An Intelligent Recommendation System Based on Collaborative Filtering and Grid Structure

Animesh Chandra Roy(✉) and Mohammad Shamsul Arefin(✉)

Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh
sarefin@cuet.ac.bd

Abstract. The rapid growth of location-based networks during the twenty- first century has greatly increased, so there is a need of providing suggestions to personals about their interest activities. Nowadays, location-based social networks (LBSN) become a common platform for users to share interests. In this paper, our main concern is to design a recommendation system that will provide suggestions to the user according to their interests. We have developed a framework based on Collaborative Filtering (CF) that analyses user activities to find the similar user. CF helps us to enrich each user profile by rating unvisited places which we can include in their interest hierarchy. Then we calculate the similarity of the user profile with the Point of Interest (POI) extracted from the user's current location and make recommendations. Here Grid Structure is used to analyse the POIs extracted from Google.

Keywords: Recommendation system · Collaborative Filtering (CF) · Grid structure

1 Introduction

Information Retrieval (IR) technologies have gained outstanding prevalence in the last two decades with the explosion of massive online information repositories. Recommendation system (RS) is one of the most effective and efficient features that has been initiated by the LBSNs. Based on the user's interest this system can recommend the users about different places and activities which are matches their interest [7]. An efficient and accurate recommender system is very much needed in this highly social network era because of the rapid improvement of Internet Technology. This system collects the user information which can include location, entertainment, activities, games, and traveling destination, etc. based on their choices. After considering this information Recommender systems provide a recommendation to the user. This prediction reflects on the user's preference history. These recommendations make the Social network more strengthen because it connects one to another through their common interest. The recommendation can also provide a suitable place, where a group of friends can meet together according to their preferred type of places and previously tagged in places. As RS considers the user profile, it calculates the similarity between different profiles and suggests the best

R. Misra et al. (Eds.): ICIoTCT 2020, AISC 1382, pp. 182–192, 2021.
https://doi.org/10.1007/978-3-030-76736-5_17

matching. In this paper, we have developed a framework based on grid structure to make a suggestion, in which each recommendation is generated after calculating the similarity of users categorized preferences and geotagged places.

1.1 Motivating Example

Because of the availability of modern GPS technology people tends to use it in their daily life to make it easier. Most of the existing collaborative filtering methods compare between the similar user and taking into account their check-in spots to provide recommendations. Whenever a user comes to visit a new place, it is necessary to suggest some POIs according to the user's interests. Since the existing works do not consider the semantic analysis of each user's interests[10]. By analyzing their choices we can easily avoid irrelevant places to recommend.

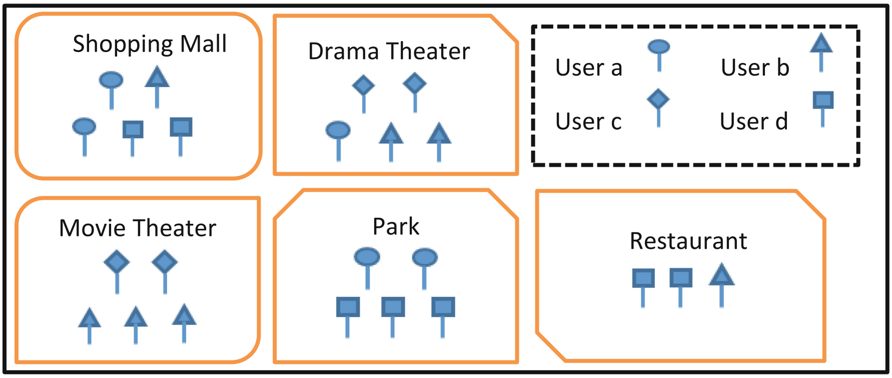

Fig. 1. Users check-in points

Now, consider four user's check-in points in Fig. 1. Here each user's check-in is pointed as a point of interest (POI) and their visiting summary is presented by Table 1. Here, user b seems like to visit 'Movie Theater' and 'Drama-theater' a lot. So our system analyses the behavior and suggests an 'entertainment' type place when the user comes to a new visiting place. Thus it can avoid unnecessary suggestions that do not match with users interest. Our proposed method works as follows:

- At first, we create a user profile based on their previous check-in history and make the hierarchy of their choices.
- Next, we extract all the possible POIs from the user's current location and divide them into grids.
- Then we measure the similarity score between the user's profile and extracted POIs from the grid.
- Finally, we select top-scoring POIs to suggest with the shortest distance.

Table 1. User check-in information

Place	User a	User b	User c	User d
Shopping mall	2	1	0	2
Drama-theater	1	2	2	0
Movie theater	0	3	2	0
Park	2	0	0	3
Restaurant	0	1	0	2

The rest of the paper is described as follows: In Sect. 2, some related works have been discussed. In Sect. 3, there are some preliminary topics related to our system. In Sect. 4 briefly we describe our framework. Section 5 describes the dataset and implementation of our system. Finally, Sect. 6 concludes and gives an idea about future research.

2 Related Works

Ashbrook et al., [1] proposed a predictive system to locate the future movements of the users. It uses Global Positioning System (GPS) to collect location data and other information such as other people's presence. These data are then clustered into different scales and then the Markov model introduced to incorporate those locations for prediction. Herlocker et al., [2] they reviewed some key metrices for evaluating a recommender system such as RMSE, Precision, Recall, Prediction-rating correlation etc. Adomavicius et al., [3] described filtering methods into three categories: content-based, collaborative, and hybrid recommendation approaches. They also showed some limitations and future extensions of a recommender system. Horozov et al., [4] introduced an enhanced collaborative filtering method using the location parameter generating recommendations. Park et al., [5] introduced a personalized recommendation system based on Bayesian Networks (BN), which uses the user's location, surrounding context, and time. Considering user requests this map-based system provides services by displaying onto the minimap. Chow et al., [6] provides a comprehensive system that covers three dimensions of location-based services such as, newsfeed, news ranking, and recommendations. They work on the designing of location and rank aware query operators, materializing query answers, and providing privacy-aware query processing.

In [7, 8, 10] used user-based CF method and a hierarchical-graph-based similarity measurement (HGSM), is introduced to create a model on each user's check-in history, and measuring the similarity of check-in activities between users. This framework is built on three factors, i) the sequence data of an individual's outdoor movements, 2) the volume of a visited geospatial region, and 3) the ordered data of geographic spaces. They used a content-based collaborating filtering method to find the suggestions. Lee et al., [9] used a semantic approach to measure the similarity using the location. In [11] a machine learning technique is used for predicting a user's location. The dataset consists of the user's points of interest (POI) or venues based on their social activities and interests. To solve data sparsity, they proposed a Probabilistic Neural Network which gave better results considering the other two types of neural networks. Mu et al., [12] proposed a collaborative method to provide service recommendations based on the service properties. At first, this framework builds a preference model of service property based on each user's information. Then, measures the service similarity score of two services of each user. Finally, the Pearson correlation coefficient of the similarity score of two services shows the final result. In [13] a novel approach was introduced by using a user-location vector to represent the relationship between user and POI. Liao et al., [14] proposed a strategy for recommendation using tensor factorization. At first, the user's information is extracted using Latent Dirichlet Allocation(LDA) to generate a probability distribution for the extracted information. Secondly, the user's check-in information is separated into a different category. Finally, the singular value decomposition (SVD) algorithm is applied to for POI recommendation. In [15] they proposed a recommendation after considering user rating and item attributes. Here, a weighted control coefficient is used to find the nearest neighbors.

3 Prelimineries

3.1 Check-In Information

Check-in information defines those physical places where users usually visit more often and share their feelings. These days online social services, such as Instagram, Facebook, and Tripadvisor provides such kind of check-in option for user using a mobile application. Global Positioning System (GPS) allows us to locate those places.

3.2 Point of Interest (POI)

Point of Interest (POI), defines the categorical partitioning of all the user's check-in information. Each check-in data is classified into a specific category presented by a POI. Mainly some GPS oriented software, GIS technology uses the POI concept. POI can be defined by geometrical coordinate as the latitude and the longitude of a specific place.

Table 2. POI data

Location	Rating	Co-ordinate	Category
Banani coffee house	4.0	<23.777176, 90.399452>	Food
Chittagong complex	3.75	<22.341900, 91.815536>	Shopping
Shapla movie theater	4.25	<25.744860, 89.275589>	Entertainment

For example, in Table 2, it shows some contents of a POI database. Here POI database contains location, co-ordinates, rating, and category of this information. The proposed method intended to extract the category of each user and find similarities with the new location. The similarity will help to recommend new places for the users to visit.

4 System Architecture and Design

Our proposed framework of the recommender system consists of three main parts: finding user interest, measuring similarity, and generating recommendations. Recommendation generation section divided into two components: recommend to a single user and for multiple users.

4.1 Discover Interest Using Density-Based Spatial Clustering

To identify users' interests, we must study on check-in places.

Nowadays, people used to visit a variety of places for different reasons. Some places like a movie theater, park, restaurant are visited with a lot of interests. Some of them are visited for regular activities such as colleges, universities. More often people rarely visit a place with interest becaues of its accessibility.

We used the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm for clustering the check-in spots. It makes clusters with high-density regions which are separated from one another. In figure, P1, P2, : : :, P8 are the clusters. Then we locate the gravity center for each cluster and store it in the user's table. Any spot that is closest to the center of a cluster named CAFE will be categorized into Food.

Two parameters (ε, *MinPts*) are necessary for DBSCAN: ε defines a value if one point can be considered as a neighbor and *MinPts* defines a minimum number of data points within ε radius. A point with more data points than ε is a core point and the least data point is a border point for a cluster. Steps for DBSCAN:

- Select an unvisited point and extract all data points within ε, and find the core point.
- Form a cluster with the selected points if they are not assigned to any other cluster.
- Recursively add density connected points until a border point is found and create a new cluster. The point which doesn't belong to a cluster is a noise.

Then, a Semantic Hierarchical Category-graph Framework (SHCF) [10] was applied shown in Fig. 2. SHCF consists of three layers, clusters from DBSCAN are in the bottom layer, the middle and top layer generates the general category of the POIs. From the figure we can see that, in "layer 1", these are check-in clusters. In "layer 2", these are the sub-type category nodes. And, the top "layer 3", contains the super-type interest categories. Thus SHCF generates each user's activity profile based on their interests. Whenever a user comes to a new place, this framework helps to find the best suitable suggestions for the user.

4.2 Collaborative Filtering (CF)

Collaborative filtering is a well-known mechanism for finding recommendations. To experiment with CF we collect user's check-in information with their reaction to those places. It filters out those items that can be liked by the user based on the reactions by similar users. It also helps us to enrich each user's interest profile previously generated by SHCF. Generally, the CF algorithm produces a rating on unvisited places based on

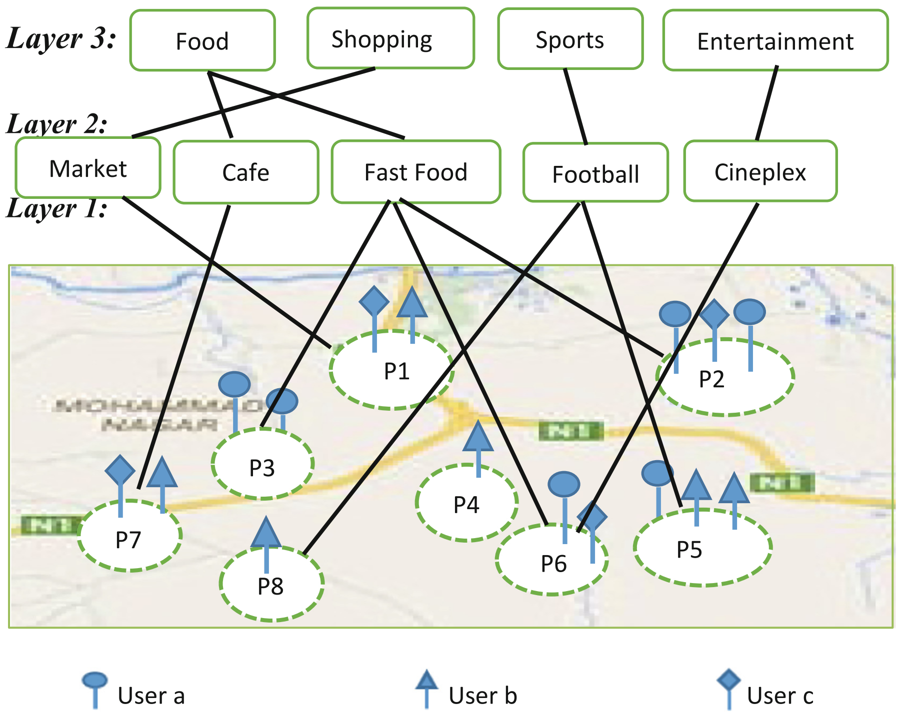

Fig. 2. Semantic Hierarchical Category-graph Framework (SHCF)

the dataset of previously rated check-ins by similar users. If we consider the rating of unvisited places is $r_{u,p}$, where for user u and place p is calculated by an aggregate of the ratings of N similar users.

$$r_{u,p} = aggr_{u^i \in u^\Lambda} r^i_{u,p} \tag{1}$$

Where u^Λ defines the N similar users to user u who have rated.

4.2.1 Discovering Similar Users and Unvisited Places

To measure the similarity of user and their unvisited spots we used Pearson Correlation Coefficient. It represents the relationship between the two users on a scale ratio or an interval. The coefficient ranges in between $+\ 1$ to -1, where $+\ 1$ indicates positive relationship, -1 represents perfect negative and 0 defines no relationship. Pearson correlation coeficient formula works as follows:

$$r_{x,y} = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{(N\sum x_i^2 - (\sum x_i).^2)(N\sum y_i^2 - (\sum y_i).^2)}} \tag{2}$$

Where, $r_{x,y}$ defines the similarity score between user x and y.

User visits different places with various rates of reaction. Here we try to find the similarity score between the user. Based on the similarity score it can say that an unvisited place might be a point of interest for a particular user. Thus it helps us to update the user profile combining both previously visited and unvisited places.

Table 3 shows the matrix representation of user to user similarity score.

Table 3. Similarity matrix

Similarity score	u_a	u_b	u_c	...	u_n
u_a	1	$SimScr_{a,b}$	$SimScr_{a,c}$	...	$SimScr_{a,n}$
u_b	$SimScr_{b,a}$	1	$SimScr_{b,c}$	...	$SimScr_{b,n}$
u_c	$SimScr_{c,a}$	$SimScr_{c,b}$	1	...	$SimScr_{c,n}$
...	...	...	...	...	...
u_n	$SimScr_{n,a}$	$SimScr_{n,b}$	$SimScr_{n,c}$	...	1

4.2.2 Grid Structure Generation

Before calculating the similarity between the user profile and POI spots extracted from google, we divide each type of geolocated (latitude & longitude) POIs into different grids. Each geolocation is converted into a cartesian coordinate and used in a grid-based data structure. Figure 3 shows the grid structure of our system where each grid contains various kinds of POI that are extracted from Google place API. Now we construct a Table 3 from this grid structure that shows the detailed information of the grid. For example, G13 contains three POIs of the type ‘t’ for theater, ‘c’ for cafe, and ‘p’ for the park. Some of the grid-like G12, G2, etc. does not contain any single POI, thus we can neglect those for our calculation. Now we have to measure the similarity of these listed grids and each user profile for recommendation generation. The algorithm describes the grid structure generation process.

Algorithm 1: Dividing given points into N * N grid

Input: places & N. Places contains each of the place geo-location and their type. N is the value of N * N grid.
Output: Location of the POIs in a N * N grid.

```
Begin
 for all poi ϵ places do
      poi.cartesian←geolocation_to_cartesian (poi.geolocation)
 end for
      minx←minimum(point.cartesian.x); for all point ϵ places
      maxx←maximum(point.cartesian.x); for all point ϵ places
      miny←minimum(point.cartesian.y); for all point ϵ places
      maxy←maximum(point.cartesian.y); for all point ϵ places
  gridsizex←(maxx - minx)/N; Size of grid in X axis
  gridsizey←(maxy - miny)/N; Size of grid in Y axis
End
```

4.3 Recommendation Generation

We measure the user interest by combining the user's check-in history and CF-based similarity correlation. The user profile also defines the priority rate for each choice of interest. Now we have to generate recommendations for a user. We need to divide extracted POI related geo-location surrounding on user's current location into different grids and made a list of POI table. We calculate the similarity between grid wise POIs and user's interest profile. A higher similarity score on a POI defines the required recommendations.

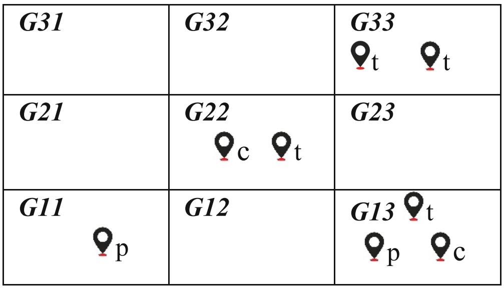

Fig. 3. Grid structure

Formula for score generation is as follows:

$$poi.score = p_s * poi.type * r \tag{3}$$

Where,

p_s is the priority score of user choices,
r is rating of POI.

Here we consider the rating of each POIs because in case of the same type of suggestions it will help us to find a better solution. The minimum distance between the user and suggested places is one of our concerns for the optimal solution. Before making the suggestion the manhattan distance between user location consisting grid and the grid contains the POI is calculated.

$$poi.man_{dis} = |poi.grid.x - user.grid.x| + |poi.grid.y - user.grid.y| \tag{4}$$

$$poi.score = (p_s * poi.type * r)/poi.man_{dis} \tag{5}$$

5 Experiment

In our experiment we take check-in places from real world and make the recommendations of our system.

5.1 DataSet and Implementation

We used a synthetic dataset where each user contains information about user_name, spot_type, latitude, longitude, rating to generate check-in spots. Table 4 shows the check-in dataset.

Table 4. Check-in information

User_name	Spot_type	Latitude	Longitude	Rating
Smith	Shoe store	−53.8343	64.99585	4.2
Carter	Park	−82.4145	−94.8644	3.5
Robinson	Spa	−11.7012	−4.23816	4.4
Bailey	Art gallery	6.171611	49.443224	4.5

In our work, the total number of check-in spots is 57,753 of 35,515 users. We considered 15,000 users who checked more than 20 spots and used DBSCAN clustering around a 100-m radius. In Pearson correlation formula we used 200 users profile for similarity calculation.

5.2 Evaluation

Precision and Recall are very well-known metrics to demonstrate a predictive system. To evaluate our system we considered 10 users for testing. There can be four possible outcome for place suggestion; true positive (t_p) - accurately predicted the recommendation, false positive (f_p) - predicted a recommendation that does not belong to a user interest, false negative (f_n) - failed to predict the actual places, true negative (t_n) - successfully avoids the places that are not on the profile.

$$precision = \frac{t_p}{t_p + f_p} = 0.9 \text{ and } recall = \frac{t_p}{t_p + f_n} = 0.9$$

The precision and recall value shows a very good outcome.

5.3 Example of Recommendation

We tested our work in Chittagong town, Bangladesh for different users. Our system extracts the user's current location and provides appropriate recommendations. For example, Fig. 4 shows some restaurants nearest to a user as a recommendation.

Fig. 4. Recommendation for a user

6 Conclusion

In this paper, we introduced Density-Based Spatial Clustering (DBSC) and a semantic hierarchical category graph framework (SHCF) to categorize the spots visited by the user. We also used Collaborative filtering to find out an unvisited place that might be a choice of activity for a user. we have tried to find the nearest possible places from the user's current location for a recommendation, which was a success.

Our work does not cover an option to suggest a group of users for a meeting place according to their similarity. In the future, it can be considered which will give more visiting options to the user.

References

1. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Springer, London (2003)
2. Herlocker, L.J., Konstan, A.J., Terveen, A.G., Riedl, T.J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
4. Horozov, T., Narasimhan, N., Vasudevan, V.: Using location for personalized POI recommendations in mobile environments. In: Proceedings of the Symposium on Applications and the Internet (2006)
5. Park, M., Hong, J., Cho, S.: Location-based recommendation system using Bayesian user's preference model in mobile devices. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73549-6_110
6. Chow, C., Bao, J., Mokbel, F.M.: Towards location-based social networking services. In: ACM LBSN, 2 November 2010
7. Zheng, Y., Zhang, L., Xie, X.: Recommending friends and locations based on individual location history. ACM Trans. Web **5**(1), Article 5 (2011)
8. Berjani, B., Strufe, T.: A recommendation system for spots in location-based online social networks. ACM, 10 April 2011
9. Lee, M., Chung, C.: A user similarity calculation based on the location for social network services. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011. LNCS, vol. 6587, pp. 38–52. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20149-3_5
10. Chen, H., Arefin, M.S., Chen, Z., Morimoto, Y.: Place recommendation based on users check-in history for location-based services. Int. J. Netw. Comput. **3**(2), 228–243 (2013)
11. Kosmides, P., Remoundou, C., Demestichas, K., Loumiotis, J., Adamopoulou, E., Theologou, M.: A location recommender system for location-based social networks. In: Proceedings of Mathematics and Computers in Sciences and in Industry (2014)
12. Mu, W., Meng, F., Chu, D.: A collaborative filtering recommendation algorithm based on user preferences on service properties. In: Proceedings of Service Sciences (2014)
13. Ajantha, D., Vijay, J., Sridhar, R.: A user-location vector based approach for personalised tourism and travel recommendation. In: International Conference on Big Data Analytics and computational Intelligence (ICBDACI) (2017)
14. Liao, G., Jiang, S., Zhou, Z., Wan, C., Liu, X.: POI recommendation of location-based social networks using tensor factorization. In: 19th IEEE International Conference on Mobile Data Management (2018)
15. Li, Z., Huang M., Zhang Y.: A collaborative filtering algorithm of calculating similarity based on item rating and attributes. In: 14th Web Information Systems and Applications Conference (2017)