Rating Prediction for Resturant on Google Local Data.

Yaqing Wang
Math Department
University of California, San Diego
yaw008@ucsd.edu

ABSTRACT

This project I explored many interesting topic in google local data set, like rating's realitonship with review number, time change, review number's realionship with review length, and positive and negative words in reviews. The task of rating prediction is focused on restaurant in google dataset. The algorithm involved with bias model, latent factor model and SVD++, and I compare difference in performance of the same model train by different way in the last part.

Keywords

Latent factor model, SVD++, Recommender System

1. INTRODUCTION

Recommender system provide option for users when they face large amount of products. It will not only save consumer's time, but also bring more profit for seller. It has two common methods to provide recommendation, collabation filtering and latent factor model. For this project, I decided to use the massive dataset from Google that contains information about places around the world, users with accounts in Google services and reviews that users have given to these places. I focus on places in US, and study many aspects of this dataset, like reviews, rating difference based on position, and distribution of these places in US. For the prediction task, I use Matrix Factorization Techniques to predict rating of places. Since category's effect, I chose to predict rating for resturants.

2. THE DATA SET

This dataset contains information about 3.7 million users, 3 million places and 11 million reviews that users gave to those locations. Each user's information entry is composed of a name, current place (city and GPS coor- dinates), level of education, jobs held, and previous places visited. Similarly, each place entry is composed of the name of the place, hours they open, phone number, address, and GPS coordinates that determine where the restaurant is located.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WOODSTOCK '97 El Paso, Texas USA

© 2015 ACM. ISBN 123-4567-24-567/08/06...$15.00

DOI: 10.475/123_4

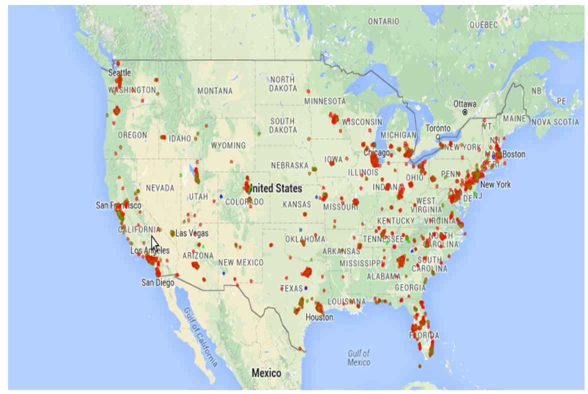

Figure 2: Rating distribution Map.

From places distribution figure1, most of them are in Japan, Europe, and US. In this project, I select data point whose places is located in US (figure 2). The challenges of this dataset are its large size, and its sparsity. So, I use streaming algorithm and take advantage of secondary sotorage to handle large size. For data cleaning, I remove the review include non-Ascii content and places outside US. I use gps to judge its location, so some places of Canada are included. Another challege is that goolge dataset's categeory of places is not in the places dataset but in the review dataset, so category is filled by users. The decriptions are not accurate. I select resturant for text mining and prediction task.

3. EXPLORATION OF DATA SET

3.1 Rating Distribution over Location

This is the an interesting topic to explore. I extracted American reviews from reivew and made a dataset only about America. Since very few reviews may have big variations, I set the threshold as ten reviews that only those exceed the threshold can get into the dataset. Finally I use color to indicate the rating. The rating increases with blue, green, yellow and red. They are marked on map fig 2 according to GPS data. I listed top ten rated restaurants on the map fig3 considering reviews number and ratings.

And I mark ten most fascinating place, considering review number and satisfied level.

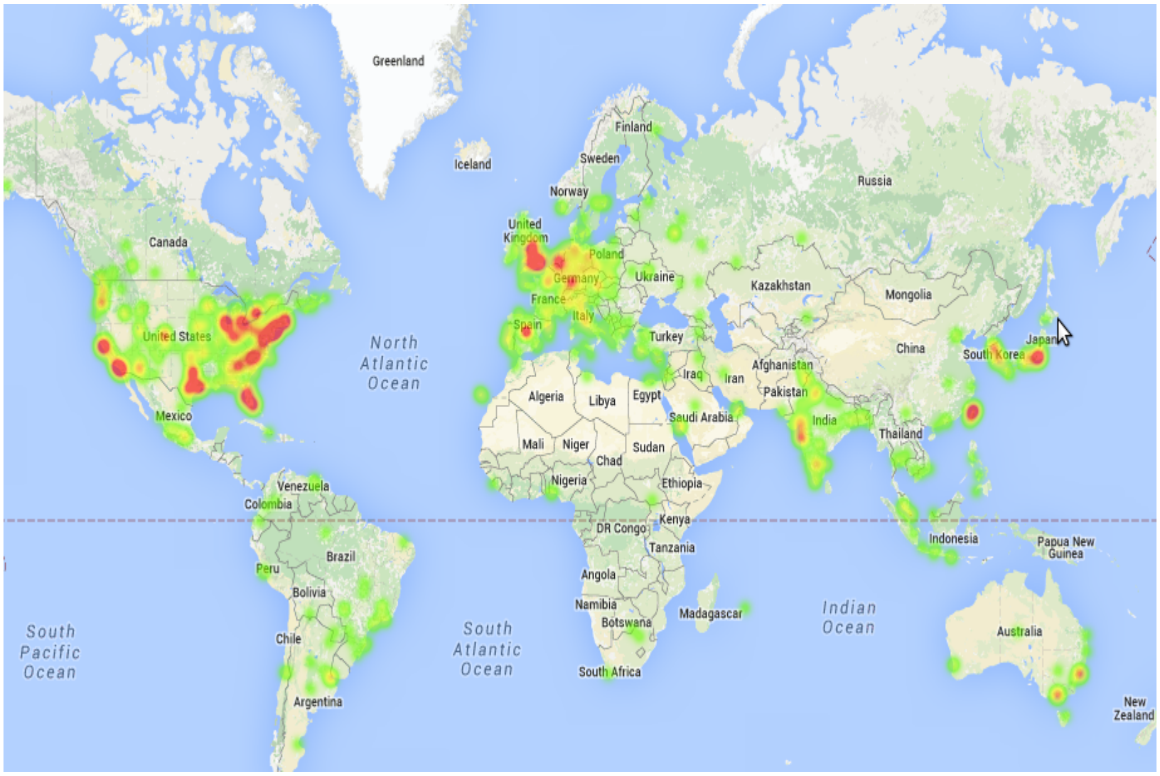

Figure 1: Worldwide Places Heatmap.

Figure 3: Top10 Business.

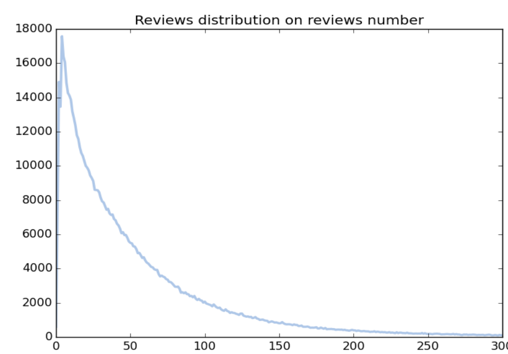

Figure 4: Reveiw distribution on Length.

3.2 Review Distribution over Length of Review

This is another interesting topic. At first I assumend the distribution is normal, however it can be seen from the fig4 that the reivew number increases sharply with word number at the beginning, then it decreases with exponential speed.

I plotted the logarithmic in figure5 and we can find it is nearly a line, so the decreasing exponentially strictly. The peak of review number is aorund 10.

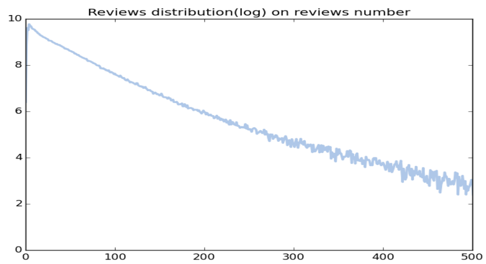

Figure 5: Log distribution on Length.

3.3 Rating Distribution over Length of Review

I extracted 1/10 dataset randomly and excluded the data without rating and review. Then I plotted the rating changes with the number of word. It can easliy tell that rating decreases with word number before 300 words, after that it does not have obvious trend. This may be because of number of reviews decrease quickly and rating is variarannt.

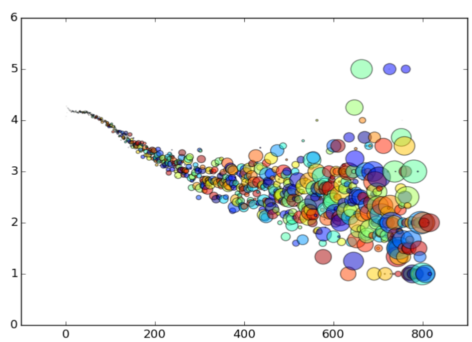

Figure 6: Rating distribution on Length.

3.4 Rating Distribution changes with time

Professor talk about time impact in recommender system. Reviews in Netflix had distinct change after review standard changing. So I want to see if reivews in Google have obvious change over time. After analysis, I found that many data lied in 2011, 2012 and 2013. And in those years, review amount distribute averagely among months.
So I take these three years as dataset, and plotted rating changes by days(fig7) and by month for three years(fig8). From the figure we can find a considerate increase from April, 2012. Then it declines but is still higher than the rating in 2011. In 2013, the rating keeps that level and does not change too much. So time impact is not obvious

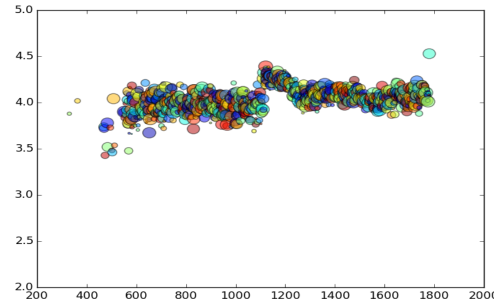

Figure 7: Rating distributio changes with time.

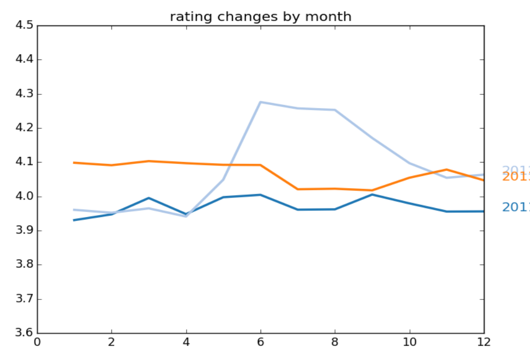

Figure 8: Rating distributio with month

3.5 Positive Words and Negative Words

I randomly took out fifty thousand reviews and made linear regression between word and ratings. Then I defined the fifty maximum theta words as positive words. On the contrary, I defined the fifty minimum theta words as negative words. From my result, this way does make sense. I scales them based on their weight and made word clouds. They can be seen that those words express obvious positive and negative tendencies.

Figure 9: Positive Words.

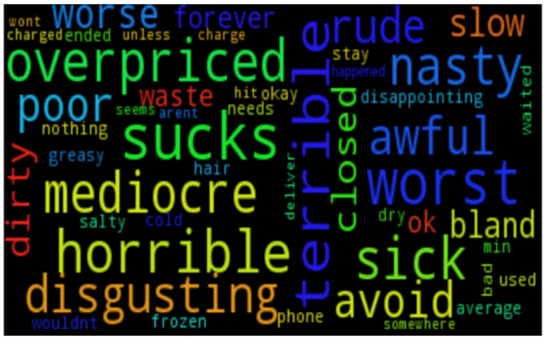

Figure 10: Negative Words.

4. THE PREDICTION

I random select 1,000,000 reviews in US business which has received more than ten reviews, and make sure their density. All of them own reviews, rate, placeId and userId. Then I seperate data set ranomly as 5 : 2 : 3, as train data set, validation data set, and test data set. Because they are random selected from a large data size, so the biggest challenge is cold-start problem. And I check the random dataset, 1/3 test data is warm-start, and 2/3 is cold-start. The cold-start problem'effect is so obvious, so I provide two result for cold-start and warm-start to compare model. I use item historical average rate and user historical average rate to handle cold- start. Since places has recived more than ten reviews. So I put place historical average rate as first option for cold-start. The performance is stable and good.

4.1 Task

In this section, we discuss the model that we pick, as well as the baseline model for comparison. I generate dense data by getting rid of the sparse data, and also considering on the number of ratings that a business has received. Number of reviews for a business is fatal to generate a stableable model. I remove places which the number of reivews is under 10. Ande for rating, I round it from 0

4.2 Bias Model

It is also the baseline model, it is simple but powerful.

$$r_{ui} = \alpha + \beta_i + \beta_u$$

r_{ui} indicates the rating that user u give item i, α is the average baseline, β_i is the bias of this item and β_u is the bias of this user. Since bias is big part of variance. So this simple model's performance is good. Also, we add the regularization terms to the opti mization problem as

$$min \sum_{u,i} (R_{ui} - \alpha - \beta_i - \beta_u)^2 + \lambda(\sum_u \beta_u^2 + \sum_i \beta_i^2)$$

We also need to use SGD to train the model and the update rule is as follows

$$e_{ui} = r_{ui} - \alpha - \beta_u - \beta_i$$

$$\beta_u = (1 - \lambda\sigma)\beta_u + \sigma e_{ui}$$

$$\beta_i = (1 - \lambda\sigma)\beta_i + \sigma e_{ui}$$

$$\alpha = (1 - \lambda\sigma)\alpha + \sigma e_{ui}$$

4.3 Latent factor model

$$r_{ui} = \alpha + \beta_i + \beta_u + \gamma_i * \gamma_u$$

γ_i and γ_u is muti-dimension vector, indicates user's preference for item's features. Also, we add the regularization terms to the opti- mization problem as

$$min \sum_{u,i} (R_{ui} - \alpha - \beta_i - \beta_u - \gamma_i * \gamma_u)^2 + \lambda(\sum_u \beta_u^2$$

$$+ \sum_i \beta_i^2 + \sum_i \|\gamma_i\|_2^2 + \sum_u \|\gamma_u\|_2^2)$$

we also need to use SGD to train the model and the update rule is as follows

$$e_{ui} = r_{ui} - \alpha - \beta_u - \beta_i - \gamma_i\gamma_u$$

$$\gamma_u = (1 - \lambda\sigma)\gamma_u + \sigma e_{ui}\gamma_i$$

$$\gamma_i = (1 - \lambda\sigma)\gamma_i + \sigma e_{ui}\gamma_u$$

$$\beta_u = (1 - \lambda\sigma)\beta_u + \sigma e_{ui}$$

$$\beta_i = (1 - \lambda\sigma)\beta_i + \sigma e_{ui}$$

$$\alpha = (1 - \lambda\sigma)\alpha + \sigma e_{ui}$$

4.4 SVD++

SVD++ includes implicit feedback, whether user bought the item. It performs very well.

$$r_{ui} = \alpha + \beta_i + \beta_u + \gamma_i(\gamma_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j)$$

Also, we add the regularization terms to the opti- mization problem as

$$min \sum_{u,i} (R_{ui} - \alpha - \beta_i - \beta_u - \gamma_i * \gamma_u)^2 + \lambda(\sum_u \beta_u^2$$

$$+ \sum_i \beta_i^2 + \sum_i \|\gamma_i\|_2^2 + \sum_u \|\gamma_u\|_2^2 + \sum_{j \in N(u)} |y_j|_2^2)$$

we also need to use SGD to train the model and the update rule is as follows

$$e_{ui} = r_{ui} - \alpha - \beta_u - \beta_i - \gamma_i(\gamma_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j)$$

$$\gamma_u = (1 - \lambda\sigma)\gamma_u + \sigma e_{ui}\gamma_i$$

$$\gamma_i = (1 - \lambda\sigma)\gamma_1 + \sigma e_{ui}(\gamma_u + |N(u)|^{-1/2} \sum_{j \in N(u)} y_j)$$

$$\beta_u = (1 - \lambda\sigma)\beta_u + \sigma e_{ui}$$

$$\beta_i = (1 - \lambda\sigma)\beta_i + \sigma e_{ui}$$

$$\alpha = (1 - \lambda\sigma)\alpha + \sigma e_{ui}$$

My fixed σ is 0.14 , λ is 1 and dimension of y and γ are 2. Because I want to compare different model's performance. So all the model has same parameter. The parameter has big impact when I use SGD, but for simplity, I did not tune it a lot. Just to be used for comparance.

4.5 SGD and ALT

Considering data size, I apply SGD to train model. Stochastic Gradient Descent(SGD) and Alternating Least Squares are both common ways to solve this kind of problem. But what's the difference in these two algorithms? I compare performance difference betweent these two ways. from efficiency and performance.

I first compare efficiency of two alogirthms. Obviously SGD is more efficiency. Considering data size, I use bias model which is faster to train to compare difference. The time of training model is relative with intial point. So I just crudely compare efficiency, SGD is better. Actually the question I am really interested in is difference in performance of two algorithms. I trained bias model by two ways. SGD parameter I chose is like above, $\sigma = 0.14$, $\lambda = 1$ and dimension of y, γ are 2.

Table 1: RMSE of different Model

Model	RMSE warm-start	RMSE cold-start
Bias (SGD)	1.0729	1.3130
Bias(ALT)	0.7134	0.8512

Considering SGD which is relied on tuning parameter, this difference is still huge. The ALT has better performance in traing bias model, but it is not effieciency. Restricted by time and my computing source, I still apply SGD, and compare different model based on same traing way.

4.6 Need to be Impoved

Restricted by time, I did not dig a lot into how to shape a new algorithm to solve this problem. But I have some ideas. The neighbour incoorporate with svd++ is a good idea. I try to define purchase network between item and user as virtual social network. This network is not stable as real network, but it is based on similirities and latent logic behind purchase. I still have many problem to be solved. How to define this similirities, whether this relationship can be transfered and what's the decay rate in this process if transfere. This is an interesting product and still have some future work to do.

5. CONCLUSION

In this project, I explore the intersting problem of goolge data set and use bias, latent-factor and SVD++ to make predications for rating. The model has relatively great perfomance, and I can not deny this good performance is based on density data point I choose. The final result is as follows.

Table 2: RMSE of different Model

Model	RMSE warm-start	RMSE cold-start
Bias (SGD)	1.0729	1.3130
Bias(ALT)	0.7134	0.8512
Latent factor model	0.7378	0.8682
SVD++	0.6784	0.8032

SVD++ has best performance. But bias based on ALT's performance is impressive. I recalled the bias's great performance in assignment 1. Now I know that's because of different traing way. How to train this model efficiently and well is a interesting problem to be explored

6. REFERENCES

[1] Yehuda Koren *Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model.* In ACM KDD, 2008

[2] Longke Hu, Aixin Sun, Yong Liu. *Your Neighbors Affect Your Ratings: On Geographical Neighborhood Influence to Rating Prediction* In ACM SIGIR, 2014.

[3] Jaewon Yang, Julian McAuley, Jure Leskovec *Community detection in networks with node attributes.* International Conference on Data Mining