Mathematical Foundations of Computing
©American Institute of Mathematical Sciences
Volume **1**, Number **1**, February **2018**

doi:10.3934/mfc.2018002

pp. **11–48**

EXPLORING TIMELINESS FOR ACCURATE RECOMMENDATION IN LOCATION-BASED SOCIAL NETWORKS

Yi Xu

Gianforte School of Computing, Montana State University
Bozeman, MT 59717, USA

Qing Yang

Department of Computer Science and Engineering, University of North Texas
Denton, TX 76207, USA

Dianhui Chu

School of Computer Science and Technology, Harbin Institute of Technology
Weihai, Shandong 264209, China

(Communicated by Zhipeng Cai)

Abstract. An individual's location history in the real world implies his or her interests and behaviors. This paper analyzes and understands the process of Collaborative Filtering (CF) approach, which mines an individual's preference from his/her geographic location histories and recommends locations based on the similarities between the user and others. We find that a CF-based recommendation process can be summarized as a sequence of multiplications between a transition matrix and visited-location matrix. The transition matrix is usually approximated by the user's interest matrix that reflect the similarity among users, regarding to their interest in visiting different locations. The visited-location matrix provides the history of visited locations of all users, which is currently available to the recommendation system. We find that recommendation results will converge if and only if the transition matrix remains unchanged; otherwise, the recommendations will be valid for only a certain period of time. Based on our analysis, a novel location-based accurate recommendation (LAR) method is proposed, which considers the semantic meaning and category information of locations, as well as the timeliness of recommending results, to make accurate recommendations. We evaluated the precision and recall rates of LAR, using a large-scale real-world data set collected from Brightkite. Evaluation results confirm that LAR offers more accurate recommendations, comparing to the state-of-art approaches.

1. Introduction.

With billions of users, location-based social networks (LBSNs), have become the most popular applications. In a LBSN, users can easily share their geospatial locations and location-based contents in the physical world [9, 34]. The rich knowledge that has accumulated in these social networks enables a variety of location-based recommendations. Recommendations in a LBSN is relevant to

2010 *Mathematics Subject Classification.* Primary: 68T20, 68T37; Secondary: 68U01.

Key words and phrases. Recommendation, Collaborative Filtering, recommendation timeliness, location-based social networks, accurate recommendation.

Qing Yang and Dianui Chu are the corresponding authors of this paper.

users since it helps users discover locations they may like, as well as guides users when they are visiting places that they are not familiar with. What's more, the recommendation timeliness that is proposed in this paper enables users to discover new locations with a dynamic preference. Location visiting history, as one of the most important components of user context, implies extensive knowledge about an individual's interests and behavior, thereby providing us with opportunities to better understand users in a social structure according to not only online users' behavior but also the users' activities in the physical world.

There is a wide range of applications that provide location-based services. For example, a user can share the location of a restaurant that he often visited with his friends through an online social network using his cellphone. Other users can expand their social networks using friend suggestions derived from overlapped location histories. Another example of the application of this service is to provide customized schedule for users instead of the search engine. Imagine that there is an app that could set up a complete visiting schedule for you that are the same as you planned without doing any search. In other words, the location-based social network allows users to share location-based information and get to know themselves and other users who have similar interests by sharing their check-in experiences. Recommendations in location-based social networks are trying to predict a user's preference based on his location visiting history and recommend locations that may attract the user.

In this paper, by understanding the process of CF-based recommendations, we proposed a novel Location-based Accurate Recommendation (LAR) approach with a recommendation timeliness. This approach takes advantage of accurate transition matrix and recommendation timeliness and provides accurate recommendation results. Based on the results of experiments using a real-world data set, LAR outperforms the state-of-art approaches on recommendation results with timeliness.

1.1. Problem statement.

In a location-based social network, users' visiting histories will somehow reflect the interests of users and further the similarity relationships between users [48]. This conclusion is based on the assumption that 'users will more likely visit places they are interested in' [48]. The applications of location-based social networks are developmentally increased as the improvement of driverless vehicles. Among them, recommendation system is one of the most critical applications in a location-based network. The primary goal of a location-based recommendation is to mine interests information and predict future interest locations based on both the social relations and the user's interest history. Apparently, recommendations in location-based social networks require the access to user's personal information, which may cause privacy leakage [19,41,62]. As this paper focuses on understanding the recommendation problem itself, we simple consider privacy-aware recommendation as our future work.

Fig.1 shows a location-based social network on a real-world geospatial map. Users are represented by U_k, and locations are marked by red and represented by l_k. The blue arrow shows the similarity relations between users (users connected by blue arrow lines are similar to each other), while the black arrow means the check-in activity of a user performed on a location (i.e., there exists a black arrow means a particular user U_i has a visiting history on a location l_j). In a location-based recommendation system, the problem is to predict locations that users are interested in. For example, user U_1's preference will be predicted based on the known and computed information of him or her: 1) U_1 has visited three locations l_3, l_5 and l_6; 2) U_1 is similar with three other users U_2, U_3 and U_4.

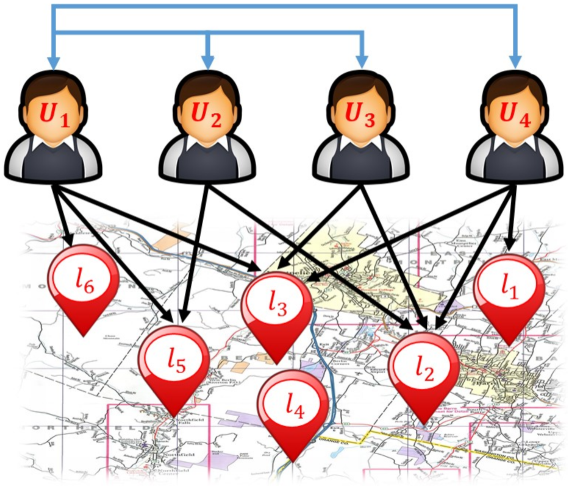


FIGURE 1. Overview a location-based social network

In conclusion, the problem of location-based social network Point-of-interests (POIs) recommendation can be formally described as, recommending POIs to a user based on observed check-in actions of all the users in the network with the consideration of influence factors, i.e., geographical influence, social influence, popularity influence.

1.2. **Collaborative Filtering.** Traditional recommendation systems in location-based social networks often use the Collaborative Filtering (CF) approach for data collection and recommendation. Collaborative Filtering makes use of historical data to mine user's preference and similarity, has been significantly successful. In location-based Collaborative Filtering, the interactions between users and locations are represented by the number of check-ins of each user performed to each location. These interactions will be stored in a user-location matrix: 1) If there are at least one interactions, the value will be larger or equal to 1, 2) If there is no interaction or no certain number of interactions, the value will be set to 0. After user-location matrix has been obtained, for a target user, it matches this user's check-in records against other users' and finds the users with most "similar" tastes. This step is known as 'collaborative'. With similar users, it recommends locations that the similar users have rated highly but not yet been rated by the target user (presumably the absence of rating is often considered as the unfamiliarity of a location). This step is the 'filtering' step, since the system filters locations that can be recommended to users based on the "similar" information from the last step. With 'collaborative' and 'filtering', a location-based Collaborative Filtering is able to recommend potential interested locations to target users.

1.3. **Limitation of collaborative Filtering.** Although the use of different forms of Collaborative Filtering models has proved to be a great success in tons of recommendation systems, there are some limitations. First, the largest restriction of all CF-based recommendations is data sparsity. In detail, there are limited number of interactions between users and locations and this will result in the sparsity of historical user-location matrix. Due to the sparsity of data, information can be collected from the history as well as the recommendation results may also be limited. What's more, users' activity histories are often locally clustered, which amplifies the data sparsity problem in the check-in data. Another limitation is a lack of proof of timeliness of recommendation results. The goal of a recommendation system is to predict the preferences of users. However, the timeliness of each prediction is unknown. In other words, there has been no research significantly computing the timeliness of recommendation results, which is the only way to make predictions worthy. Furthermore, the assumptions made about external influences in different CF-based recommendations make timeliness more difficult to compute.

1.4. **Proposed approach.** To systematically handle these limitations, this paper first analyzes and understands the process of Collaborative Filtering (CF) approach, which mines an individual's preference from his/her geographic location histories and recommends locations based on the similarities between users. We find that a CF-based recommendation process can be summarized as a sequence of multiplications between a transition matrix and the visited-location matrix. The transition matrix is usually approximated by the user's interest matrix that reflect the similarity among users, regarding to their interest in visiting different locations. The visited-location matrix provides the history of visited locations of all users, which is currently available to the recommendation system. We find that recommendation results will converge if and only if the transition matrix remains unchanged; otherwise, the recommendations will be valid for only a certain period of time.

Based on our analysis, a novel location-based accurate recommendation (LAR) method is proposed, which considers the category information of locations, as well as the timeliness of recommending results, to make accurate recommendations. In LAR, the transition matrix and visited-location matrix are obtained by the user-category matrix, where history locations are classified into categories. The categories of locations are identified by considering both geographic distance between locations using K-means clustering method and semantic meanings of locations using SVD. After that, by multiply the transition matrix by visited-location matrix, a predicted user's preference matrix containing all the influence factors is obtained. At last, to perform more accurate recommendations, in LAR, the timeliness of each recommendation result will be computed.

Our empirical studies consist of multiple parts. First, we conduct several experiments to evaluate the impacts of several factors in a CF-based recommendation process. Next, we evaluate the recommendation results of three benchmarks and our proposed approach. We generate the conclusion that our proposed Location-based Accurate Recommendation outperforms other approaches. At last, we assess the effectiveness of our integrated framework by comparing with several competitive benchmarks considering recommendation timeliness.

1.5. **Contributions.** The main contributions of this paper are summarized as follows. First, by a systematic analysis of the generation of recommendation results, we were able to understand the process of Collaborative Filtering (CF) approach. To the best of our knowledge, the process of recommendation in CF-based recommendations can be approximated as the process of matrix multiplication between a transition matrix and the visited-location matrix. What's more, by proof of the unsatisfactory of convergence of recommendation results, we make the conclusion that all recommendation results have timeliness. Second, by understanding the importance of the transition matrix, we proposed a category-based Collaborative Filtering recommendation approach, which takes distance between locations and semantic meaning of each location, as well as other factors that will influence the recommendation results into consideration. This approach will result in a relatively precise user's similarity matrix that is approximate to the accurate transition matrix. Third, by considering the real world appearance that user's preference may change over time, we propose a novel recommendation approach, Location-based Accurate Recommendation (LAR). LAR considers the category information of locations, as well as the timeliness of recommending results, and is able to make more accurate recommendations. Fourth, based on a real word data-set, we have conducted extensive experiments to evaluate the effectiveness of our model. The results reveal that our approaches significantly outperform other state-of-art approaches.

2. **Related works.**

2.1. **Collaborative Filtering.**

Collaborative Filtering (CF) [16, 20, 32, 37, 57] is a technique that has been widely used in recommender systems. Traditional Collaborative Filtering is used by Amazon.com, which solves the problem of predicting unknown interested items for a user based on his or her purchase history on Amazon. In detail, as one of the most successful approaches to building recommender systems, Collaborative Filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. The general idea behind Collaborative Filtering is that similar users vote similarly on similar items. In the more general sense, Collaborative Filtering is the process of filtering for information or patterns using techniques involving collaboration among data sources.

In a typical CF scenario, there is a list of m users $\{u_1, u_2 ... u_m\}$ and a list of n items $\{i_1, i_2 ... i_n\}$, and each user u_k, has interactions (either purchased or left comments) to a list of items I_{i_k}. The ratings are left by users after the interaction happens, which ranging from 0 to 5 in most examples. Traditional recommender systems make recommendations by exploiting ratings for items. For example, we can convert the list of people and the items they like or dislike to a user-item rating matrix (Table 1), in which Tony is the active user that we want to make recommendations to. There are missing values in the matrix where users did not give their preferences for certain items.

Table 1 shows an example of a user-fruit matrix, where binary values represent the preferences of users to items. In this example, the particular question is that "What is the predicted preference of Tony on watermelon?" To answer that question, according to Collaborative Filtering, first, the similarities between users will be calculated. After computation, results show that Alice is the one that shares two same interests with Tony on Apple and Grape. The next step is to make the prediction based on Alice's history data. As shown in Table 1, Alice dislikes watermelon, which results in a "dislike" rating for the prediction of Tony's interest.

Through the use of Collaborative Filtering approach has been significantly successful on previous works, there are some disadvantages: 1) The principal problem is that CF systems cannot produce recommendations if there are no ratings available. In general, we call this limitation of Collaborative Filtering as the data sparsity problem, which shows that the recommendation performs poorly when the user-item matrix is sparse. 2) They demonstrate poor accuracy when there is little data

Table 1. An example of user-item matrix

	Apple	Pear	Grape	Watermelon
Alice	Like	Like	Dislike	Dislike
Bob		Dislike	Like	Like
Chris		Dislike	Like	
Tony	Like		Dislike	

about users ratings. This and previous disadvantage are called the above-mentioned Cold-Start problem. 3) Another principal disadvantage is that CF systems are not content aware, meaning that information about items are not considered when they produce recommendations.

Most of the remarkable advantages of Collaborative Filtering Recommender systems can be merely derived from the Collaborative Filtering disadvantages: 1) CF systems do not require content information about neither users or items to be machine-recognizable. Pure CF methods utilize only ratings and do not require any additional information about users or items. These systems can make an assessment of quality, style or viewpoint by consideration of other people's experience. 2) The notable advantage is that CF systems can produce personalized recommendations because they consider other peoples experience and recommendations are based on that experience.

There are two main categories of the Collaborative Filtering method: memory-based CF and model-based CF. Memory-based CF uses user rating data to compute the similarity between users or items. Model-based CF models are developed using data mining, machine learning algorithms to find patterns based on training data.

2.1.1. *Memory-based CF.* In memory-based CF [20], the most commonly used method is user-based CF. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie, for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. This information is used in the decision on which movie to see. In [46] and [50], the similarity between items is computed by building a user-user matrix by iterating through all item pairs and computing a similarity metric for each pair. After the similarity has been computed, the system will make recommendations based on user's similarity. choosing the top-k interested items of top-k similar other users.

The advantages of this approach include the explainability of the results, which is an important aspect of recommendation systems; easy to create and use; smooth facilitation of new data; content-independence of the items being recommended; and good scaling with co-rated items.

There are also several disadvantages to this approach. Its performance decreases when data gets sparse, which frequently occurs with web-related items. This hinders the scalability of this approach and creates problems with large datasets. Although it can efficiently handle new users because it relies on a data structure, adding new items becomes more complicated since that representation usually relies on a particular vector space. Adding new items requires the inclusion of the new item and the re-insertion of all the elements in the structure.

2.1.2. *Model-based CF.* This approach has a more holistic goal to uncover latent factors that explain observed rating. Most of the models are based on creating a classification or clustering technique to identify the user based on the training set [45]. The number of the parameters can be reduced based on types of principal component analysis. By analyzing the latent semantic meaning of each interest, model-based CF can achieve more general results than memory-based CF. An example of model-based CF is the algorithm called sparse matrix SVD [30]. This approach models both users and movies by giving them coordinates in a low dimensional feature space, i.e., each user and each movie has a feature vector. Each rating (known or unknown) is modeled as the inner product of the corresponding user and movie feature vectors. In other words, we assume there exist a small number of (unknown) factors that determine (or dominate) ratings, and try to determine the values (instead of their meanings) of these factors based on the training data. Mathematically, based on the training data (sparse data of a large matrix), we try to find a low-rank approximation of the user-movie matrix A. This algorithm solves the data sparsity problem as well as considers the semantic meaning of each item. However, the results of SVD is more general than the ones we get from a memory-based model.

One advantage of using this approach is that instead of having a high-dimensional matrix containing a sufficient number of missing values we will be dealing with a much smaller matrix in lower-dimensional space. A reduced presentation could be utilized for either user-based or item-based neighborhood algorithms that are presented in the previous section. There are several advantages with this paradigm. It handles the sparsity of the original matrix better than memory based ones, and comparing similarity on the resulting matrix is much more scalable especially in dealing with large sparse datasets.

2.2. **Current approaches to location-based recommendation.** There are several unique characteristics of location-based social networks, which distinguish POI recommendations from traditional recommendation tasks.

2.2.1. *Tobler's first law of geography.* The first law of geography is "everything is related to everything else, but near things are more related than distant things" [2,27]. This indicates that geographically proximate points of interest (POIs) are more likely to share similar characteristics. Also, the probability of a user interested in a POI is inversely proportional to geographic distance. This is different from traditional online social networks by both the spatial factor and the relations between people. For example, in reality, a person usually visits a POI, e.g., museums, and then travel to its nearby POIs, e.g., restaurants and stores. In [54], they consider the geographical correlations between POIs and make the assumption that the nearby POIs have the stronger geographic associations than the POIs that are far from them.

2.2.2. *Regional popularity.* Two POIs with similar or the same semantic topics can have different popularities if they are located in the various regions [29, 49, 59]. This indicates that even for two similar or same POIs, the locations of them may affect the number of visited people. As a result, besides the semantic meaning of each location itself, popularities of each location should be considered separately when making a recommendation. [6] treats popularity as one of the most important factors that influence peoples choice of different locations. In [6], locations with high popularity will receive higher scores and further changes peoples decision.

2.2.3. *Dynamic user mobility.* In LBSNs, a user may check in POIs at different regions. For example, an LBSN user may travel to different cities. Dynamic user mobility imposes enormous challenges on POI recommendations. Furthermore, some locations have never been visited by any user. In most of the previous works, i.e. [1] and [3], that were built on CF models, it was hard to recommend new items since these items have never received any feedbacks from users in the past. In [52], they solve this problem by considering a knowledge base embedding for recommender systems. In their model, the semantic meanings of locations were mined based on their structural and nonstructural knowledge.

2.2.4. *Implicit user feedback.* In the study of POI recommendations, the explicit user ratings are usually not available [5]. The recommender system has to infer user preferences from implicit user feedback regarding user check-in frequency data [63]. This was used from the beginning when Collaborative Filtering model was used for recommendation system.

More specifically, we divide location-based social network recommendation systems into two broad categories according to the influence factors: Geographical influence enhanced recommendation and Social influence enhanced recommendation.

2.2.5. *Geographical influence enhanced recommendation.* In a location-based recommender system, geographically proximate points of interest (POIs) are more likely to share similar characteristics. That is based on the Toblers First Law of Geography: “everything is related to everything else, but near things are more related than distant things” (Tobler 1970). Several studies [10, 12, 17, 25, 45, 53] argue that geographical clustering phenomenon in users check-in activities, known as the geographical influence, can be utilized to improve the POI recommender systems. [45] integrated geographical influence into a location-based recommendation by making an assumption that the willingness that a user moves from a location to another location is a function of their distance. The function is under the power law distribution assumption. In detail, the willingness of the user to visit a location distance far away as well as the probability that user will check in l_j, given the user is currently at location l_i, is defined by the following two Equations.

$$w_i\left(dis\right) = a \times dis_1^k \tag{1}$$

$$P(l_j|l_i) = \frac{w_i\left(dis\left(l_i, l_j\right)\right)}{\sum_{l_k \in L, l_k \neq l_i} w_i\left(dis\left(l_i, l_k\right)\right)} \tag{2}$$

where a and k are parameters of power-law distribution. By applying this, [8] was able to take the geographical influence into consideration.

Location-based Geo-Social (LGS) [6] is another previous work which takes geographical influence into consideration. They first measure the social knowledge of both the locations and the users by considering the popularity of a location and the expertise of a user. Locations that were visited by some experts are considered attractive locations. The inverse is also true, users that have visited a number of popular locations are treated as experts. The TF-IDF value computes user’s preference in this approach. Intuitively, a user would visit more locations belonging to a category if the user likes it. Further, if a user visits locations of a category that is rarely visited by other people, the user could like this category more prominently. User’s similarity in this baseline was calculated by comparing the expert score of users and the popularity of the corresponding locations that were visited by these users. Finally, the recommendation was made by the matrix multiplication method.

By taking geographical influence into consideration, previous works were able to categorize POIs into clusters based on their current locations and further solve the data sparsity problem in the traditional Collaborative Filtering method. Furthermore, according to the success of these works, the Tobler's first law of geography was proved to be reasonable, which means locations with similar geographical locations do share similar semantic meanings. Although geographic influence in location-based recommender systems is a valuable factor to be considered, there are several limitations: 1) In most previous works, they assume the distribution of closeness over distance as power-law distribution, and 2) They ignore other semantic meanings of locations other than the geographical influence.

2.2.6. *Social influence enhanced recommendation.* Social influence enhanced recommendation studies have been extensively explored in traditional recommender systems, include memory-based methods [20, 28] and model-based methods [24, 61]. Inspired by the assumption that friends of LBSNs share more common interests than non-friends, several POI recommendation approaches to improve the quality of recommendation by taking social influence into consideration [35, 38, 45]. Particularly, social trust among users has been widely studied [18, 21–23] and considered an important parameter in recommender systems. In fact, trust-aware recommendation was applied in vehicular networks to intelligently recommend information to vehicles [33, 42–44].

Social influence was taken into consideration by examining the similarity between users when recommending locations to a user. For example, in [45], Ye et al. proposed the user-based and location-based Collaborative Filtering methods that can be used in a location-based social network. User-based Collaborative Filtering makes the assumption that similar users would be interested in similar locations. On the other hand, location-based Collaborative Filtering assumes that similar locations would be visited by similar users. In their work, they use the cosine similarity to measure the similarities between users and locations. The cosine similarity is defined as follows:

$$(sim_{u,v})^{cosine} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}} \tag{3}$$

where u_i and v_i are components of vector c_u and vector c_v. The conclusion of their work shows that user-based Collaborative Filtering performs better than location-based Collaborative Filtering. This is because that number of locations in a recommendation system is way larger than the number of users which results in a more accurate user's similarity than location similarity.

Another commonly used way to generate similarity between users is called Pearson correlation [20]. Pearson correlation measures the extent to which two variables linearly relate with each other. For the user-based algorithm, the Pearson correlation between users u and v is

$$(sim_{u,v})^{ppc} = \frac{\sum_{i\in I}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i\epsilon I}(r_{u,i} - \bar{r_u})^2}\sqrt{\sum_{i\epsilon I}(r_{v,i} - \bar{r_v})^2}} \tag{4}$$

where the $i \in I$ summations are over the items that both the users u and v have rated and $\bar{r_u}$ is the average rating of the co-rated items of the uth user.

In a real-world situation, different users may use different rating scales, which the vector cosine similarity cannot take into account [31]. To address this drawback, adjusted cosine similarity is used by subtracting the corresponding user average from each co-rated pair. The adjusted cosine similarity has the same formula as Pearson correlation. In fact, Pearson correlation performs cosine similarity with some normalization of the users ratings according to his rating behavior.

GeoSoCa [54] is another work that takes the social influence into consideration and generates the highest precision ratio when recommending new locations to user u_i (locations have never been visited by user u_i). First, in GeoSoCa, they propose a kernel estimation method with an adaptive bandwidth to determine a personalized check-in distribution of POIs for each user that naturally models the geographical correlations between POIs. Then, GeoSoCa aggregates the check-in frequency or rating of a users friends on a POI and models the social check-in frequency or rating as a power-law distribution to employ the social correlations between users. Further, GeoSoCa applies the bias of a user on a POI category to weigh the popularity of a POI in the corresponding category and models the weighed popularity as a power-law distribution to leverage the categorical correlations between POIs.

Social opinions were first proposed in the traditional Collaborative Filtering [20] and were further improved in the following works after that. The idea of social opinions has proved to work well when the user-location matrix contains enough information that was needed for the recommendation. However, when the matrix is sparse, which is true in most cases, social opinions cannot be fully considered. This is one of the largest limitations of social influence enhanced recommendation, where leaves the cold start problem. Although in [52], the cold start problem was solved by using deep learning to filter the knowledge base of each item (location), social influence is not the key point in solving this problem.

In conclusion, previous works proposed recommendation methods based on traditional Collaborative Filtering and improved the recommendation rate by taking several influence factors into consideration. Both memory-based and model-based CF were used in previous studies for different purposes. In this paper, we propose a novel periodical recommendation method, which takes the similarity matrix as a transition matrix and generates a relatively high recommendation accuracy. To the best of our knowledge, this is the first work combining matrix multiplication and the Markov chain method with Collaborative Filtering to perform recommendations.

3. Generation of recommendation results.

3.1. Understanding the problem.

An online social networking service is a representation of real-world social networks. The social networking services reveal users' real social connections and also enhance the growth by allowing them to share and communicate about ideas, activities, events, news and interests in a much easier way. A location-based social network is a new structure of the social network, which contains a geographical layer that contains real-world locations and each user's location check-in information. This is different than traditional rating information since, except social influence, location and distance between locations may also affect users' opinions. In an LBSN, users can easily share their geospatial locations and location-based contents in the physical world. The rich knowledge that has accumulated in these social sites enables a variety of location-based recommendations. For example, in a traditional social network, we can describe a users interest by "Alice likes Chinese food." However, in a location-based social network, we should include location information when describing a user's interest, such as "Bob likes Chinese food of the Chinese restaurant located on the main street which is 1 mile away from his home." In this case, location and distance are extra information in a location-based social network compared with a traditional online social network, which will result in a new recommender system based on location-based social networks.

Zheng elaborates the concept of these location-based social networks [17], as: "A location-based social network (LBSN) does not only mean adding a location to an existing social network so that people in the social structure can share location-embedded information, but also consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and text. Here, the physical location consists of the instant location of an individual at a given timestamp and the location history that a person has accumulated over a period. Further, the interdependency includes not only that two persons co-occur in the same physical location or share similar location histories; but also the knowledge, e.g., common interests, behaviors, and activities, inferred from an individuals location (history) and location-tagged data."

Assume there are m users $u_1, ..., u_m$ and n locations $l_1, ..., l_n$ in a location-based social network, where $m << n$. Let $u = \{u_1, ..., u_m\}$ be the set of users and $l = \{l_1, ..., l_n\}$ be the set of locations. For each user u_k, there is a history of check-in information on a set of locations L_k. Check-in information show the interaction between users and locations and each check-in has a timestamp. The number of check-ins for each user at a specific location are different. This may somehow show the difference in interests of users on that particular location. $C \in \mathbb{R}^{m \times n}$ is the check-in matrix with each value C_{ij} representing the number of observed check-ins made by user u_i at location v_j.

$$C_{ij} = \begin{cases} C_{ij}, & \text{if at least one check-in exists} \\ 0, & \text{if no check-in history} \end{cases} \tag{5}$$

As a result, we can transform all the users' check-in histories into interest extents. Other than the check-in information, several factors will influence the choice of users in new locations. As we discussed in the previous section, existing works take geographical influence, social influence, and popularity influence into consideration when predicting user's preference. By considering the historical check-in information and all the influence factors in a location-based social network, a user's preference is predictable. For a user u_k, his or her preference can be represented by a location vector $\vec{P_k}$ and $\vec{P_k} = \langle P_{l_1}, ..., P_{l_n} \rangle$, where P_{l_a} is the preference value of user u_k to location l_a.

Under such a location-based social network, where users' preference can be predicted, a location-based recommender system is a valuable and unique application. Specifically, location recommendations provide a user with POIs that match his or her interests within a spatial area. This application becomes more worthy when people travel to an unfamiliar city. In a recommender system, recommendations are made to users to save time searching for places they would like to go, and helping them with choices when they travel to unfamiliar cities.

In such a recommender system, the goal is to predict all users' preferences on all locations based on the training set (check-in history). In other words, we want to replace all "0"s in a user-location matrix by some optimal prediction. As a result, the goal is to minimize the root mean square difference between training set and test set:

$$d_{tt} = \sqrt{\frac{1}{|S_{test}|} \sum_{(i,j) \in S_{test}} (C_{ij} - P_{ij})^2} \tag{6}$$

where $(i, j) \in S_{test}$ if user i checks in at location j in the test set. C_{ij} is the true check-in number and P_{ij} is the predicted preference based on the recommender system.

As a result, the problem of location-based social network Point-of-interests (POIs) recommendation can be formally described as follows. **A location-based social network Point-of-interests (POI) recommendation seek to recommend POIs to a user based on observed check-in actions of all users in the network with the consideration of influence factors, i.e., geographical influence, social influence, popularity influence.**

3.2. **The principle of CF recommendation.** Collaborative Filtering (CF) has become one of the most commonly used approaches to provide recommendations in location-based recommender systems. The key to this collaborative filtering approach is to summarize each user's check-in information and find relations between users or locations using user-location interaction history so that the system can perform recommendations for users. Furthermore, in most works related to this approach, user's relations and user's check-ins are measured by a matrix. The process of combining user's relations to user's check-ins is performed by the multiplication of these two matrices. As a result, in this paper, we summarize the process of Collaborative Filtering recommendations as a process of matrix multiplication.

In a Collaborative Filtering recommender system, the system needs to consider both user's check-in information as well as social opinions. Values in a user's check-in matrix are based on number of check-ins and other factors, like popularity, that might influence users choice. On the other hand, a user's relation matrix is based on the similarity between pairwise users' check-in histories. To consider the social opinions before recommendation, we multiply these two matrices, a $M * M$ user-user (user's relation) matrix, and a $M * N$ user-location (user's check-in) matrix, as defined in the previous section. After multiplication, the results will be a user-location matrix that containing the social opinions from other users in the network. Equation 7 shows an example of this matrix multiplication process.

As shown in equation 7, the leftmost matrix represents user's similarity and relationship, while the user-location matrix in the middle is the weighted users' check-ins we get from the previous step. Users similarity is represented by values ranging from 0 to 1, while user's preference is represented by values ranging from 0 to 5. In this example, to simplify the results, neither user's similarity matrix nor user' preference matrix is normalized. (However, in our work, we need to make sure that user's similarity matrix is normalized. We will discuss this in the following section.) The rightmost matrix shows the predicted user-location (user's check-ins) matrix. In this matrix, the values that are known (> 0) in the original check-in matrix are updated considering social opinions. What's more, in the check-in matrix, despite the values that are already known (> 0), there are some 0 values which represent the unknown interactions. In the prediction preference matrix, these unknown interactions are also predicted based on similarity and user's check-in histories. In brief, this multiplication process considers the opinions of all users other than the predicted one and performs the prediction process of a recommender system.

$$\underset{\text{Similarity}}{\begin{pmatrix} 1 & 0.75 & 0.25 \\ 0.75 & 1 & 0.5 \\ 0.25 & 0.5 & 1 \end{pmatrix}} \times \underset{\text{Check-ins}}{\begin{pmatrix} 3 & 0 & 4 & 3 \\ 2 & 5 & 4 & 2 \\ 0 & 5 & 0 & 3 \end{pmatrix}} = \underset{\text{Predictions}}{\begin{pmatrix} 4.5 & 5 & 7 & 5.25 \\ 4.25 & 7.5 & 7 & 6.5 \\ 1.75 & 7.5 & 3 & 4.75 \end{pmatrix}} \tag{7}$$

In general, matrix multiplication is not only a way to take social opinions into consideration, but is also a generalized foundation among all different forms of Collaborative Filtering recommendations. In other words, all recommender systems using Collaborative Filtering can be treated as the process of matrix multiplication. Moreover, in our model, we name the two matrices as the user's interest matrix and the transition matrix. The user's interests matrix shows the historical check-in information and further performs the interests of each user on locations without considering social opinion. The transition matrix represents the user-user relations, which takes the social opinions into consideration.

In the following section, we perform an example of a traditional CF recommender system to prove the generalization of matrix multiplication process.

Though there are many different forms of the recommender system, many typical systems use the idea of traditional user-based Collaborative Filtering. These systems can be reduced to two steps:

1. Look for users who share the same rating patterns with active users from historical data
2. Use the ratings from those top-k like-minded users found in step 1 to calculate a prediction for active users

An active user is user the recommender systems making recommendations to. Before these two phases, there is a pre-step of Collaborative Filtering, which is to generate user's check-in matrix according to user's historical rating data. After that, for the first step listed, this method is trying to look for users who share the same visiting pattern. In other words, this step is trying to find similar users who share similar interests which can be represented by a user's similarity matrix. In the second step, the method attempts to calculate the prediction for an active user. This, as well, can be represented by matrix multiplication where the user's check-in matrix is multiplied with the user's similarity matrix. As a result, the process of traditional Collaborative Filtering can be treated as a matrix multiplication process.

In brief, we conclude that all the models were used in a recommender system in a location-based social network as processes of matrix multiplication. The two matrices are represented by user's check-in and user's similarity information.

3.3. **Convergence of recommendation results.** In the previous section, we have shown that almost the recommendation process of all the previous Collaborative Filtering-based recommender systems can be treated as a process of matrix multiplication. What's more, based on the assumption of previous works, the predicted results generated by Collaborative Filtering recommendations hold over a long period (usually years after prediction). For example, in [11], the time interval between the last check-in of the training set and the last check-in of the test set is two years. According to the paper, there is only one fixed prediction result generated based on the training data. This assumption is reasonable when and only when the predicted results of users' preference converges to some values at some time point. Otherwise, the predicted results can only hold for a limited period and will change after that timeliness. In this section, we will prove that the recommendation results will converge to some values as long as the transition matrix is normalized and remains unchanged.

In the following section, however, we will also show that the transition matrix can be normalized but changes over time in the real world. According to the real-world data, the user's similarity matrix, which will be proved to be an approximation of the transition matrix, will change over time. This will result in a non-convergent and unpredictable user's preference. The timeliness of the predicted results is also limited. Therefore, in this paper, we propose a periodical recommendation approach which predicts users' preference in a limited timeliness.

3.3.1. *Normalization of transition matrix.* When measuring the similarity between users, the closeness between users are computed. In real-world, users' relations ranging from "completely different" to "exactly same." In traditional Collaborative Filtering recommendation, a user's similarity matrix is formed by values ranging from 0 to 1, which correspond with relations "completely different" and "exactly same." Higher values mean higher correlations or similarities between users. In our model, to aggregate user's similarity values, it is necessary to normalize them. Otherwise, it will be unfair when comparing users' closeness when using different users similarity scale. For one user, it does not make any change when his similarities with other users are normalized or not. As a result, we define a normalized similarity value n_{ij} as $n_{ij} = \frac{max(s_{ij},0)}{\sum_j max(s_{ij},0)}$. This ensures that all values will be between 0 and 1.

3.3.2. *Statement of convergence.* Suppose we have a fixed matrix P, each column of P is normalized. Then if there is another matrix R, we are 100 percent sure that $P^n R$ will converge to a vector when n is large enough. This is also known as stationary distribution in the Markov chain [14].

By definition, we know that matrix P is a normalized matrix. For values t_{ij} in matrix P, we have $t_{ij} \geq 0$ and $\sum_{j=1}^{n} t_{.j} = 1$ or $\sum_{i=1}^{n} t_{i.} = 1$(depends on which side of multiplication we are considering). In the following process, we will take $P^n R$ as an example, the result may also apply to RP^n.

The proof of this proposition can be addressed by the convergence of the Markov chain [14]. The stationary distribution of a Markov Chain with transition matrix T is some vector, ψ, such that $\psi T = \psi$. Over the long run, ψ is invariant by the matrix P [13, 15]. In a Markov chain, the transition matrix T is normalized. In other words, a normalized transition matrix will reach the stationary distribution and result in the convergence of the Markov chain. In our proposition, there is a normalized transition matrix P. The process of n times of multiplication is a representation of the Markov chain process. As a result, the results of this process will also converge.

By this proposition, we could conclude that if there is a normalized matrix, we can get a stationary distribution by multiplying this matrix n times where n is large enough. According to the proposition, the most important factor that will change the result is the normalized matrix. In the Markov chain, this matrix is known as a transition matrix. When the transition matrix is decided, this process will converge to a fixed value no matter what the original matrix is when n is large enough. In another word, the converged result of this matrix multiplication process is only determined by the eigenvalues and eigenvectors of the transition matrix.

In our system, we also have two matrices R and T, where the transition matrix T, a user-user relations in our model, is normalized as we defined before. As a result, using this proposition, we could also prove that the process of Collaborative Filtering, which has been established to a matrix multiplication process, will also converge to a value when n is large enough. If this result holds, we could also confirm that a recommender system is meaningful if and only if the result can converge to some stationary state. And we can also announce that we could generate the potential final users preference T^nR if the following factors are known: a fixed transition matrix T and the value of n. However, these two factors are unknown in practice, and other factors might influence them. In next section, we will try to change other factors that might affect the result to make it as close to the optimal result as possible.

3.4. **Impact of transition matrix.** In this section, the importance of the transition matrix will be evaluated. As introduced in previous section, the transition matrix is the key factor that will influence convergent recommendation results when it reaches the stationary distribution. In this section, first, a method that will generate a transition matrix that is as accurate as possible is proposed. After that, we will prove that it is reasonable to use a similarity matrix to approximate a transition matrix. Last, we will explain the truth that our model is better than previous ones by comparing the difference between transition matrices we generated.

We have shown that the recommendation results will converge when n is large enough in the previous subsection. In our work, there are still three unknown factors that might influence the recommendation results. The first one is the matrix being multiplied in our work, which is known as the transition matrix in Markov chain method. To solve this problem, we need to prove that the transition matrix in our work can be approximated by the similarity matrix. If the transition matrix is known, the value of n will also become known. Since for each transition matrix, the stationary matrix will only depend on the eigenvalues and eigenvectors of the original transition matrix. Another unknown factor in a system that built on real world data would be how to divide data into regions that best match each iteration when doing the multiplication.

Now we can conclude that ideally when n is large enough, the final result of preference prediction will converge to a fixed value thus you can use the similarity matrix to approximate the transition matrix. However, in the real world, this ideal situation only happens when user's interests matrix and user's relation matrix remains the same at any time. This is not held during our experiment. As a result, in our work, we make an assumption that user's interests and user's relationships change over time and due to the integrity of information. Based on this assumption, the ideal situation will not hold anymore. In next section, we will discuss the detail of changes if user's interests and relations evolve over time.

3.4.1. *Change of transition matrix.* Though it doesn't influence the convergence result of the recommendation process, the user's interest matrix somehow plays a major role in a location-based recommendation. Since user's interest generates their user's relation matrix (transition matrix in our model), the change of user's interest matrix will also result in the change in user's similarity matrix. Furthermore, the variation of this matrix will influence the matrix convergence process when n is not sufficiently large since it represents the historical data of users.

Suppose that a user's interest remain the same all the time. There will actually be a contradiction with recommender system itself since the whole system is trying to predict and explore the potential new locations that the user might be interested in. In other words, users' interest changes over time and the system is trying to mine the changing pattern of users interest.

Another factor that may also influence user's interest is the known historical check-in information of a user. In other words, depending on the known information, new visiting locations, new recommended locations by a system and so on, a user's interest may also vary. For example, a user visits a new Italian restaurant he has never tried before, and he feels that this new restaurant is better than a Chinese restaurant that he is familiar with and visited a lot of times before. In this way, this user's interest changes from "Chinese food" category to "Italian food" category since then.

To prove these two assumptions, an experiment is performed, which considers the impact of known data. In an online rating media, the number of ratings and check-in information increases by time. As a result, we can conclude that in the real world, the size of known information should increase as time goes by. In the experiment, we use this assumption and consider the recommendation rate change over time, which can also be considered as change over known data based on the hypopaper.

Fig.2 shows that precision changes over time. Precision here means the percentage of accurate recommendation results. As shown in Fig.2, the precision increases when time passes by and when the size of training data increases. As a result, we conclude that users' preference changes over time instead of remaining the same all the time.

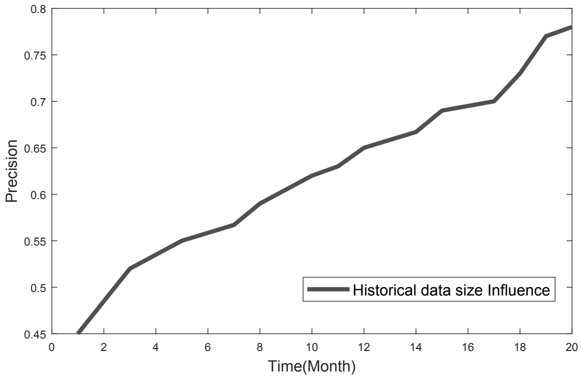


Figure 2. Influence of training size to the precision.

As we mentioned before, the transition matrix is represented by user's relations in a collaborative filtering approach. Furthermore, user's relation is represented by user's similarity matrix in most recommender systems. As a result, we will analyze the change of user's similarity to reflect the change of transition matrix in our model, proving that user's similarity changes over time is more complicated. First, we have shown that user's interest changes over time. As a result, we could also conclude that user's similarity, which is generated based on users' interests, changes over time in most cases. However, if two users change their minds in exactly the same pattern all the time, we would say that their similarity remains the same all the time. On the contrary, if there does not exist the situation that two users with the same preference change pattern, we can conclude that user's similarity also changes over time.

If we confirm that user's interest and similarity changes over time, we would also conclude that we are only predicting the user's potential preference for a period. After a period, when a user's interest and user's similarity changes, a new user's preference will appear. In other words, almost all of the recommendation results we got from previous systems or models are only valid for a certain period. As a result, we propose a novel periodical recommendation system. According to the proportion, the most important factors in our system is the transition matrix, which will influence the recommendation results.

3.4.2. *Obtaining of the accurate transition matrix.* The first problem we are going to solve here is how to generate a transition matrix that is as close to the accurate one as possible. As we introduced in the previous section, the matrix multiplication process will repeat n times before it reaches to the stationary distribution and generates a converged result. Here, we call each one out of n times multiplications as one iteration of this process. Ideally, if training data (the original matrix) and test data (the converged matrix) of one iteration is known, it is easy to generate the accurate transition matrix by division. For example, if we have a training matrix A and a test matrix B that is generated exactly from one iteration of multiplication, it is easy to get the transition matrix T by $T = BA^{-1}$. However, in the real world, even we have training data and the corresponding test data that is generated by a fixed transition matrix, it is still unclear that how many iterations it has been going through. In this case, we have the same training matrix A and a different test matrix $B^{'}$ which was generated after k iterations (the value of k is unknown). As a result, from $T^k = B^{'}A^{-1}$, where k is unknown, the accurate transition matrix cannot be generated.

One of the best ways to solve the problem of unknown n would be initializing the training matrix and test matrix within one exact iteration. In this case, assume we could represent the accurate transition matrix by our known similarity matrix. To we generate the test matrix from the training matrix, there will be three steps: 1) Based on the training data $A(m*n)$ we have, we can perform a users similarity method, which we have introduced in previous sections, to get a similarity matrix $S(n*n)$. 2) By multiplying A and S once, we will get a result matrix $\tilde{B}(m*n)$. 3) In this step, we compare matrix $\tilde{B}$ with all possible test data combinations and pick the one with highest similarity and output the corresponding matrix B. After these three steps, we will have a pairwise matrix A and B, generated by one iteration under the assumption that similarity matrix is the transition matrix.

3.4.3. *Substitute of transition matrix.* In the previous section, we have got a pairwise matrix A and B with is generated by one iteration. In this case, the transition matrix T will be equal to BA^{-1}. To further prove that a transition matrix can be represented by a similarity matrix, we can compare the closeness of matrix S and matrix T to generate the difference.

There are numbers of methods that were used in previous works to compare the closeness of two matrices. The most accurate one would compare the eigenvector and eigenvalue of two matrices since a matrix is a representation of a linear

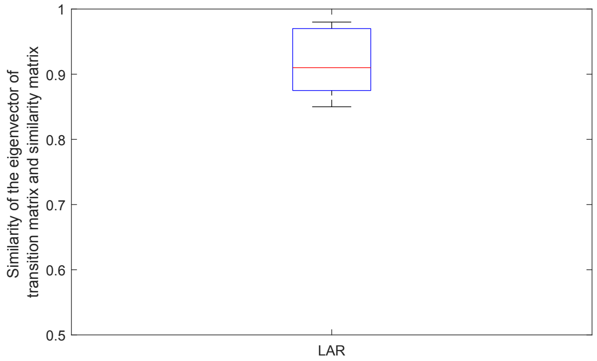


FIGURE 3. Cosine similarity between the transition and similarity matrix.

transformation. In our work, as we have proved in the previous section, the eigenvector that corresponds to the eigenvalue that equals to 1 best represents the linear transformation of the transition matrix when the matrix is normalized. In this case, comparison of two matrices could be represented by comparison of two vectors. The best way to compare two vectors is cosine similarity: Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

In Fig.3, the y axis means the cosine similarity value ranging from 0 to 1. As shown in the figure, the value is ranging from 0.87 to 0.95, which means the angle between these two matrixes is ranging from 6 to 28 degree. The average is about 7 degree, which is on average 3 degrees smaller compared with the other two previous works we considered in the evaluation section. The main reason for the difference is the way the similarity matrices were generated using historical data. In our model, we divide locations into categories based on their distance and semantic meanings, while the category information was predefined in the dataset for most of the previous works. Though other factors like considering the popularity of a location or expertise of a person will also impact the similarity matrix, category information is still one of the most influence factors that would make changes to the similarity matrix we get. In general, the similarity matrix we generated from a semantic meaning category-based method results in a more accurate transition matrix.

3.5. **Recommendation timeliness.** In the previous section, by the proof of the matrix multiplication process and convergence of recommendation results, as well as the representation of the transition matrix, we propose a novel periodical recommendation system. In this new proposed recommendation system, there is a timeliness of recommendation. In other words, the recommendation results will become highest at some time point and decrease after that time point. The time interval between the last known historical data to the highest time point is the timeliness of a recommendation. Different from previous works, where the recommendation results hold all the time, our model provides a timeliness of each recommendation and achieves higher recommendation rate.

In this section, the timeliness of recommendation will be further studied. First, we will investigate the flexibility of the timeliness by looking into the data pattern itself. After that, the detailed process of generating the recommendation timeliness will be introduced. Finally, later in the next section, we will prove that different recommendation timeliness hold for various transition matrices.

The first challenge is to answer whether the "timeliness" will be fixed or change as well as the historical check-in time change. This depends on the most recent time of the training data since the known information increases when time goes by. Since the recommendation process is a process of predicting changes, the training data itself will reflect the recommendation timeliness by the periodical changes on new location visiting. To answer this question, we looked into our data. According to our previous definition, for each recommendation iteration, there should be a short period where many newly visited locations show burst increases for each user. On the other hand, in a recommender system, the predictions are the approximation of the burst increase of newly visited locations. As a result, the growth burst time point will be the corresponding timeliness where the highest recommendation rate can be achieved.

In our data-set, we looked at some newly visited locations for all the users in two separate months. On average, there is a burst of new location visiting history for all the users every 3-6 months. As shown in Fig.4, the highest number of new location visiting is achieved in 24 months after the beginning time, and the lowest happens around ten months after the start time. Beginning time here means the time when the first check-in was made by some user in the whole data-set. Also, the time interval between each two bursts is relatively fixed with an average of about 4 months.

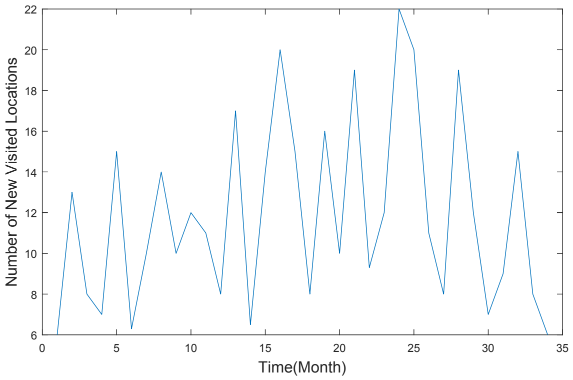


FIGURE 4. Average number of new visited location changes through each month.

In other words, in our work, there is a high probability that the most accurate recommendation timeliness is four months when applying the matrix multiplication process.

3.5.1. *Obtaining of recommendation timeliness.* The previous section has shown that the recommendation timeliness is relatively fixed and doesn't change over time and over different known data for our given historical data. In this section, we will further look into the recommendation timeliness and try to answer the question: "what is the timeliness of our recommendation where the user's preference can be predicted as accurately as possible?"

The time interval of each timeliness is evaluated in this section. The solution is to plot a CDF graph according to our CF recommendation model. As shown in Fig.5, the average timeliness shown in our data is about 3 month, which matches the burst of a number of new visited locations. We are 90 percent confident that the time interval will be no longer than six months. Now we have a promising periodical recommendation system, which takes users interest and similarity changes into consideration. In our system, when recommending a set of locations L_k to a user u_k, we are 90 percent confident that user will visit these locations L_k within the following six months. After six months, there is a high probability that the visiting probability will decrease, and new recommendations can be made based on the original historical information and the ground truth visiting information among the six months.

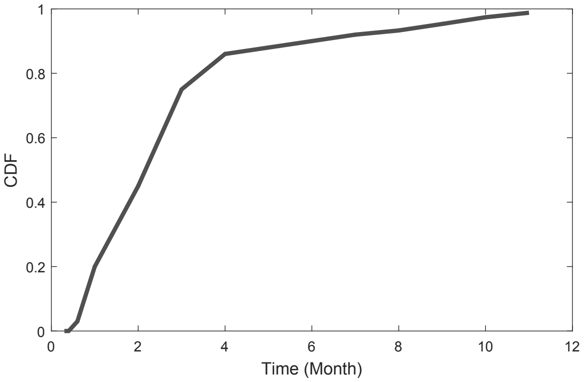


FIGURE 5. Cumulative distribution function (CDF) of time interval between timelinesss.

In conclusion, in this section, we analyzed the recommendation process of our novel recommendation approach with recommendation timeliness. 1) Matrix multiplication was first proposed and proved to be applied in most recommender systems. 2) Based on the matrix multiplication process, the convergence of recommendation result was further proposed, which was built in the Markov chain method and proved to be applied to all matrix multiplication process when the transition matrix is normalized. 3) Real world matrix convergence situation was then analyzed by considering both interest and transition matrix change. 4) After that, the primary factor that will change the convergence value, the transition matrix, was discussed. The similarity matrix was proved to be a representation of transition matrix in our model. Furthermore, the difference between three baselines and our work was discussed according to their different similarity measures. 5) Finally, the recommendation timeliness, which is the main difference between the previous works and our proposed work, was analyzed.

4. **Recommendation.** According to the conclusion we made in the previous section, we propose a novel accurate recommendation approach that uses traditional Collaborative Filtering method to make more accurate recommendations, compared to previous systems. There are several related works mentioned that time sequence of check-in information as one of the most important content information [39, 40, 51, 58]. However, none of these works pointed out the importance of

recommendation timeliness. In our approach, we first apply category classification method considering both the distance between locations and the semantic meanings of locations. Then, we use a category-based Collaborative Filtering to generate the predicted preference matrix where each value in the matrix represents the predicted value of the interest of each user to each location. As we discussed in the previous section, all CF-based recommender system can be summarized as a process of matrix multiplication, the primary goal of our approach is to identify the two matrices that represent user's interest and user's similarity. Finally, we make an accurate recommendation with a limited timeliness based on the predicted preference matrix.

There are two forms of recommendations in our approach: 1) recommend user's preference within a timeliness and 2) recommend user's preference at any time with different preference values. For example, for an active user i, by applying our approach, we will predict user i's preference in the following six months as well as his new preference in one year, two years, etc. In our approach, we are trying to answer the following two questions: 1) Is the recommendation timeliness fixed or also changes over time? and 2) How can we obtain the timeliness of each recommendation result? If these two challenges are addressed, we could then apply our accurate recommendation approach to all location-based data sets.

There are four data structures in a location-based recommendation system: user, POI, check-in, user's location history. In a location-based social network, a user u_k's profile information is unknown, and a user ID represents each user. Moreover, a user can mark a POI (e.g., a restaurant) and leave some comments, which is also known as check-in in an LBSN. A user can visit multiple locations and may generate a check-in for each of the visit. All of the users check-ins reflect her location history in the real world. Each POI is a location associated with a pair of coordinates indicating its geographical position and a hash value denoting its identification.

4.1. **Category-based Collaborative Filtering.** By Tobler's first law of geography, Points of Interest (POIs) that provide similar services are more likely to be clustered in the same geographical area. Moreover, the total number of points of interest is large, which makes the division of clusters a challenge. Also, users are most likely to visit some points of interest for a short period, and these points of interest are usually limited to some geographical regions. For example, if there are different stores in a shopping mall, a user may visit several stores in this shopping mall within an hour. Lastly, as we all know, most users' activities are in the center of cities instead of suburbs.

Previous work has proved that an individual's interests will be more semantic than specific [55, 56]. For instance, a user who is interested in food will not only focus on one specific restaurant but several restaurants in different unique locations and different categories (i.e. food - Chinese food - Chinese restaurant). A user's preferences span multiple interests instead of binary decisions. As a result, we propose a category-based CF by considering the category of POIs in a location-based social network. The most commonly used method to solve a category problem is clustering. In our model, two factors of clustering are being considered, geographical and semantic meaning. In a location-based social network, POIs can be clustered based on their geographic locations since POIs that are near each other are more likely to share similar interests to users. For example, there are different types of stores in a shopping mall, which share the same interest type "shopping." Another factor that will also influence the clustering is semantic meanings. That is because locations that are different in geography may share the same semantic meaning. For example, two similar or exactly same grocery stores located in two separate streets of a city.

In our approach, we use both K-means clustering and Singular Value Decomposition (SVD) to category POIs into different clusters. In our example, we assume that the geographical locations have been grouped into $\mathbb{R}$ latent regions and denoted as R. In this paper, we name these latent regions as neighborhoods N. To cluster POIs into neighborhoods N, in this paper, we combine two clustering method: K-means clustering and SVD. K-means clustering is used first to divide the whole spatial region into clusters considering the geographical distance influence. Secondly, the semantic meaning of locations is being considered by using SVD method. After this step, clusters that share the similar semantic meaning are further classified to make a large neighborhood N. As a result, instead of using POIs themselves, the interests of users are performed on each neighborhood N based on the visiting histories. In other words, after the neighborhoods have been identified, user's interest matrix will be compressed from user-location $(m * n)$ matrix to user-neighborhood $(m * n^{'})$ matrix.

There are two advantages of this compression: 1) the data sparsity, which is one of the largest problems in CF systems, is partly solved, where $n^{'} < n$; 2) users' similarities can be more general, and the problem of finding friends for users that have limited visiting history is also solved. A category-based Collaborative Filtering can be performed using the user-neighborhood-based user's interest matrix and further in user's similarity matrix.

4.2. **Approximation of transition matrix.** The previous section has proved that a user's similarity matrix can be an approximation of the transition matrix in a recommender system. According to Collaborative Filtering, recommender systems in location-based social networks need to take the location history of the target user and the relationships between users into consideration [4, 26, 47, 60]. In other words, a recommender system in a location-based social network work need to consider the following two factors: 1) User's interest: the knowledge of a user that was reflected from his or her historical visiting data. For example, food hunters may be more interested in high-quality restaurants, while movie lovers would pay more attentions to nearby theaters. 2) User's similarity: the similarities between the target user and other users based on the opinions of locations given by all users in the network. The views of other users, especially the users that share similar interests, are usually a valuable resource for recommendations. As a result, in a recommender system, when performing a recommendation, we often select candidates that share interests. These types of relations between users are shown by a user's similarity matrix.

To learn about geographical user's interest, we need a model to encode the spatial influence and user mobility into the user check-in decision process. Unlike the traditional online social networks or other rating services, in a location-based social network, the interests of a user to a location can be measured by the implicit feedback. Hence, to produce recommendations, several studies [7, 36, 45] use traditional recommendation algorithms to infer users interests for locations by mining the check-in frequencies of users.

The check-in frequencies of users somehow reflect the visiting patterns and further the preferences of users. As a result, the rating of a user to a location can be divided to "positive" and "negative" by setting a threshold. With that, existing approaches can be employed for location-based recommendation by representing locations by items using both User-based Collaborative Filtering [6] and Item-based Collaborative Filtering [20]. In these approaches, a Collaborative Filtering method is directly used for recommendation. As a result, in general, we use the number of check-ins in a neighborhood to measure user's preference. That is based on the assumption that people will be more likely to visit their favorite places and POIs.

We consider the impact of both geographic distance impact and popularity impact. 1) Instead of using user-location matrix directly, we use a user-neighborhood matrix, which takes geographic range impact into consideration. 2) We weight the user-neighborhood matrix by total number that a neighborhood has been visited, which takes the popularity of each POI into consideration (e.g. location or neighborhood with a higher popularity will have a larger number of visiting records). By doing so, the final predicted user-preference matrix will contain both the opinions of users and impact of multiple influence factors.

In most existing work, users' relationships are either friends or strangers. In the data we used for evaluation, friendships are also binary relations. That makes it unclear that whether the closeness of friendship is a factor that will influence user's choice on POI. According to our assumption, there is a significant chance that friends' opinion will cause a user to make different different choice. To make it clear, in this paper, we add a weight value to each friendship by compare number of shared locations of two users. This is based on the assumption that people in a particular area who share the same interests will potentially be friends with each other.

As a result, there will be three different types of relationship in this location based social network: "friend", "potential friend", and "stranger". For users that are already friends of each other, their closeness can be measured by their shared interests; For users not related to each other, we will compare the size of the intersections of their interests: for users shared interests number larger than a parameter k, we call them potential friends; and for users shared smaller number of interests, we assume they are strangers. In this way, we measure users' similarity by their closeness. Cosine correlation has been proved to be one of the most significant methods to calculate users similarity value. To evaluate that, we test 5 different ways of measuring the similarity between two vectors. We will discuss more detail in the results section.

In detail, the process of generating a user's similarity matrix is as follows:

1) User's preference vectors. The first step is to generate the preference vectors among the categories for all users. As introduced in previous section, the values of each preference vector are represented by the number of check-in histories of each user. For example, for two users u_i and u_j, the preference vectors are $v_{u_i} = \{3, 9, 10, 5, 1\}$ and $v_{v_i} = \{5, 1, 2, 15, 3\}$.

2) Vector weights. In the last step, the vectors from the previous steps are used to convert all history information to a numerical format. This is done by representing the interests of each user to each location by the check-in frequency in step 1. However, to consider the influence factors, the values in the vector needs to be further weighted by a weighting method.

The weighting procedure we used in this paper is the Term Frequency and Inverse Document Frequency (TF-IDF) weighting approach. This approach uses two scores, i.e. the term frequency score and the inverse document frequency score. Intuitively, a user would visit more locations belonging to a category if the user likes it. Further, if a user visits locations of a category that is rarely visited by other people, the user could like this category more prominently. For example, the number of visits to restaurants is more than other categories like museums in people location histories. It does not mean the food is everyone's primary interest. However, if we find a user visits museums very frequently, the user may be truly interested in arts or history.

After this, we will get a weighted preference vector for each user.

3) User's similarity. In this step, we calculate the similarity between each two pairwise users by using cosine similarity. The cosine similarity is defined as follows:

$$(sim_{u,v})^{cosine} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}} \tag{8}$$

In this way, user's similarity matrix is finally generated by the cosine similarity values. According to the assumption we made the previous section, we further weighted this similarity matrix by normalizing each role.

4.3. **Generation of recommendation results.**

4.3.1. *Generating user's interest matrix.* In this step, a weighted user-neighborhood matrix is generated. The values in user's interests matrix are first represented by check-ins each user has performed in each neighborhood. After that, the popularity of each neighborhood is considered. As a result, a weighted user's interest is shown in equation 9,

$$w_{ic} = \frac{|\{u_i, n_c\}|}{|V|} \tag{9}$$

where $|\{u_i, n_c\}|$ is user i's number of visit in neighborhood c and V is total number of visits.

As a result, the weighted value captures a users interests well, having the following advantages: 1) reduce the concern raised by the different data scales of different users, 2) handle the data sparsity problem and reduce the computational loads for further user similarity computing (from physical locations to categories), and 3) enable the computing of similarity between users who do not share any location histories, e.g., living in different cities. After that, we will have a user's interest matrix $R \in \mathbb{R}^{m*n}$ where each value w_{ic} representing the weighted number of interests of user i to neighborhood c.

4.3.2. *Computing user's similarity matrix.* In the previous section, an un-normalized user's similarity matrix has already been generated where larger values mean higher closeness between users. However, in our model, to aggregate users similarity values, it is necessary to normalize them. Otherwise, it will be unfair when comparing users closeness when using different users similarity scale. For one user, it doesn't make any change when his similarities with other users are normalized or not. For example, the similarity vectors of two users are $v_i = [0.5, 0.5, 0.3]$ and $v_j = [0.9, 0.8, 0.7]$. Before normalization, the scales of similarities are different for these two users. The normalized two user vectors are: $n_{v_i} = [0.385, 0.385, 0.23]$ and $n_{v_j} = [0.375, 0.0.333, 0.0.292]$. In this case, the scale problem can be solved. On the other hand, for one user, the closeness between other users and this target user remain unchanged. As a result, we define a normalized similarity value s_{ij} as

$$s_{ij} = \frac{max\,(sim_{ij}, 0)}{\sum_j max\,(sim_{ij}, 0)} \tag{10}$$

where sim_{ij} is the similarity between user i and j. That is, two users are more likely to be similar if they share more nodes with a bigger interest weight. A user's similarity matrix $S \in \mathbb{R}^{m*m}$ is generated where each value s_{ij} representing the similarity between user i and user j.

4.3.3. *Generating predicted user's preference matrix.* Generating the predicted user's preference matrix is a process of matrix multiplication of user's interest and user's similarity. As a result, the predicted matrix $P \in \mathbb{R}^{m*n}$ can be computed as:

$$P = R * T \tag{11}$$

Finally, the system returns the full user-neighborhood matrix with the user's preference scores to all the neighborhoods as the location recommendations.

4.4. **Timeliness of recommendation results.** As we discussed in previous sections, there is a timeliness for the recommendation results we generated. That is, user's interest and similarity will change after the timeliness, which will result in a new user's preference. In other words, the main difference between our system and the previous works would be we propose a recommendation system considering the recommendation timeliness. A new predicted recommendation result and the corresponding timeliness of that result will be generated periodically.

4.4.1. *Obtaining timeliness of each recommendation.* As we introduced in the previous section, with an empirical CDF, the timeliness can be generated with a 90 percent of confidence. That is, for a recommendation result P_a that is generated at time a, the corresponding timeliness will be t_a. We are 90 percent confident that the recommendation rate will be decreased after this timeliness. After this timeliness, since user's interest changes, the recommendation results will be obsoleted, and a new recommendation result will be generated.

4.4.2. *Generating new predicted recommendation.* From the time point that the last check-in record was made to the time point that the recommendation result is obsoleted, both user's interest and user's similarity will change in the real world. A new recommendation result will be generated using both historical data and predicted data. First, the predicted user's preference matrix P_{a-1} from the last iteration will be the latest user's interest matrix R_a. Second, based on the new interest matrix R_a, a new similarity matrix S_a will be obtained following the same step of the previous section. As we proved before, a similarity matrix, which is the transition matrix, placed a significant role when generating the final multiplication result. By applying the matrix multiplication process using these two new matrices, a new recommendation result P_{a+1} will be generated.

By repeating the above two steps, a periodical recommendation system considering the recommendation timeliness is proposed.

4.5. **Top-k_i recommendation.** After applying Collaborative Filtering in the last section, the preference distribution through users on locations in the whole network is provided. In this case, most previous works choose the method of top-k recommendation. The idea of top-k recommendation is simply to choose k locations that have the highest preference weight to make the recommendation. Although the use of this method has worked well in different previous works, it has some deficits, mainly due to the difference of users' probability distributions. For example, for two users i and j, the distributions of their preferencces on three locations are as follows: $D(i) = \{0.7, 0.2, 0.1\}$ and $D(j) = \{0.4, 0.3, 0.3\}$. In this case, it is unscalable to use the same top-k recommendation for both of these two users.

To this end, in our work, we will provide a novel recommendation method that takes the scale problem into consideration which called Sum-top-k_i recommendation. In a Sum-top-k_i recommendation, k_i is determined by each user's distribution. For each user i, we will recommend the top-k_i locations which have a sum of a fixed number n. For the two users in the previous example, we have $k_i = 1$ and $k_j = 2$. Therefore, we will recommend the first location to user i and the first two locations to user j.

In conclusion, in this section, we propose a novel accurate recommendation model (location-based accurate recommendation) which use traditional Collaborative Filtering method to make an accurate recommendation compared with previous systems. 1) By applying category-based Collaborative Filtering method, the system can achieve higher accurate recommendation results. The category information is generated by using K-means clustering and SVD, which takes both the geographical influence and semantic meaning into consideration. 2) The approximation of the transition matrix, which is the most important matrix to change a recommendation result, is also approximated by comparing with user's similarity matrix. 3) User's interest matrix and user's similarity matrix is then computed. The predicted user's preference matrix will be generated using matrix multiplication. 4) After that, recommendation timeliness, which is the main difference between the previous works and our proposed work, was analyzed. 5) At last, a novel top-k_i recommendation method is proposed to perform the recommendation based on the predicted results.

5. **Experimental results and analysis.**

5.1. **Data set.** The dataset we use is from Brightkite: Brightkite was once a location-based social networking service provider where users shared their locations by checking-in. Users built relationships in the network classifying by either friends or strangers. When a user posted a piece of check-in information, we consider that the user has checked in the POI physically. They have also collected a total of 4,491,143 check-ins of users over the period of Apr. 2008 - Oct. 2012. The network is originally directed, but they have constructed a network with undirected edges when there is a friendship both ways. The friendship network is undirected and was collected using their public API, and consists of 9,591 nodes and 90,327 edges.

In this dataset, check-in information is formed by user ID, timestamps, coordinate and a hash function of the location. The required data format is shown as follows.

$UserID$: [number]
$Time$: [year-month-day HH:MM:SS]
$Coordinate$: [longitude]
$Coordinate$: [latitude]
$Location$: [32-bit hash value]

In such data, it is possible to apply geometry distance between locations since it contains the coordinate information. We assume that most of the locations clustered in a particular area belong to the same or similar category in a location-based social network. This assumption is from Tobler's first law of geography: 'Point of Interests with related services are likely to be clustered in the same geographical area.' In this case, by calculating the distance between locations, we can divide locations into different clusters while each cluster represents a category. Moreover, the friendship information is also available as binary values. In location-based social networks, assume each user u has some check-in histories in real life locations l. Here location

l is represented by a coordinate in realistic geography points. Initially, we treat some visit of a user to a location as rating l for that particular location.

Gowalla was another location-based social network service launched in 2007 and closed in 2012. The dataset that was collected from Gowalla contains 6,442,890 check-ins. The coordinate information of each location is also available in Gowalla. We performed the traditional CF on the Gowalla data set and got similar precision and recall results as we generated from Brightkite. However, since the check-in number is larger and the time range is longer, Brightkite outperforms Gowalla when applying our LAR model. Foursquare is another widely used open source data that has been commonly used in recommender systems. The semantic meaning of locations is known based on categories information. However, each location is represented by a location ID instead of coordinate information, which makes the consideration of geographical influence impossible. As we introduced before, the category of locations can be computed when considering both geographic influence and semantic meaning. As a result, in this paper, we choose Brightkite over Foursquare for our experiments.

5.2. **Evaluation metrics.** Recommender systems in LBSNs have typically used two methods to evaluate the effectiveness of their recommendations, 1) precision and recall ratios and 2) Top-k recommendation. We have introduced the special Top-k_i recommendation in the previous section, the precision and recall ratios will be discussed in this section.

Precision and recall ratios are also used to evaluate the effectiveness of recommendations in LBSNs. To use this evaluation method, a users location history is divided into two parts, 1) the location history generated within a query area, which is used as ground truth, and 2) the rest of the users location history, which is used as a training set to learn the users preferences and build the recommendation model. The system is then evaluated by whether it can suggest those sites within the querying region that the user has visited based on the training data (the location history outside of the query region).

$$\text{Precision} = \frac{\text{number of recovered ground truth}}{\text{total number of recommendations}} \tag{12}$$

$$\text{Recall} = \frac{\text{number of recovered ground truth}}{\text{total number of ground truth}} \tag{13}$$

5.3. **Benchmarks.** We compare our method with the following three benchmark approaches, where the first three baseline approaches are existing recommender systems and the fourth one “LAR” means our method.

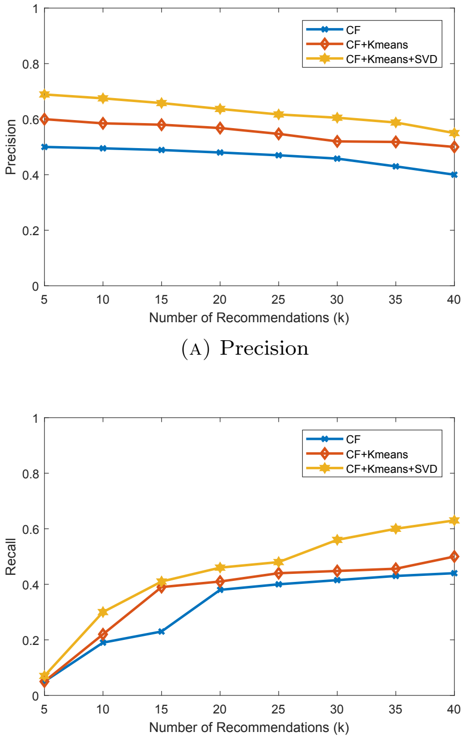
5.3.1. *Traditional location-based Collaborative Filtering (CF).* Traditional Collaborative Filtering (CF) is the most commonly used approach, which applies the collaborative filtering method directly over the locations. This benchmark utilizes the users location histories in an area with a user-location matrix and uses the traditional user-based CF method to make recommendations. The Cosine similarity between two users location vector is employed as the similarity between the two users. Finally, the locations with a high score considering both users preference and users similarity will be recommended.

5.3.2. *Location-based Geo-Social (LGS).* This benchmark is one of the previous works that comes with the highest precision result. They first measure the social knowledge of both the locations and the users by considering the popularity of a location and the expert of a user. Locations that were visited by a number of experts are considered as attractive locations. On the other hand, users that have visited a number of popular locations are treated as experts. Users preference in this baseline was computed by TF-IDF value. Intuitively, a user would visit more locations belonging to a category if the user likes it. Further, if a user visits locations of a category that is rarely visited by other people, the user could like this category more prominently. Users similarity in this baseline was calculated by comparing the expert score of users and the popularity of the corresponding locations that were visited by these users. Finally, the recommendation was made by the matrix multiplication method.

5.3.3. *GeoSoCa.* First, in GeoSoCa, a kernel estimation method was proposed with an adaptive bandwidth to determine a personalized check-in distribution of POIs for each user that naturally models the geographical correlations between POIs. Then, GeoSoCa aggregates the check-in frequency or rating of a users friends on a POI and models the social check-in frequency or rating as a power-law distribution to employ the social correlations between users. Further, GeoSoCa applies the bias of a user on a POI category to weigh the popularity of a POI in the corresponding category and models the weighed popularity as a power-law distribution to leverage the categorical correlations between POIs. Finally, they conduct a comprehensive performance evaluation for GeoSoCa using two large-scale real-world check-in data sets collected from Foursquare and Yelp.

5.4. **Impact of recommendation process.** In this section, three factors that will influence the recommendation results, as well as the recommendation process, is analyzed. 1) The impact of categories on user's interest matrix will be first evaluated. In our proposed LAR, categories of locations will impact user's interest matrix and further the recommendation result. 2) The impact of the transition matrix, which is the most important factor, will then be evaluated. According to the experimental results, the change of the transition matrix will impact the recommendation results. What's more, the difference of the results of the three benchmarks is mainly caused by the difference of transition matrix. 3) At last, the recommendation timeliness will be analyzed.

5.4.1. *Impact of categories on user's interest matrix.* To prove that category-based CF can perform better recommendation results than traditional CF method, we categorized locations using K-means and SVD and applied the category results to regular CF. In this section, precision and recall ratios are performed without considering the timeliness of recommendation results. We partition the check-in data temporally into training/test split by time. The recommendation results will be compared with the test set, which will generate the precision and recall ratios. For traditional CF, the user-location-based user's interest matrix will be used. However, for the other two methods, either K-means itself or both K-means and SVD will be used to category locations into groups. After categorization, the interest matrix will be compressed from a user-location matrix to a user-category matrix. The new user-category-based user's interest matrix will then be used to the CF model and come out with the recommendation results. The detail is shown in Fig.6.



(a) Precision

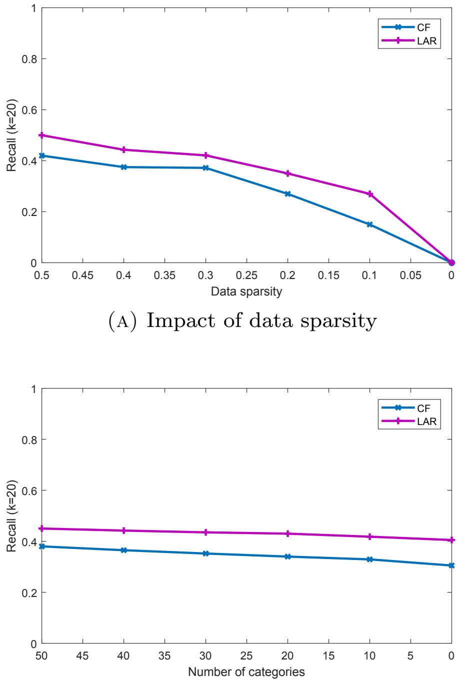
(b) Recall

Figure 6. Precision and recall of CF with or without K-means clustering and SVD.

Fig.6 shows the average precision and recall of different methods varying in the number of recommended locations (k). The precision ratios are shown in Fig.6(a) and the recall ratios are shown in Fig.6(b) Clearly, our method which considers both K-means and SVD outperforms traditional CF significantly.

The blue lines show the precision and recall results of CF without considering category information. Red lines are the results of CF taking K-means, which is one of the most commonly used geographically clustering method, into consideration. Better results of red lines mean the geographical influence will result in better recommendation results. This can also be proved by the Tobler's first law that we have introduced before. To further perform the impact of geographical distance, we experiment the recommendation difference over the different distance between locations. What's more is that the yellow lines, which further take SVD into consideration, outperforms the previous two baselines. This is because that, applying SVD to a traditional CF will lower the sparsity of data as well as clustering locations based on their semantic meanings. Data sparsity, which is one of the most challenging problems of CF models, is partly solved by using SVD. To prove this, an experiment of the impact of data sparsity was performed, and the results are shown in Fig.7.

As shown in Fig.7(b), the recommendation results increase when the impact of a category has been taken into consideration. And the impact of data sparsity is shown in Fig.7(a).



(A) Impact of data sparsity

(B) Impact of category

Figure 7. Influence of data sparsity and category to the recommendation rate (recall ratio) where number of recommendations $k = 20$

Moreover, different category measures also result in various recommendation results. For the three benchmark approaches, three distinct category measures are used. In a traditional memory-based CF model, the similarity between users is obtained by comparing similar locations. In LGS and GeoSoCa, a predefined category information is used. In this predefined category, locations with the same semantic meanings are clustered as the same category. These different category measures will result in different transition matrix and further different recommendation results. The difference between the transition matrix generated by the three benchmarks will be compared in the following sections.

5.4.2. *Impact of transition matrix.* Since the recommendation results will reflect the accuracy of similarity matrix, we could test the change of similarity based on the generated recommendation rates by dividing the data into training data and test data. As a result, to prove the change of similarity matrix, we examined the recommendation results in two different situations: 1). When user's similarity remains constant (The same situation as the ideal one) and 2). When user's similarity changes periodically. The results of these two cases are shown in Fig.8.

As shown in Fig.8, in general, the recommendation rate (precision) increases when many recommendations k increases. When we use a constant similarity matrix, the recommendation rate increase at the beginning but decreases after a period of time. However, when the dynamic similarity matrix is used, the recommendation

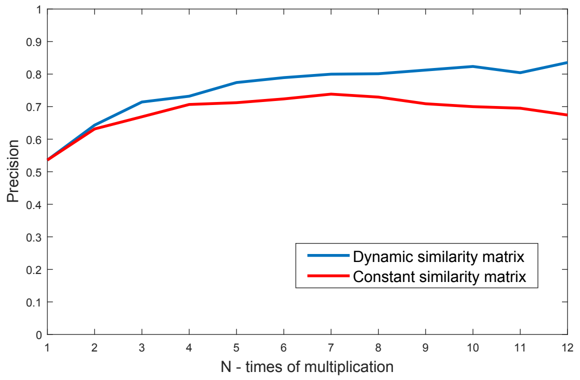


FIGURE 8. Influence of constant and dynamic similarity to the recommendation rate (precision ratio) where N is number of multiplications.

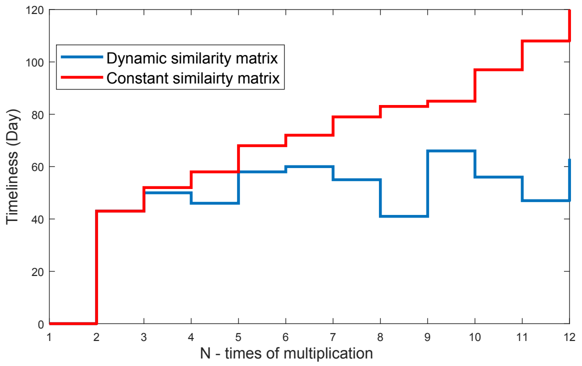


FIGURE 9. Recommendation timeliness comparison for constant similarity matrix and dynamic similarity matrix.

rate increases over times of multiplication. The results show that the recommendation rate increases when we assume that the similarity changes over time and that matches the result we got from Fig.6. Fig.6 indicates that the recommendation rate increases over time. Therefore, we conclude that users similarity also changes over time instead of remains the same all the time.

Moreover, in traditional recommender systems, as we introduced before, user's preference is considered a constant matrix. As a result, those systems predict a fixed user's preference value by combining possible factors that may impact the result. To further prove that user's preference changes over time, we divided our data into 12-time intervals. We separately use constant and dynamic user's similarity matrixes to do the matrix multiplication.

Fig.9 shows the recommendation timeliness difference when using a constant or dynamic matrix. When user's similarity remains constant, the recommendation timeliness increases exponentially and ranging from 40 to 120 days over times of multiplication. On the other hand, the recommendation timeliness is nearly constant and ranging in the same range from 40 to 70 days when using dynamic similarity matrix. As a result, we can conclude that for the same data sets, the recommendation timeliness is relatively constant when using dynamic similarity matrix, which is also the real world situation according to our experiment.

As we introduced in the previous section, the user's similarity matrix in our model outperforms the similarity matrix in other related works and is highly similar to the accurate transition matrix. In this section, we will compare the similarities between our similarity matrix and the similarity matrix we get from the three benchmarks. The results are shown in Fig.10:

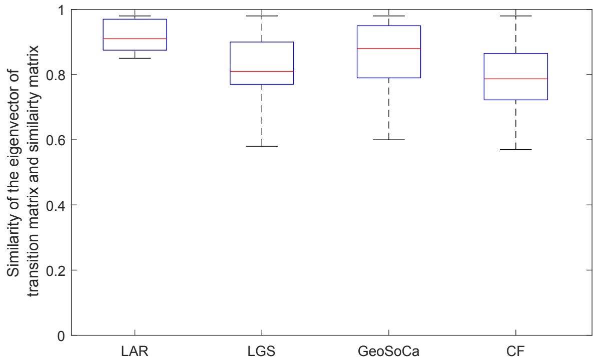


Figure 10. Similarity of the eigenvector of transition matrix and similarity matrix comparisons of our method and the three benchmarks.

Fig.10 shows the similarity of the eigenvectors, which corresponding with the eigenvalue equals to 1, of transition matrix and similarity matrix comparisons of our method and the three benchmarks. As we proved in previous sections, the convergence result of recommendation highly depends on this eigenvector. What's more, the accurate transition matrix can be obtained by accurate data division. As a result, by comparing the cosine similarity of transition matrix and similarity matrix, we will be able to predict the performance of different recommendation systems.

As shown in Fig.10, our model outperforms the other three benchmarks. On average, the similarity between the accurate transition matrix and the similarity matrix of our model is relatively higher than other three benchmarks. The similarity matrix of LGS and CF are similar since both of them generate a similarity matrix using memory-based data.

5.4.3. *Analysis of recommendation timeliness.* The following figure (Fig.11) shows the recommendation rate (precision) of our method and the three baselines varying in the recommendation timeliness. As shown in Fig.11, all these three methods show a peak of value when the period is at some time before the eight months. it can be significantly concluded that for all the methods we are considering, the periodical recommendation results are better than the traditional recommendation results. As shown in the figure, the max recommendation rate for our method appears at 3.5 months after the time the recommendation was made. And the time for LGS and GeoSoCa are four and half months and six months after the time the recommendation was made. In another word, there is a difference between the time periods for the different method when applying to the same dataset. However, we can conclude that in general, periodical recommendation works better than traditional recommendation among our model and the three benchmarks we selected.

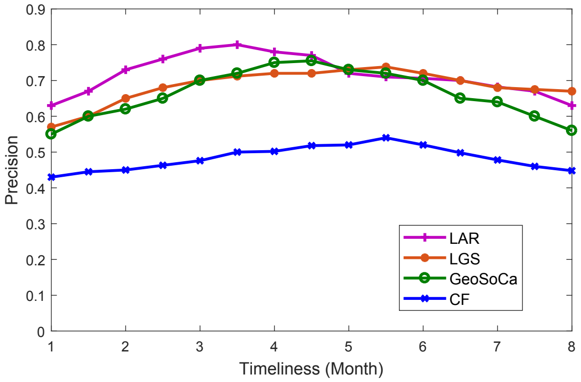


FIGURE 11. The recommendation rate of our method and the three baseline varying in the recommendation timeliness (month).

Fig.12 shows the CDF of the timeliness of our method and the three baselines except CF varying in time (month). As shown in the figure, we are 90 percent confident that the recommendation time interval of our method, LGS and GeoSoCa are four months, five months, six months from the time the recommendations were made. These results match with the results generated from Fig.5.6, the recommendation rate of our method, LGS and GeoSoCa reaches the peak at 3.5 months, 4.5 months and six months from the time the recommendations were made.

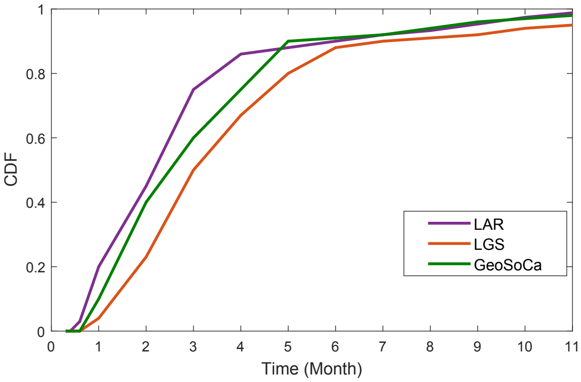
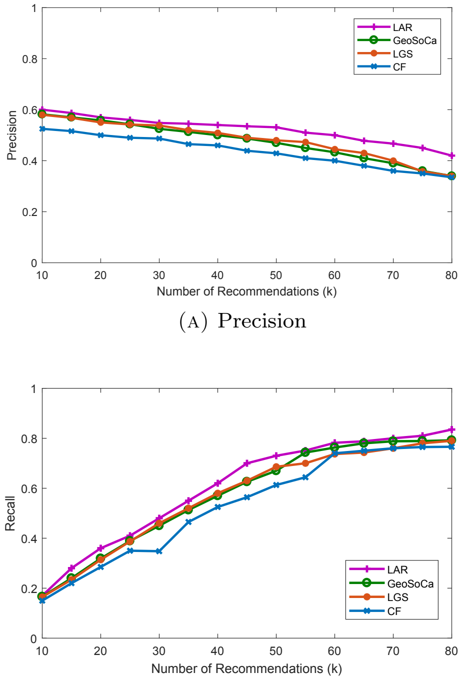


FIGURE 12. The empirical CDF of time intervals of our method and the three benchmarks except CF varying in the recommendation period time (month).



(A) Precision

(B) Recall

Figure 13. Precisions and recalls of LAR, CF, LGS and GeoSoCa.

5.5. **Effectiveness of accurate recommendation.** The following two figures (Fig.13(a) and Fig.13(b)) show the average precision and recall of different methods varying in the number of recommendations (k) considering the same recommendation timeliness (4 months). Our method outperforms benchmark approaches. Another obvious result is that all methods, including LAR and all other benchmarks of collaborative filtering fail to outperform the CF benchmark. First, the traditional CF drops behind other three methods, showing the advantage of using model-based CF as well as taking other influence factors into consideration, since all the other three models except CF are using model-based CF and taking other influence like popularity into consideration. Second, our model performs well for the recall value, justifying the benefit brought by considering category information based on the location and distance information, as well as the semantic meaning of locations. Third, our method outperforms all other baselines, which was benefit from our similarity matrix, which is more similar to the relatively accurate transition matrix as we discussed before. That is, according to our previous similarity matrix comparison results, the results in Fig.13 here also shows the importance of transition matrix.

Table 2 summarizes the best and average precision and recall for the three benchmark approaches. We can conclude that the LAR models perform the best with our dataset with very high prediction accuracy. Also, the geographic information captures the users check-in behavior relatively well and friendship connections have influence on users check-in preference.

Table 2. A summary of precision and recall ratios of the four approaches

Models	Precision	Recall	Avg.Precision	Avg.Recall
CF	0.51	0.69	0.3573	0.6395
LGS	0.59	0.77	0.3682	0.6762
GeoSoCa	0.58	0.76	0.3657	0.6684
LAR	0.62	0.83	0.4582	0.7523

In conclusion, Figure 13 and Table 2 summarizes the best precision, recall for our model and the three benchmark approaches varying in some recommendations considering the same timeliness. There is an improvement of our model LAR on both the prediction result accuracy and transition matrix compared with the three benchmark approaches.

6. Conclusions.

In this paper, by understanding the matrix multiplication process of CF-based recommendations and combined this process with Markov chain, we proved the radiation of recommendation results. Moreover, in contrast to previous recommendation systems, recommendation timeliness was proposed and applied to all CF-based recommendations.

This paper then presents a Location-based Accurate Recommendation (LAR) approach, which provides users with location recommendations around the specified geolocation based on 1) the users interests matrix learned from her location history and 2) transition matrix based on users who could share similar interests. This recommendation system can predict user's preference not only with a high accuracy but also with timeliness. By taking advantage of the category information of a users location history using both K-means clustering and SVD, our system overcomes the data sparsity problem in the original user-location matrix, and also taking both distance influence and locations' semantic meaning into consideration.

We evaluated our system using extensive experiments based on a real data set collected from Brightkite. According to the experimental results, our approach significantly outperforms some existing location recommendation methods (CF, LGS, and GeoSoCa), i.e., 83.5 percent precision ratio and 62 percent recall ratio within 4-month timeliness. These results outperform the state-of-art results that were generated by other works, i.e., 77 percent precision ratio and 59 percent recall ratio without considering recommendation timeliness. The results also justify each component proposed in our system, e.g., taking into account location history of others, change of transition matrix, category-based preference modeling, user similarity measures, and CF-based inference.

Acknowledgment. We thank the National Science Foundation (NSF) for supporting this work through grant NSF CNS-1644348.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans Knowl Data Eng*, 734–749.

[2] Y. Arase, X. Xie, M. Duan, T. Hara and S. Nishio, A game based approach to assign geographical relevance to web images, *Proceedings of the 18th International Conference on World Wide Web*, (2009), 811–820.

[3] Y. Arase, X. Xie, T. Hara and S. Nishio, Mining people's trips from large scale geo-tagged photos, *Proceedings of the International Conference on Multimedia*, (2010), 133–142.

[4] M. Attarl, M. Manguoglu, I. Toroslu H, P. Symeonidis, P. Senkul and Y. Manolopoulos, Geo-activity recommendations by using improved feature combination, *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 996–1003.

[5] L. Backstrom and J. Leskovec, Supervised random walks: Predicting and recommending links in social networks, *ACM*, (2011), 635–644.

[6] J. Bao, Y. Zheng and M. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, *SIGSPATIAL '12 Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, (2012), 199–208.

[7] B. Berjani and T. Strufe, A recommendation system for spots in location-based online social networks, *Proceedings of the 4th Workshop on Social Network Systems*, (2011), Article No. 4.

[8] D. Brockmann, L. Hufnagel and T. Geisel, The scaling laws of human travel, *Nature*, **439** (2006), 462–465.

[9] J. Cai, Z. Li, M. Yan and Y. Li, Using crowdsourced data in location-based social networks to explore influence maximization, in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, IEEE, 2016, 1–9.

[10] X. Cao, G. Cong and C. Jensen, Mining significant semantic locations from gps data, *Proc VLDB Endowment*, **3** (2010), 1009–1020.

[11] R. Cheng, J. Pang and Y. Zhang, Inferring friendship from check-in data of location-based social networks, *ASONAM*, (2015), 1284–1291.

[12] C.-Y. Chow, J. Bao and M. Mokbel, Towards location-based social networking services, *ACM SIGSPATIAL International Workshop on Location Based Social Networks*, (2010), 31–38.

[13] M. Cowles Kathryn and B. Carlin P, Markov chain monte carlo convergence diagnostics: A comparative review, *Journal of the American Statistical Association*, **91** (1996), 883–904.

[14] W. Gilks, S. Richardson and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice,* Interdisciplinary Statistics. Chapman & Hall, London, 1996.

[15] W. Gilks, S. Richardson and D. Spiegelhalter, Studying convergence of markov chain monte carlo algorithms using coupled sample paths, *Journal of the American Statistical Association*, **91** (1996), 154–166.

[16] L. Herlocker J, A. Konstan J, A. Borchers and J. Riedl, An algorithmic framework for performing collaborative filtering, *SIGIR*, 230–237.

[17] K. Kodama, Y. Iijima, X. Guo and Y. Ishikawa, Skyline queries based on user locations and preferences for making location-based recommendations, *ACM*, (2009), 9–16.

[18] X. Li, Q. Yang, X. Lin, S. Wu and M. Wittie, itrust: Interpersonal trust measurements from social interactions, *IEEE Network*, **30** (2016), 54–58.

[19] Y. Liang, Z. Cai, Q. Han and Y. Li, Location privacy leakage through sensory data, *Security and Communication Networks*, **2017** (2017), Article ID 7576307, 12 pages.

[20] G. Linden, B. Smith and J. York, Amazon.com recommendations: Item-to-item collaborative filtering, *IEEE Computer Society*, **7** (2003), 76–80.

[21] G. Liu, Q. Chen, Q. Yang, B. Zhu, H. Wang and W. Wang, Opinionwalk: An efficient solution to massive trust assessment in online social networks, in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, IEEE, 2017, 1–9.

[22] G. Liu, Q. Yang, H. Wang, X. Lin and M. P. Wittie, Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic, in *INFOCOM, 2014 Proceedings IEEE*, IEEE, 2014, 1698–1706.

[23] G. Liu, Q. Yang, H. Wang, S. Wu and M. P. Wittie, Uncovering the mystery of trust in an online social network, in *Communications and Network Security (CNS), 2015 IEEE Conference on*, IEEE, 2015, 488–496.

[24] C.-T. Lu, P.-R. Lei, W.-C. Peng and I.-J. Su, A framework of mining semantic regions from trajectories, *International Conference on Database Systems for Advanced Applications*, (2011), 193–207.

[25] A. Noulas, S. Scellato, C. Mascolo and M. Pontil, Exploiting semantic annotations for clustering geographic areas and users in location-based social networks, *Fifth International AAAI Conference on Weblogs and Social Media*.

[26] A. Noulas, S. Scellato, N. Lathia and C. Mascolo, Mining user mobility features for next place prediction in location-based services, *IEEE 12th International Conference on Data Mining*, (2012), 1038–1043.

[27] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim Joon and S. Chong, On the levy-walk nature of human mobility, *IEEE/ACM Transactions on Networking (TON)*, (2008).

[28] L. Rossi and M. Musolesi, It's the way you check-in: Identifying users in location-based social networks, *COSN*, (2014), 215–226.

[29] J. Sang, T. Mei and C. Xu, Activity sensor: Check-in usage mining for local recommendation, *ACM Transactions on Intelligent Systems and Technology (TIST)*, **6** (2015), Article No. 41.

[30] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, Item-based collaborative filtering recommendation algorithms, *WWW '01 Proceedings of the 10th International Conference on World Wide Web*, (2001), 285–295.

[31] M. Sarwat, J. Levandoski J., A. Eldawy and M. Mokbel F., Lars*: An efficient and scalable location-aware recommender system, *Transactions ON Knowledge and Data Engineering*, **26** (2014), 1384–1399.

[32] X. Su and K. Taghi M., A survey of collaborative filtering techniques, *Advances in Artificial Intelligence*, **2009** (2009), Article ID 421425, 19 pages.

[33] Z. Su, Y. Hui and Q. Yang, The next generation vehicular networks: A content-centric framework, *IEEE Wireless Communications*, **24** (2017), 60–66.

[34] Z. Su, Q. Xu, F. Hou, Q. Yang and Q. Qi, Edge caching for layered video contents in mobile social networks, *IEEE Transactions on Multimedia*, **19** (2017), 2210–2221.

[35] Y. Wang, G. Yin, Z. Cai, Y. Dong and H. Dong, A trust-based probabilistic recommendation model for social networks, *Journal of Network and Computer Applications*, **55** (2015), 59–67.

[36] Y. Wang, N. Yuan Jing, D. Lian, L. Xu, X. Xie, E. Chen and Y. Rui, Regularity and conformity: Location prediction using heterogeneous mobility data, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2015), 1275–1284.

[37] J. Weston, C. Wang, R. Weiss and A. Berenzweig, Latent collaborative retrieval, *Proceedings of the 29th International Conference on Machine Learning*, 9–16.

[38] X. Xiao, Q. Zheng Yu andLuo and X. Xie, Finding similar users using category-based location history, *GIS*, (2010), 442–445.

[39] L. Xiong, X. Chen, T.-K. Huang, J. Schneider G and G. Carbonell, Temporal collaborative filtering with bayesian probabilistic tensor factorization, *Siam International Conference on Data Mining*, (2010), 211–222.

[40] D. Yang, D. Zhang, V. Zheng W and Z. Yu, Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 129–142.

[41] Q. Yang, A. Lim, X. Ruan and X. Qin, Location privacy protection in contention based forwarding for vanets, in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, IEEE, 2010, 1–5.

[42] Q. Yang, A. Lim, X. Ruan, X. Qin and D. Kim, Location-preserved contention-based routing in vehicular ad hoc networks, *Security and Communication Networks*, **9** (2016), 886–898.

[43] Q. Yang and H. Wang, Toward trustworthy vehicular social networks, *IEEE Communications Magazine*, **53** (2015), 42–47.

[44] Q. Yang, B. Zhu and S. Wu, An architecture of cloud-assisted information dissemination in vehicular networks, *IEEE Access*, **4** (2016), 2764–2770.

[45] M. Ye, P. Yin, W.-C. Lee and D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, *ACM SIGIR Conference on Research and Development in Information Retrieval*, (2011), 325–334.

[46] H. Yin, B. Cui, L. Chen, Z. Hu and X. Zhou, Dynamic user modeling in social media systems, *ACM Transactions on Information Systems (TOIS)*, **33** (2015), p10.

[47] H. Yin, B. Cui, Y. Sun, Z. Hu and L. Chen, Lcars: A spatial item recommender system, *ACM Transactions on Information Systems (TOIS)*, **32** (2014), p11.

[48] Y. Yu and X. Chen, A survey of point-of-interest recommendation in location-based social networks, *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[49] J. Yuan, Y. Zheng and X. Xie, Discovering regions of different functions in a city using human mobility and pois, *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2012), 186–194.

[50] Q. Yuan, G. Cong, Z. Ma, A. Sun and N. M. Thalmann, Time-aware point-of-interest recommendation, *ACM SIGIR*, (2013), 363–372.

[51] Q. Yuan, G. Cong and A. Sun, Graph-based point-of-interest recommendation with geographical and temporal influences, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, (2014), 659–668.

[52] F. Zhang, J. Yuan, D. Lian, X. Xie and W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, *KDD*, (2016), 353–362.

[53] J. Zhang and Y. Chow C.and Li, igeorec: A personalized and efficient geographical location recommendation framework, *IEEE Transaction on Service Computing*, *8* (2015), 701–714.

[54] J.-D. Zhang and C.-Y. Chow, Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations, *SIGIR*, (2015), 443–452.

[55] J.-D. Zhang and C.-Y. Chow, igslr: personalized geo-social location recommendation: A kernel density estimation approach, *n Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (2013), 334–343.

[56] J.-D. Zhang, C.-Y. Chow and Y. Zheng, Orec: An opinion-based point-of-interest recommendation framework, *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, (2015), 1641–1650.

[57] W. Zhang and J. Wang, Location and time aware social collaborative retrieval for new successive point-of-interest recommendation, *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, (2015), 1221–1230.

[58] Y. Zhang, Z. Zheng and M. Lyu, Wspred: A time-aware personalized qos prediction framework for web services, *Software Reliability Engineering (ISSRE)*, (2011), 210–219.

[59] S. Zhao, I. King and M. Lyu, Capturing geographical influence in poi recommendations, *International Conference on Neural Information Processing*, (2013), 530–537.

[60] S. Zhao, H. Zhao Tong anf Yang, M. Lyu R and I. King, Stellar: Spatial-temporal latent ranking for successive point-of-interest recommendation, *Thirtieth AAAI Conference on Artificial Intelligence*.

[61] V. Zheng M., Y. Zheng, X. Xie and Q. Yang, Collaborative location and activity recommendations with gps history data, *WWW*.

[62] X. Zheng, Z. Cai, J. Li and H. Gao, Location-privacy-aware review publication mechanism for local business service systems, in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, IEEE, 2017, 1–9.

[63] Y. Zheng, L. Zhang, X. Xie and W.-Y. Ma, Mining correlation between locations using human location history, *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (2009), 472–475.

Received September 2017; revised November 2017.

E-mail address: yi.xu001@montana.edu
E-mail address: qing.yang@unt.edu
E-mail address: chudh@hit.edu.cn