Asian Journal of Information Technology 15 (7): 1166-1174, 2016
ISSN: 1682-3915

A Community-Based Hybrid Location Recommendation System in Location-Based Social Networks

K.P. Madhu and D. Manjula
Department of Computer Science and Engineering, Anna University, 600 025 Chennai, India

Abstract: This study is aimed at developing a location recommendation system in a location based social network. The location history of the user refers to the set of Point Of Interests (POI) that the user has visited in the past. The location history of the users reflect the interests and preferences of the user in a well-deserved manner. The location history associated with the users are modeled to determine the community of the user, using an apriori based frequent POI set mining. The set of POIs visited by the members in the community and that are not visited by the user are assigned a weight calculated from the polarity and the timestamp of the tip left by the visitors from the community. Then, the POIs with highest weights are recommended to the user. The cold start problem is tackled by identifying a representative user for each community and matching the user profile with the representative user for recommendation. A comprehensive study is performed on the dataset obtained from foursquare, the popular LBSN. The empirical analysis shows that location recommendation using our proposed model gives better and accurate results when compared with the systems that use collaborative filtering or content based techniques.

Key words: Community, location based social networks, point of interests, location recommendation, hybrid LRS

INTRODUCTION

With the advent of latest technologies, there is a radical change in the way the information is being stored and exchanged in the real world. The challenge is to retrieve and manipulate the pertinent knowledge from the appropriate source in an efficient manner. Social network is a proficient means of managing the information irrespective of the user's location and the type of the knowledge ingrained. But, providing appropriate techniques for the effective retrieval of apposite knowledge is still in its infancy. Filtering during knowledge retrieval is essential for the effective retrieval of information. The recommendation in Location Based Social Network (LBSN) refers to the recommendation of locations, users, activities or social media (Bao *et al*., 2015). This recommendation must be buoyed by regimented information retrieval techniques. Different studies show that the likelihood that a user would be interested in a location is influenced greatly by his friends. A prominent study on the point-location data gathered from foursquare reveals numerous patterns viz. 20% of the user check-ins occur within a distance of 1 km, 60% occur between 1 and 10 km, 20% occur between 10 and 100 km and a small percentage extend beyond 100 km (Noulas *et al*., 2011). Another pattern observed is like the user's activities vary during weekdays and weekends. These kinds of studies together with the analysis on the user and location correlations provide information about user preferences which can be utilized for recommendations in LBSNs. In this study, user check-in logs are analyzed to generate location histories.

Location based social networks: Social networks are constructs composed of interconnected individuals who interact with each other. These interconnections are defined by their interactions or relations. The relations can be like friendship, association, follow, subscribe, etc. For better comprehension, analysis and visualization, the social networks are represented as graphs where the nodes of the graph represent actors (individuals, locations, etc.) and the links between the nodes represent the relations between the nodes. The relation can be characterized as tags or name of the links. There is a dramatic hike in the amount of location data in information that are openly available in the internet with the popularization of social networking sites such as facebook places, google plus, google latitude, gowalla, brightkite, twitter, flickr, foursquare, etc. These networks duly create geo-tagged or location-tagged data and thus these networks are otherwise regarded as Location Based Social Networks (LBSNs). The services given by LBSNs

Corresponding Author: K.P. Madhu, Department of Computer Science and Engineering, Anna University, 600 025 Chennai, India

in this context allow users to share information in the form of text, image and video. Though, there are constraints on the availability and usage of location information because of security measures, there are still a large number of users who gracefully use their personal information in geo-tagged data.

The technique of location-tagging or geo-tagging facilitates the user to share the data in the social network along with location as an additional attribute. Various services provided by the social networks, especially the media services, make use of the location attribute on check-in data, text comments and other media contents, effectively. Other users to whom the data is shared would be able to learn the location or venues of the event. This is more appreciated with the popularization of the state-of-the art mobile phones and other such hand held devices which has given a boost to the number of common people using the services given by such networks. With these devices the users are able to login or check-in from any location and at any time. Also, the users could even go with low end devices supporting only a connection with Internet and GPS technology. With the growth of the LBSNs, majority of the researchers' focus is on the behavior analysis of user activities, users' relationships and correlations, user mobility inside the social network and of course, the predictions in respect to the links and the communities (Jin *et al*., 2012). The data integrity and security are also a major concern.

Foursquare is considered for a comprehensive analysis in this study. It's a point-location-based LBSN. In Foursquare, users can check-in at any venue and post their current locations. Points and badges are awarded to the users who check-in at a venue. The user with most number of check-ins at a particular venue is crowned as the "Mayor". The users would have to compete for retaining the crown. The users can find out friends around a certain physical location, from the real time location data updated by the checked-in users. This can enable them to go for social activities to happen. Users are allowed to add tips to the checked-in venues. This can be shared and can be read by others. These user generated contents like tips and badges are associated with the venues or the point of interests. This could enable the required users to gain location information. This information is exploited for location recommendation. The recommendations to users could be like friends, activities, venues to visit, travel itinerary and so on.

These recommendations are the driving force behind the success of the foursquare networking site. The opportunity of forming potentially useful groups among the users of the network is an absolute advantage of such networks. Users with similar tastes can be dragged into the group using such recommendations. People use the foursquare services to share their location with friends, meet new people and get coupons. Users are allowed to connect and publish their check-ins to facebook and twitter. Even people who are not a friend of the user on foursquare can still track or follow the users' where abouts through facebook. Foursquare allows the users to check-in at their actual location and announce, it in a dynamic environment along with their opinions about those locations. The working of the system will be discussed in the proposed methodology section of the study.

The location recommender systems: The proposed location recommender system is a collaborative filtering based system which makes use of the implicit rating for the location given by the users. The comments about different locations given by the users are considered to refine the recommendations. The user location history and the tips left by the users at these locations are the main inputs for the system.

This is a stand-alone location recommendation system which is meant to give the user with recommendations composed of individual locations matching their preferences, for instance, restaurants, tourist places or cities. The system is developed with the objective of information filtration through interactive intervention by the users. That is recommendation is made out of the implicit rating given by users in a community best representing the user for which the recommendation is to be done. This method generates a location recommendation which depends on the individuals' behavior and location history. Also, it contributes to the formation of personalized or customized aggregations in which the diversity in the source or the input tastes will enhance the recommendation.

The methodology deployed in recommendation systems includes content-based, link analysis-based, collaborative filtering-based and hybrid methods. Each of the methods has merits and demerits. Some of the short comings of the existing methodologies addressed by the proposed system include cold start problem, similarity/gray sheep and overspecialization.

Cold start problem occurs when a new user is added to the system and there is only diminutive information about the user is available with the system for recommendation. Similarity/gray sheep problem occurs when there are users with interests of no standards. Overspecialization occurs when the system recommends the same location to the user who has already appreciated it. All these problems can be handled well by the proposed technique.

Literature review: With the development of technologies that facilitate location-acquisition, the location-based social networking services like Foursquare, Twinkle, and GeoLife are getting flourished. Studies show that users with similar location history are more likely to have similar interests and preferences (Li *et al*., 2008; Xiao *et al*., 2010). It is also justified that the user's historical behavior is a strong indicator of the user's preferences (Eagle and Pentland, 2006, 2009). This suggestion is more strongly supported by the work which finds out that a user's historical behavior gathered in an LBSN is more effective and accurate than his online behavior in giving a user's preferences, patterns, interests and experience (Zheng and Zhou, 2011).

Location features are found to be equally important in profile-based location recommendation systems. Location recommendation systems using Collaborative Filtering (CF) models give personalized recommendations for locations by taking into consideration other users' ratings (Chow *et al*., 2010; Prete and Capra, 2010; Horozov *et al*., 2006; Ye *et al*., 2010). Each work gives a different version or enhancement to the basic CF method. But, most of these systems fail miserably in handling the cold start problem.

The three main components in a location recommendation system are considered to be the user's current location, the user's location histories and the location histories from the other users (Bao *et al*., 2012). Ye *et al*., 2011. Presents a location recommendation system that incorporates the user's preferences, the user's social connections and the geographic distance between the user and the candidate locations. But, these existing systems are not taking into account the comments given by the visitors on these locations which play a vital role in influencing their friends.

The CF-based location recommendation systems works by similarity inference, candidate selection and recommendation score prediction. They suffer from the following problems:

- Because of the large number of the users and items in the system, the similarity model construction is time consuming. This may also pose a problem to the scalability. This may be further worsened by the rapid growth and evolution of LBSNs
- When the rating matrix is sparse the system would fail in making the accurate or effective recommendations. This may happen when the number of user ratings is low
- The system performs poorly with the cold start problem
- The system is constrained if the number of visitors is comparatively less

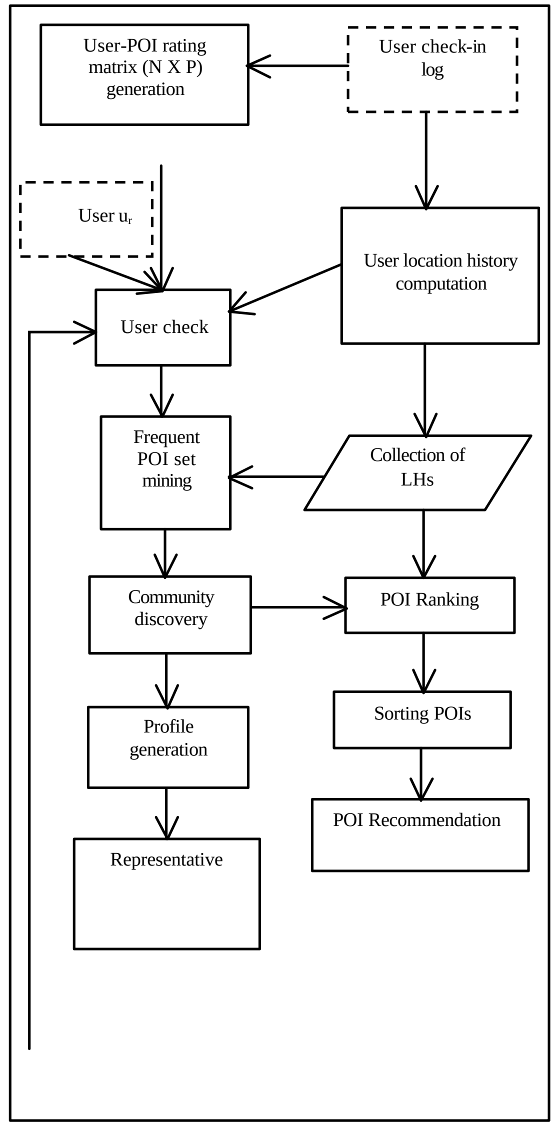

Fig. 1: Location recommender system

The proposed solution tries to give better results taking into account these limitations (Fig. 1).

MATERIALS AND METHODS

A mathematical model for the hybrid methodology is proposed to provide the recommendation in the location-aware social networks. Foursquare has been adopted as the domain for the deployment of the methodology. It is one of the emerging location based social networks in this sphere with >10 million users and numerous activities happening in the domain. It supports more of a collaborative approach in its social relations and recommendations which often raises the need for more effective techniques of content-based approach to generate expressive results in the social recommendation. We propose a hybrid method which incorporates the essence of both the techniques.

This approach combines the CF-based and the content-based approaches to overcome the demerits of the existing systems. The CF based component is accountable for the better recommendation of locations to the users by considering the preferences and ratings given by the fellow members of the community which the user belongs to. The system deploys a content-based component which is responsible for dealing with the cold start problem for users. It maintains the user profiles, finds the communities and generates an approximate profile for the group.

Then, a representative user is identified in the group whose profile will be set as the profile of the community. When a user with no rating is encountered, the system extracts the user's profile and compares it with that of the representative users to find the similar community for the user. Once found, the system will recommend the user with POIs that would be expected for the representative user. Thus, the system tackles the cold start problem.

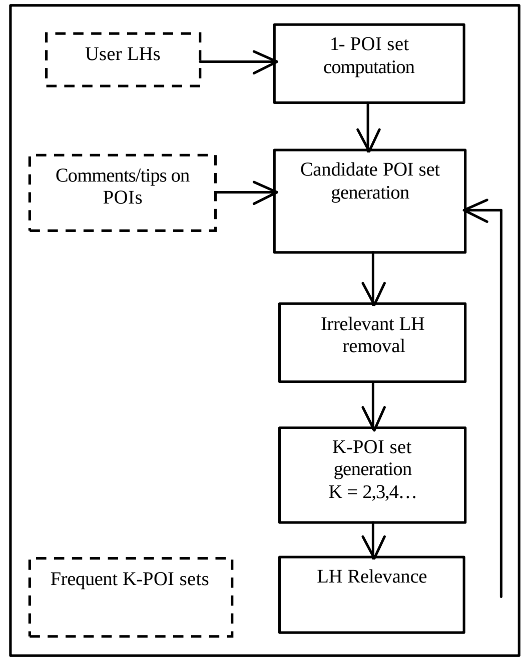

Fig. 2: Frequent POI set mining

Proposed location recommendation system: The system is meant to recommend locations based on user rating and profile matching of individual preferences. The hybrid recommender system has two main components. The CF component, first infers the implicit user ratings for all the POIs which is the number of times a user has visited the POI. It then determines the community to which the user belongs, based on the location history. Then, it determines the members of the community. The system then finds out the POIs visited by other members in the community and that are not visited by the user. These are the candidate POIs. Tips left by the users on these POIs are extracted and their sentiment score and relevance are calculated. A weight is assigned to the POI based on the above scores. The POIs are ranked based on these weights. The community is found out by means of the apriori based frequent POI set mining. The users whose location history has the maximum number of same POIs are considered to be more similar and are put in the same community. The content-based component is responsible for generating an approximate profile for the community. The system will then determines the best representative user for the community whose profile matches more with the approximate profile generated for the community. This user is assumed to represent the community and his profile is the community profile. When a new user with not even a single rating needs to be given with a POI recommendation, the system deduces the user's profile and tries to match it with the representative users' profiles. Thus, the system can find out the community of the user and the recommendation is done in such a manner that the POIs that may be recommended to the representative user would be recommended to the new user. The system makes use of wrappers for different APIs to extract the data from the social networks.

The overall architecture of the proposed location recommendation system is given in Fig. 1. The architecture of the two components for finding the frequent POI set mining and the POI rank calculation are shown in Fig. 2 and 3, respectively.

Mathematical model of the system: The steps for determining the rating matrix, R from the location history are given in Eq. 1:

- From the check-in data extract the check-in information corresponding to each of the venues
- $r_{ij} \in R$ can be set as the number of visits by the user u_i to the location l_j. r_{ij} will be incremented by one upon each visit. $r_{ij} = 0$ means that user u_i has not visited the location l_j at all

Form the rating matrix, the location history of the users can be determined as below:

$$LH_i = \{l_i \mid r_{il} \geq 1\} \text{ is the location history of } u_i \quad (1)$$

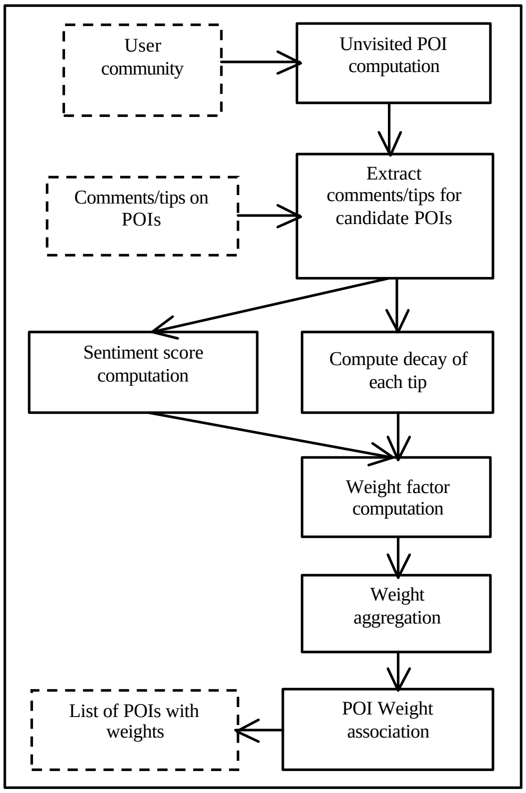

Fig. 3: POI rank calculation

The algorithm for recommending POIs to a user u_i where i = 1, 2 ... N are as follows. For a user u_i, identify potential friends of u_i from the community formed by apriori based frequent pattern mining. The members in the community except u_i are considered as the potential friends of the user u_i. Let us denote the set of potential friends of u_i as U_i^p. For the user $u_{,i}$ identify the POIs visited by user u_i and not visited by u_i from the rating matrix, R. It is represented as

$$P_j = \{1 \mid r_{jl} \geq 1, r_{il} = 0, u_j \in U_i^P\} \quad (2)$$

Where r_{ji} represents the rating of jth user u_i at the lth POI. The set of candidate POIs for recommendation to is u_i given as Eq. 3:

$$P = \bigcup P_j \quad (3)$$

For a POI, l∈p, a weight is assigned to the POI based on the polarity and the timestamp of the comments/tips given by the users $u_j \in U_i^p$ at various timestamps. The weight w_l is given by Eq. 4 and 5:

$$w_{il} = \frac{1}{K}\sum_{i=1}^{K} d_{il} t_{il}; \text{ where } t_{il} = \frac{S_{il}+1}{2} \quad (4)$$

$$d_{il} = 1-(|t_{c_l} - t|/T) \text{ and } d_{il} = 1-(|t_{c_l} - t|/T) \text{ and } S_{il} = [-1,1] \quad (5)$$

Here, S_{il} represents the sentiment score of the comment/tip for a particular POI, determined by using the vader Sentiment package in Python. The T_{cl} refers to the timestamp of the comment, in seconds. tis the current time in seconds. The value of T is the time range in seconds. The T is a variable set by the user. The value of T is taken corresponding to five years in our system. The POIs are ranked based on the weights. The POIs with the largest weight values w_l are then recommended to the user u_l.

The major algorithms in the system: The major algorithms implemented in the system are discussed in detail in this Algorithm 1.

Algorithm 1; Recommendation (check-in_log, u_r)

Input: (1) user check-in log, (2) user u_i
Output: none
Steps
    Start
    Access the check-in log crawled from Foursquare
    LH←Φ for all users//location history
    $R[u,POI] \leftarrow 0$ for all users and POIs
    Traj←Φ
    for each entry in the check-in_log do
    if the user is u_i then
    $LH_i \leftarrow LH_i \cup \{POI\}$
    $LH_i \leftarrow LH_i \cup \{POI\}$
    end if
    end for
    if $R[u_r, POI]$ = 0 for all POIs then
    $u_{rep} \leftarrow Find\_Community(u_r, REP_{users})$
    Hybrid_recommend (u_{rep})
else
    Hybrid_Recommend (u_r)
end if
    Stop

This is the root algorithm of the proposed hybrid recommender system which takes input from the user for the recommendation. The algorithm accesses the check-in log crawled from Foursquare, generates the location history for all the users and populates the rating matrix, R. It then checks whether the user is a new user or not. For a new user, there won't be any rating associated with him in the rating matrix. So, the system finds a similar community for the user by invoking the find_community algorithm. This algorithm will return the representative user for the community which closely matches the user profile. Then, the algorithm invokes the hybrid_recommend algorithm for recommendation Algorithm 2.

Algorithm 2; Hybrid_recommend u_r:
Input: (1) user u_r
Output: a list of recommended POIs for user u_r
Steps
Start
Traj←Φ
for each user u_i∈U do
Traj←Traj∪LH_i
end for
R_{CF}←Φ; K←0
$K_{POI} \leftarrow \text{Compute\_Frequent\_}K_{POI}\text{ set}(u_r, K)$
for each $K'_{POI} \in K_{POI}$ do
$\text{Users}_{K'_{POI}} \leftarrow \phi$
$C_{K'_{POI}} \leftarrow \text{Compute\_Community}\ (K'_{POI})$
$\text{profile}_{K'_{POI}} \leftarrow \text{Compute\_profile}\ (C_{K'_{POI}})$
$\text{REP}_{K'_{POI}} \leftarrow \text{Compute\_rep\_User}\ (\text{profile}_{K'_{POI}})$
$\text{REP}_{users} \leftarrow \cup \text{REP}_{K'_{POI}}$
$R'_{CF} \leftarrow \text{Compute\_POI Rank}\ (C_{K'_{POI}}, u_r)$
$R_{CF} \leftarrow R_{CF} \cup R'_{CF}$
end for
if R_{CF} = Φ then
K←K-1
Repeat steps b to d
else
Sort (R_{CF})
Recommend POIs in R_{CF}//Output
end if
Stop

This algorithm is the main algorithm responsible for giving the recommendation to the user. It takes input from the recommendation algorithm. It then re-computes the community of the user by finding the users whose location history contains frequent POI set. This is done to accommodate all the changes that has happened after the previous computation. The frequent POI set is found out with the help of the Compute_Frequent_K_{POI}s et algorithm. It also determines the users in the community, generates an approximate profile for the community and finds a representative user for the community by profile matching. Then, it ranks the POIs by means of the Compute_POIRank algorithm. The POIs with highest weights are recommended to the user Algorithm 3.

Algorithm 3; Compute_frequent_K_{POI} set (u_i, K):
Input: (1) user u_i, (2) K-size of the frequent POI set
Output: A collection of frequent POI sets
Steps
Start
minsup←2
L_1←{large 1-POIsets}
k←2
While L_{k-1}≠Φ do
C_k←candidate-gen (L_{k-1})//candidate POI set
for each c in C_k do
C. count←0
end for
for each LH l∈ Traj do//Traj is global
C_l←subset (C_k, l)
for each c in C_l do
C. count←c.count+1
end for
end for
L_k←{c∈C_k|C_k∈LH_i and c.count≥minsup}
if k>K then break
k←k+1
end while
return$\cup_k L_k$//collection of frequent POI sets
Stop

This algorithm is responsible for finding the frequent POI set for determining the community of the user. This algorithm is a contextualized apriori algorithm for frequent itemset mining. It determines the maximal and most frequent POI set that is appearing in the location history of the users and relevant to the user. The minimum support value is set as 2, so that the frequent POI set should be contained in at least two location histories Algorithm 4.

Algorithm 4; Compute_POI Rank (U, u_i):
Input: (1) friends of user u_i in the community, (2) user u_i
Output: A list of POIs with weights
Steps
Start
L←Φ; R_{CF}←Φ
for each user $u' \in U$ do
$L \leftarrow L \cup$ {POI visited by u and not visited by u_i}
end for
for each $L' \in L$ do
for each use $u' \in U$ do
$L'.tip_{u'} \leftarrow \text{Extract\_Tip}\ (\text{Tip}, L', u')$
//tip uploaded by u' on L'
$L'.ts_{u'}$ timestamp of the visit/tip by u' on L'
$SS_{u'} \leftarrow \text{Compute\_SentiScore}\ (L'.tip_{u'})$
$decay_{u'} \leftarrow 1-((L'.ts_{u'} - t)/T)$
//decay is the loss in relevance of the tip
//t is the current time
//T is the time range-5 years
end for
$L'.w_{u_i} \leftarrow (\sum decay_{u'} \times SS_{u'})/|U|$
$R'_{CF} \leftarrow R'_{CF} \cup \{L', L'.w_{u_i}\}$
end for
Return R'_{CF} //list of POIs with weights
Stop

This algorithm is used to rank the POIs according to the weights associated with them. The candidate POIs are determined as the POIs visited by at least one of the potential friends and not by the user. For these candidate POIs, the tips left by the visitors are extracted. We assume that there is a loss in relevance of the tip as the time goes on. So, the relevance of the tip will be based on the time elapsed since the tip is uploaded. The polarity of the tip is also calculated. Based on these two values, a weight is calculated and assigned to each of the candidate POIs. The POIs and their associated weights are returned by this algorithm. Here, the decay actually refers to the loss in relevance of the tip.

The complexity of the algorithm could be no more than O(NP) where N is the number of users, P is the number of POIs and P<<N.

RESULTS AND DISCUSSION

The data collected spans over Foursquare, the popular LBSN. For getting the individual user check-in data from Foursquare, there are two options. The first option is to use an application which makes use of the Foursquare API for gathering the required data of the user. This application will act as a wrapper for the Foursquare API. But, it is essential that the application should be authenticated by the corresponding users. The application has to specify the permissions to the users. The alternative option is to collect the data from Twitter or facebook. Foursquare facilitates the users to share their check-ins on twitter and facebook publicly. The publicly available check-ins can be crawled using the twitter's streaming API or the facebook API. This collected data is just a fair sample of the corresponding data available from the Foursquare LBSN. Filters are used to avoid replicated crawls.

We have chosen the first option by developing a java application for extracting the data from Foursquare. A group of >100 users have authenticated the application. The check-ins of these users were extracted using the application and communities were created dynamically. Approximate profiles were created for each of the scommunities. Representative users for each of the communities were identified. A weak social connectivity is being observed in Foursquare. Also, recommendations are not considering the reviews given by users on these locations. So, our model is preferred over the existing method.

The proposed recommendation system using hybrid technique is experimentally compared with the systems which are based on collaboration filtering technique and content based techniques. Significant improvement in performance is obtained. As part of a comprehensive analysis, particular nodes with known activity history, interests and profiles were selected and the system was made to recommend locations for the known user. Our system is found to give more accurate recommendations compared to its counterparts. The first evaluation method that is used in the study is normalized Discounted Cumulative Gain (nDCG) (Manning *et al*., 2008). The effectiveness of the recommendation can be found by using nDCG. Discounted Cumulative Gain (DCG) is commonly used in information retrieval systems. DCG can be used to measure the effectiveness, usefulness or gain of a document on the basis of its position in the result list. Here, the recommended POIs with respect to a check-in can be regarded as similar to a document in information retrieval. DCG uses a graded relevance scale of POI in the recommended list. DCG at a particular rank position p is given by Eq. 6:

$$DCG_p = gr_1 + \sum_{i=2}^{p} \frac{gr_i}{\log_2(i)} \tag{6}$$

Where gr_i is the graded relevance of the POI in the recommended list. The length of the recommended list varies. Thus for getting an accurate result, the DCG is normalized. The normalized DCG is given by Eq. 7:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{7}$$

Where:

$nDCG_p$ = The normalized DCG at a particular rank position p
DCG_p = The DCG at a particular rank position p
$IDCG_p$ = The ideal DCG at a particular rank position p

Here, the ideal DCG refers to the ideal ordering of the recommendations for the given check-in. The users were asked to check-in at a particular POI. The recommendations were made based on the query given by the user. The users were asked to judge the relevance of each of the recommendations with respect to the check-in. Each of the recommendation was judged on a scale of 0-3 with 0 representing irrelevant, 3 representing completely relevant and 1 and 2 representing the in between values. The graded relevance value of each of the recommendations is taken as gr_1 through gr_n Here, n is assigned values from 1 through 10. Then the DCG, IDCG and nDCG values are calculated for recommendation list of each of the check-ins by the users and it is used to calculate the effectiveness of the system. Table 1 shows the average $nDCG_p$ at the various rank positions from 1-10 in the recommended list calculated over the location histories of about 100 users selected randomly from the experimental data available.

Figure 4 shows the comparison of average $nDCG_p$ at the different rank positions in the recommended list given by different recommendation algorithms. It is obvious that the roposed Hybrid Recommendation algorithm clearly outperforms all the other algorithms which are being compared. The recommended POIs in the list are almost correct for each of the rank positions. That is, recommendations are more accurate in the case of hybrid recommender system compared to the content based or collaborative filtering based techniques.

Table 1: Average nDCGp at the various rank positions in the recommended list

	Average $nDCG_p$			
Rank position in the recommended list	Content-based RS	CF-based Rs	CF-based RS with tip polarity and decay	Hybrid RS (Proposed)
1	2.10	2.40	2.74	2.89
2	1.98	2.10	2.30	2.50
3	1.56	1.80	2.40	2.68
4	1.45	2.09	2.30	2.53
5	1.34	1.54	2.10	2.80
6	1.60	2.10	2.59	2.90
7	1.93	2.20	2.47	2.51
8	1.76	2.39	2.67	2.76
9	1.99	2.43	2.56	2.71
10	1.33	1.98	2.43	2.82

Table 2: Mean average precision versus the number of user check-ins

	Mean average precision			
No. of user check-ins	Content-based RS	CF-based Rs	CF-based RS with tip polarity and decay	Hybrid RS (Proposed)
50	0.50	0.51	0.60	0.70
100	0.49	0.55	0.63	0.74
150	0.52	0.57	0.65	0.74
200	0.47	0.59	0.67	0.75
250	0.51	0.55	0.56	0.77
300	0.45	0.64	0.69	0.78
350	0.45	0.65	0.70	0.77
400	0.47	0.66	0.69	0.80
450	0.46	0.64	0.69	0.82
500	0.48	0.69	0.71	0.85

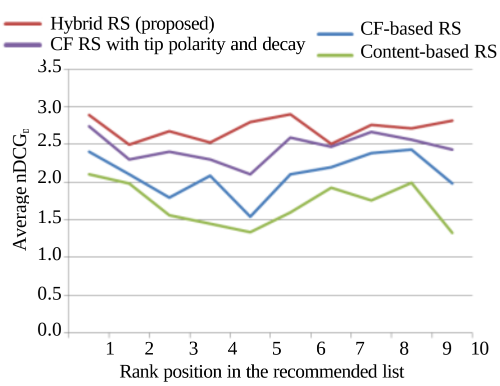

Fig. 4: Average $nDCG_p$ versus rank position

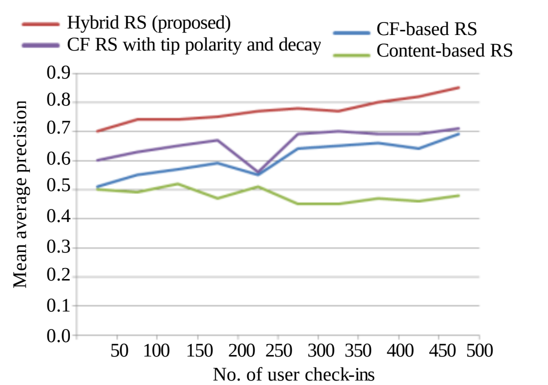

Fig. 5: Average precision versus number of user check-ins

The second evaluation method that we have used is the Mean Average Precision (MAP). MAP assumes that the user is interested in finding more relevant POIs for each of the check-ins. Table 2 gives the mean average precision for different number of user check-ins. The precision values for the different recommendations, selected randomly, are used to measure the correctness of the system. The graph plotting the comparison of MAP values corresponding to the various recommendation algorithms under consideration are given in Fig. 5. The graph is self-explanatory. The average precision of the recommendations done by the hybrid recommendation system is always better than the other recommendation algorithms like the content based and the collaborative filtering based algorithms.

As part of the input, some of the new users were added in between and the recommendation systems were to give recommendations for them. Irrespective of these inputs, the hybrid system gave better recommendations, without any drop in precision values where as other systems have shown drop in precision. This shows clearly that the hybrid recommendation system is better than the other algorithms in giving the near perfect recommendations to the users. The merits of this research include the possibility to increase the capacity of recommendation through an association between semantics and textual information in the form of tips given to venues.

The analysis proves unambiguously that the hybrid recommender system outperforms collaborative filtering based recommendation systems and content based recommendation systems. It gives better and accurate recommendations with consistent and good average precision values. The output or the quality of the recommendation never falls and are always well above compared with the other algorithms considered.

CONCLUSION

This study presents a methodology to perform the Location recommendation on location based social networks. The location recommendation is done based on the visits by the potential friends of the user. This is further refined based on the user generated comments/tips at the individual locations.

The development of a system using hybrid technique for the recommendation in location based social networks is the main contribution in this work. The technique tries to reduce the ambiguities and redundancies that are found in terms of semantic relations.

RECOMMENDATIONS

As a future research, it is planned to extend this model to a distributed environment using Hadoop mapreduce. Itinerary recommendation based on the recommended POIs is also planned to be done.

ACKNOWLEDGEMENT

The researchers would like to thank all the colleagues and well-wishers who have rendered their immense help and support to make this work a success.

REFERENCES

Bao, J., Y. Zheng and M.F. Mokbel, 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. Proceedings of the 20th International Conference on Advances in Geographic Information Systems, November 7-9, 2012, ACM, New York, USA., ISBN: 978-1-4503-1691-0, pp: 199-208.

Bao, J., Y. Zheng, D. Wilkie and M. Mokbel, 2015. Recommendations in location-based social networks: A survey. GeoInformatica, 19: 525-565.

Chow, C.Y., J. Bao and M.F. Mokbel, 2010. Towards location-based social networking services. Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, November 3-5, 2010, ACM, New York, USA., ISBN: 978-1-4503-0434-4, pp: 31-38.

Eagle, N. and A. Pentland, 2006. Reality mining: Sensing complex social systems. Pers. Ubiquitous Comput., 10: 255-268.

Eagle, N. and A.S. Pentland, 2009. Eigenbehaviors: Identifying structure in routine. Behav. Ecol. Sociobiol., 63: 1057-1066.

Horozov, T., N. Narasimhan and V. Vasudevan, 2006. Using location for personalized POI recommendations in mobile environments. Proceedings of the International Symposium on Applications and the Internet, January 23-27, 2006, IEEE, SAINT, Phoenix, Arizona, ISBN: 0-7695-2508-3, pp: 6-129.

Jin, L., X. Long and J.B. Joshi, 2012. Towards understanding residential privacy by analyzing users' activities in foursquare. Proceedings of the 2012 ACM Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, October 16-18, 2012, ACM, New York, USA., I SBN: 978-1-4503-1661-3, pp: 25-32.

Li, Q., Y. Zheng, X. Xie, Y. Chen and W. Liu et al., 2008. Mining user similarity based on location history. Proceedings of the 16th ACM Sigpatial International Conference on Advances in Geographic Information Systems, November 5-7, 2008, ACM, New York, USA., ISBN: 978-1-60558-323-5, pp: 1-34.

Manning, C.D., P. Raghavan and H. Schutze, 2008. An Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK., ISBN-13: 9780521865715, pp: 482.

Noulas, A., S. Scellato, C. Mascolo and M. Pontil, 2011. An empirical study of geographic user activity patterns in foursquare. ICWSM., 11: 70-573.

Prete, L.D. and L. Capra, 2010. Differs: A mobile recommender service. Proceedings of the Eleventh International Conference on Mobile Data Management (MDM), May 23-26, 2010, IEEE, Kansas City, MO, USA, ISBN: 978-1-4244-7075-4, pp: 21-26.

Xiao, X., Y. Zheng, Q. Luo and X. Xie, 2010. Finding similar users using category-based location history.

Ye, M., P. Yin and W.C. Lee, 2010. Location recommendation for location-based social networks. Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, November 3-5, 2010, ACM, New York, USA., ISBN: 978-1-4503-0428-3, pp: 458-461.

Ye, M., P. Yin, W.C. Lee and D.L. Lee, 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, July 24-28, 2011, ACM, New York, USA., ISBN: 978-1-4503-0757-4, pp: 325-334.

Zheng, Y. and X. Zhou, 2011. Computing With Spatial Trajectories. Springer Science & Business Media, Beijing, China, ISBN: 978-1-4614-1628-9, Pages: 306.