Discover new places using Location Recommendation System

Uma Nataraj
Department of Computer Engineering
San Jose State University
Student-Id: 013727259
San Jose, CA

***Abstract*—Location based services are exploiting user data to an extent which actually helps other user explore a new location. The void between visitor's lack of knowledge about a place and user's interest to know about a place is filled by using Social media networks. Social media networks are helping users to develop knowledge about the place that they have not visited. This network also includes details of user preferences, visitor behaviors for various locations. Analysis and aggregation of the data generated by such social networks is useful in predicting users' preferences on an unvisited location also by building user friend network. In this project we aim to recommend location to a targeted user using collaborative filtering approaches: Matrix Factorization without friend network influence and matrix factorization with friend network influence. We evaluate the two methods using precision@K metrics and develop a web UI which runs on Google Cloud and uses HTML/Flask demonstrate the results of both approaches.**

I. Introduction

SOCIAL network applications like Facebook, Foursquare, Gowalla has helped their peers discover a new place because of the check in data updated on the website by their users. Check-in data is becoming popular these days which is defined as an update given by a user when he visits a new location such as park, night club, restaurant etc. User check-in data has helped significantly to ameliorate location recommendation system. This data can be used by other users to discover a new place or advertisers to provide specific advertisements for recommending a location. User check-in data has also helped data scientists to know more about user interests, pattern of preferences etc.

Recommendation systems play an important role in providing successful recommendation to a user. Recommendation systems will explore data of registered users such as places visited, number of times the user visited the location, user check ins, friend's network. Users generosity in sharing all details in the social network website is attracting more users to exploit the website. It will also reveal multiplicity of invisible cities. In this paper we explore Matrix Factorization and implement improvements via user-friend social network data to recommend location to user. Matrix factorization is a popular approach that uses collaborative filtering. Collaborative filtering exploits the friend network in order to recommend a location to the user. CF aims at finding similarity between two users and based on the similarity score it will predict the rating that user might give to that location. Location will be recommended based on the predicted rating.

This paper collects data for about 407533 users from a location based social network website Gowalla. This dataset contains information in the comma separated values format. This dataset also entails information about user, users' friends' network, location details such as its latitude, longitude and user check ins data. The raw data that we have obtained from Gowalla dataset cannot be directly used in the project and hence data preparation is an important step in this project. Data preparation includes checking for missing values, checking for valid data and combining data that can be directly used in the code. User-Location matrix will make use of this processed data for recommendation.

This paper helps in understanding how to recommend an unvisited location to a user. The goal of this project is to recommend new location to users using matrix factorization methods and also build user friend network to explore its influence. Post introduction talks about data preparation in detail, followed by approaches used in this project, implementation, technical challenges faced, experimental results and conclusion.

II. Data Preprocessing

Data preprocessing is a data mining technique that converts raw data into understandable format. The data we obtain from the real world is inconsistent, lacking certain behaviors, certain trends, certain attributes and it might contain errors also. Data preprocessing plays a significant role in making things easier with respect to data. There are few important steps that needs to be followed in data preprocessing: Import the libraries – In this paper we have used numpy, pandas, scikit, networkx libraries. Data is imported from Gowalla social network website. This dataset consists of 407533 total users. The total number of locations recorded in the dataset is 2844145 and total number of check-in recorded is 36001959. Checking the missing values has been split to many steps. The raw data which was obtained contained those users who did not have any check-in history, locations which has no check-in history, few attributes contained none values, users who did not have any friends, users whose friend's network is smaller etc.

	No of users	No of locations	No of check-ins
Data before pruning	407533	2844145	36001959
Data after pruning	145823	277043	2145337

In order to have clear vision on our data, our goal is to find those users and locations which got continuous check-ins. We have filtered the attribute values in such a way that we have retained the user count who has made more than 25 check-ins. Similarly, we have retained those location ids which has got more than 25 check-ins.

This is followed by finding those users in the above filtered data set who have done more than 25 check-ins in locations which has got more than 25 check-ins. The data before and after pruning is as shown in the box above.
Noisy data has to be cleaned as meaningless data cannot be interpreted by machines. It might be generated due to various reasons such as data faulty collection, data entry errors, data loss etc.

The data after pruning which is clean data is directly used by our code to generate recommendations. In this paper we are going to approach matrix factorization, build users friend network and recommend based on his friend's user check-ins and also recommend those places which is not visited by the user.

III. Friend Network

Gowalla dataset also includes social network data of users. We have built directed graph of the network using networkx library in python. Friends have enormous influence on the likes and dislikes of a person, so using this network to filter the collaborative results should yield in a better recommendation quality of this system. Graph analysis of the friend network yields the graph is sparse, but the breadth first search reveals most of the users are connected via several edges.

```
# friends network stat
print("Nodes = ",DG.number_of_nodes())
print("Edges = ",DG.number_of_edges())
```

```
Nodes =  145040
Edges =  1832987
```

IV. Methodology

We have approached this problem using Collaborative filtering. This is one of the most popular approaches that is used for recommendation systems. Collaborative filtering exploits the user's ratings given to items. This is executed based on the assumption that user who liked an item past will like it in the future, which can be concluded as user preference will stay the same over a period of time. Collaborative filtering does not consider into account about any user preferences, it only considers user's past behavior. There are two categories of collaborative filtering namely, User based, and Item based.

User based can be used calculate similarity score between the user and other users. There are many ways to find similarity between users such as, Euclidean distance, Manhattan distance, Minkowski distance, Cosine similarity and Jaccard similarity. To recommend a product that is not seen by the user, we calculate similarity between other users who has rated that unseen product and based on the similarity score we can predict whether to recommend that product or not.

Item based can be used to calculate similarity between items. An unseen product that is to be recommended to the user, unlike user based where we will calculate similarity between users, we calculate similarity between the items that is to be recommended and other items already rated by the user. The above-mentioned list of similarity measures can be applied to calculate similarity between items. Based on the similarity score we can predict the rating that user might assign to that product.

We observed a few challenges while using standard collaborative filtering methods, the two biggest ones are data sparsity and scalability. Out of various methods of collaborative filtering to resolve the challenges involved, in this paper we have explored one of the popular methods that is Matrix Factorization. This is one of the widely used method to solve the data sparsity and scalability problem in user-item matrix. The user item matrix is decomposed into user matrix and item matrix, which is as shown below,

$$M = U \times V^T$$

The above matrix can be decomposed as $U \times K$ and $K \times V^T$ matrices, where K is the number of latent factors. Matrix factorization might help in increasing the prediction quality because we are filtering out noisy data and detecting nontrivial correlations in the data. With the help of matrix factorization method, we top five or ten locations that user has not visited. We use Singular value decomposition (SVD) approach to keep the difference between the actual and the predicted rating as low as possible. SVD decomposes the matrix into three matrices user matrix, item matrix and a diagonal with all positive entries.

The next approach that we have used in this paper is to build user friend network. The Gowalla dataset provides the list of friends of all users. By exploiting this data, we can build user friend network. This also helps the targeted user discover the places that he has not visited. When a user's friend shares his experience about a location such as reviews, ratings etc., the targeted user who has no knowledge about that particular location. Therefore, User friend network helps the targeted user explore an unknown location.

V. Implementation

We have implemented two methods of collaborative filtering approach. First one is using simple matrix factorization and second one using user friend network matrix factorization. Both of the mentioned approaches exploit historical user check in data.

According to literature, matrix-factorization based approaches are the most successful and widely used recommendation methods. Simple matrix factorization can be defined as m users x n location matrix. For a target user, our responsibility is to recommend set of locations that user might be interested in but has not visited the place. This can be done by introducing a matrix M to describe user's rating over each location. The ith element in the matrix M describes the rating of user$_i$ over location$_i$. If user$_i$ has checked in to that location at least once we set the value of ith element in the matrix M I set to 1, otherwise the value of the ith element is set to 0. If the ith element value is set to 0 in matrix M doesn't conclude that user is not interested or it is against his preference, it can be concluded as user$_i$ has no knowledge about location$_i$. With the help of matrix factorization, we can predict and recommend user$_i$ about location$_i$. The matrix M is decomposed into two different matrices namely user matrix and location matrix. User matrix is composed of latent factors K cross number of users. On the other hand, the location matrix is composed of K latent factors cross number of locations. Once we get these two matrices and multiply these matrices, we will have a complete user-location matrix. This step is crucial because this helps in resolving the missing values issue which will cause the matrix to be sparse. After obtaining the values of multiplication, we can reverse sort it and recommend set of locations to the targeted user.

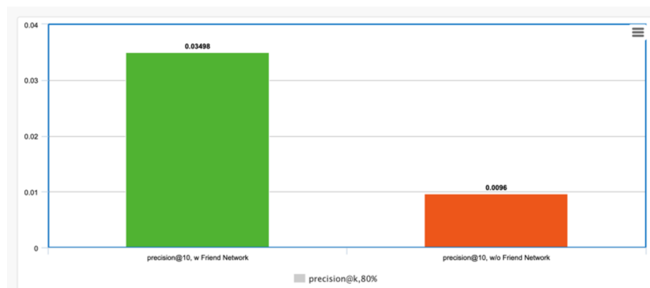
User Friend network matrix factorization helps in improving the recommendation system to a targeted user. Similar matrix factorization method is followed in this approach the only change is that, the targeted user will get recommendations based on the places visited by his friends. User-location matrix is decomposed into two matrices user cross K latent factor elements and K cross locations matrix. The multiplication of these elements will help in removing missing values and replacing it with the predicted values. These predicted values will be generated based on friend network only. Gowalla dataset entails details about the user's friend list and the user will be able to see the experiences shared by his friend. Exploiting this data will help programmers classify if the user will be interested to explore the location or not. Now the top picks for the user which is provided by recommender system will now search in friends list which is easier to run and faster. The matrix multiplication of the two decomposed matrices will help us predict the rating that will be given by the user to that particular location.

Evaluation method

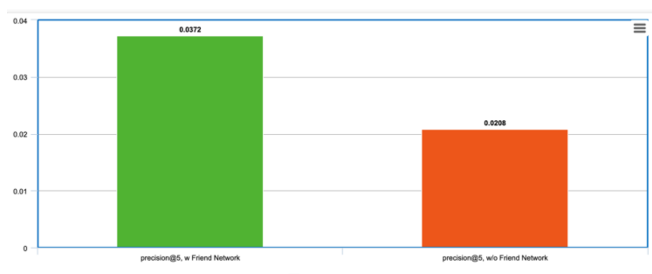
Popular methods for classic evaluation metrics are Precision and Recall in binary classification algorithms. Using this type of metrics have been translated to help evaluate our recommendation systems. Translation is based on the assumption that that true rating above a value for example 3.00 corresponds to a relevant item and any rating below this value is irrelevant. There are multiple values to set the threshold value such as considering the user's history of ratings. The evaluation method that is used to predict the quality of rating and prediction in this paper is precision@k. Precision@k can be defined as the proportion of recommended items in the top-k set that are relevant. Mathematically precision@k is defined as follows,

Precision@k = (number of recommended items @k that are relevant) / (number of recommended items @k)

In this paper, we have implemented the recommendation using two approaches matrix factorization and user friend network matrix factorization. We have calculated precision@k results for k=10 and k=5. We observed the results for precison@10 for matrix factorization without friend network is 0.09600 and precision@10 for matrix factorization with friend network is 0.03498 which is relatively high precision which is obtained by taking advantage of building friend network. Similarly, results for precison@5 for matrix factorization without friend network is 0.02080 and precision@5 for matrix factorization with friend network is 0.03720 which is relatively high precision with the help of friend's network.
The figure below demonstrates the quality of the prediction using precision@10,



The figure below demonstrates the quality of the prediction using precision@5.



VI. Technical Challenges

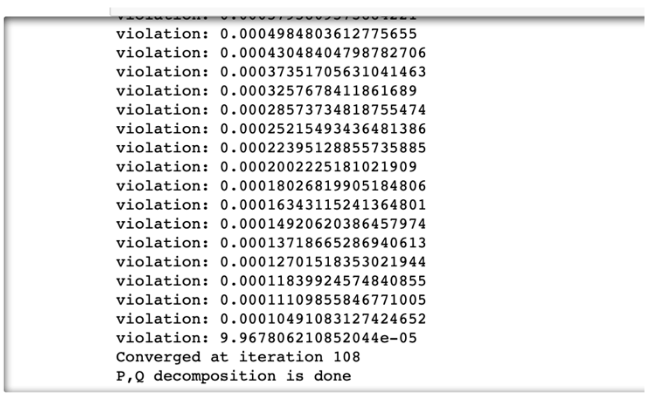
This part of the report will talk about the problems faced during the implementation of this project.

- Our data set contained of approximately 400,000 users in total. Generating user-item matrix for this large amount of data set was very time consuming. Matrix factorization convergence was not faster either because of the large amount of user data. We did copy all the results to a .csv file every time we ran our matrix factorization code for our recommendation output and analysis of output.
- Web UI – Appspot was not able to process file

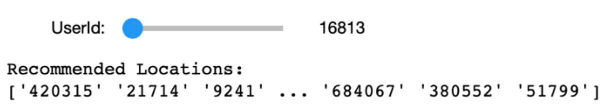
size more than 30MB. We tried splitting the whole preprocessed data into smaller chunks of file size 30MB. Therefore, for the UI we reduced the dataset size to lower size for UI interpretation and faster execution of the code and output. For the UI that we have presented, it has approximately 14,000 user's data being processed to provide output.

VII. EXPERIMENTAL RESULTS

This topic speaks about various steps and results involved in obtaining the final output. The below picture will demonstrate basic decomposition of matrix M.



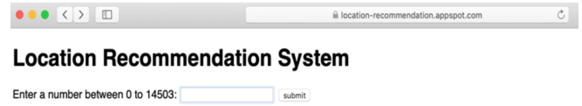
The user-id can be chosen with the help of the sliding button, so that it will recommend set of locations for the targeted user is as shown below,



The web UI console professionally demonstrates the actual output which includes details of output obtained by using friend network, without using friend network and user's friend's list. It is discussed in detail in Web UI section.

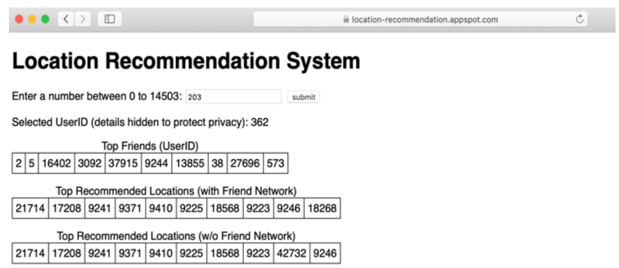
VIII. WEB USER INTERFACE

The user interface is built for displaying results in an interactive manner so that the person using the UI can get good exposure of location recommendation system. The UI for this project will allow the user to enter the user-id and the output display the user's friends list, list of locations recommended with the help of user friend network and without user friend network. The web UI is as shown in the picture below,



The above picture demonstrates the basic interface with an option for the user to enter any user-id between 0 to 14503.

The picture below will demonstrate the output shown with all the details regarding the targeted user which includes the top location recommendations and details of friend network.



CONCLUSION

We implemented two versions of Matrix Factorization including friend network influence. Successfully implemented matrix factorization method to recommend locations to a targeted user and help him explore his neighborhood, unvisited locations. This is accomplished by exploiting the user friend network and without exploiting user friend network. The evaluation favors the results got by exploiting user friend network (higher precision). Dataset collected from Gowalla is preprocessed by picking those locations and users who have more than 25 check-ins. Generated data graphs to have clear vision and understanding of data. Approaches to recommend a location is explained in detail. Simple UI using iPywidgets in Jupyter and web UI using HTML/Flask is implemented. For future work and better quality of the predicted rating can be improved by exploring other recommendation models and other approaches to collaborative filtering.

ACKNOWLEDGMENT

I would like to thank Professor Shi Yu Chang and Teaching Assistant Mr. Surya Sonti for guidance and support in executing this project successfully.

REFERENCES

[1] Dataset *https://snap.stanford.edu/data/loc-gowalla.html*
[2] *http://www.ntulily.org/wp-content/uploads/conference/Personalized_Point-of-Interest_Recommendation_by_Mining_Users_Preference_Transition_accepted.pdf*

[3]
https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=4772&context=sis_research

[4] C. Cheng, H. Yang, I. King, and M. R. Lyu. *Fused matrix factorization with geographical and social influence in location-based social networks. In Proceedings of the 26th AAAI, 2012*

[5]
https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/download/4748/5113