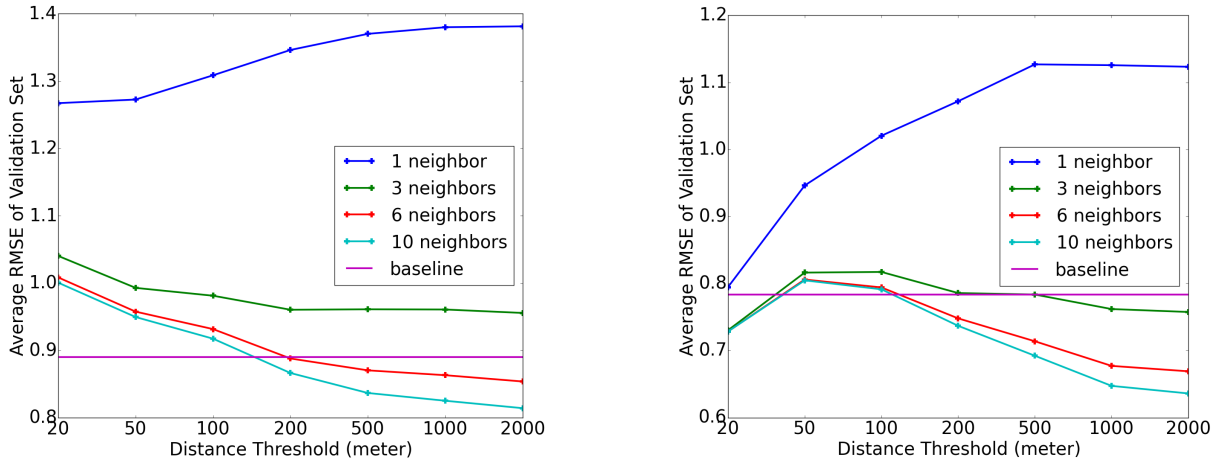
After running the 10-fold cross validation on the training set, we train our models on the whole train set, and calculate the RMSE of the trained model on the test set. Table IV reports the test set RMSE value of the different models. It shows that the SGD-R Combination model has the lowest RMSE, with an improvement of around 30% from the baseline that uses global mean as its predicted rating.

Model	Yelp RMSE	Google RMSE
Baseline	0.893673	0.780819
SGD-R SelfTopic	0.598529	0.564264
SGD-R NeighborTopic	0.878314	0.770883
SGD-R NeighborRating	0.811687	0.712748
SGD-R Combination	**0.596970**	**0.562559**

TABLE IV: Test set RMSE of the different models.

IX. Discussion

Through this exercise, we manage to get a better idea of how influential neighborhood effects can have on a business' rating prediction. From Table IV, we can see that even though SGD-R Combination gives us the lowest RMSE, it only differs slightly from SGD-R SelfTopic model, an improvement of only around 0.3%. Looking at this result alone, it shows that



(a) Yelp

(b) Google Local

Fig. 11: Average RMSE on 10-fold cross validation, using the average of the neighbors' ratings only.

the intrinsic characteristics are still the predominant factors in a business' ratings prediction. Using topic modeling on the review texts alone has already provided us a model that achieves an RMSE that is lower than the baseline, which uses global mean, by around 30%. Furthermore, SGD-R SelfTopic also far outperforms the SGD-R NeighborTopic and SGD-R NeighborRating models, which only lowers the RMSE by around 1% from the baseline model.

One thing that can possibly increase the neighborhood influence in our models is if we use a different feature vector to represent the extrinsic characteristics. As can be seen from Tables II and III, the topics vectors seem to reveal the latent categories of the businesses. In their paper, Hu. et. al. [5] observe that the weakly positive correlation between a business and its neighbors is independent on the categories of the businesses. Our result seems to match this observation. Therefore, it is possible to get a model with lower RMSE values by using a different representation of the extrinsic characteristics. One such alternative is by utilizing only the adjectives and nouns of the neighbors' review text, similar to what is suggested by Hood et. al. [4], since it can reveal the overall condition of the neighborhood.

Furthermore, there are other information provided in the dataset that we have not used yet, such as tips, check-in information or the social graph [10]. We can utilize them to come up with a more semantically cohesive features, or using some temporal analysis as done by Potamis [9] and Hood et. al. [4] to predict businesses' future performance. The social graph might also be useful in the rating prediction of a business, by taking into account friends who have reviewed nearby places.

In conclusion, even though our SGD-R Combination model does not perform as best as we hope, since the decrease in RMSE compared to the SelfTopic model is insignificant, we believe that there are potentials that can be further explored. Specifically with the Yelp dataset, since it has an abundance of other information that can be used to better represent the intrinsic and the extrinsic characteristics of each business.

References

[1] Bentley, J. L. Multidimensional binary search trees used for associative searching. *Commun. ACM 18*, 9 (Sept. 1975), 509–517.

[2] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *the Journal of machine Learning research 3* (2003), 993–1022.

[3] Fan, M., and Khademi, M. Predicting a business star in yelp from its reviews text alone. *CoRR abs/1401.0864* (2014).

[4] Hood, B., Hwang, V., and King, J. Inferring future business attention.

[5] Hu, L., Sun, A., and Liu, Y. Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval* (2014), SIGIR '14, pp. 345–354.

[6] McAuley, J., and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems* (2013), ACM, pp. 165–172.

[7] McCallum, A. K. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[8] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography 3*, 4 (1990), 235–244.

[9] Potamias, M. The warm-start bias of yelp ratings. *CoRR abs/1202.5713* (2012).

[10] Tiroshi, A., Berkovsky, S., Kaafar, M. A., Vallet, D., Chen, T., and Kuflik, T. Improving business rating predictions using graph based features. In *Proceedings of the 19th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2014), IUI '14, ACM, pp. 17–26.

[11] Wang, J., Zhao, J., Guo, S., and North, C. Clustered layout word cloud for user generated review.

[12] Yelp. Yelp dataset challenge. yelp.com/dataset_challenge, 2015.