LBSN Collaborative Personalized Link Prediction Algorithm Postprint

Authors: Hu Min, Yongsheng Cui, Huang Hongcheng, Chen Yuanhui

Date: 2019-01-28T00:00:00+00:00

Abstract

In location-based social networks, there exists an intrinsic correlation between user links and location links, and different users exhibit distinct behaviors within the social network. Therefore, we propose a collaborative personalized link prediction algorithm to address these issues. For user-specific characteristics, kernel density estimation is employed to model users in temporal and spatial dimensions, overlapping community partitioning is performed based on interest groups, and personalized user link prediction is conducted through communities, friendships, and check-in relationships. Based on the personalized user link prediction results, random walks restarted from communities are utilized to predict personalized location links for users. The collaborative personalized link prediction algorithm iteratively enhances the performance of both user link prediction and location link prediction, and experimental results demonstrate that the proposed algorithm achieves superior prediction performance compared to existing algorithms.

Full Text

Cooperation-Based Personalized Link Prediction Algorithm in LBSN

Hu Min[1],[2], Cui Yongsheng[1], Huang Hongcheng[1]†, Chen Yuanhui[1] 1. School of Communication & Information Engineering, Chongqing University of Posts & Telecommunications, Chongqing 400065, China 2. Chongqing Engineering Research Center of Communication Software, Chongqing 400065, China

Abstract: In location-based social networks (LBSNs), user links and location links exhibit certain intrinsic correlations, and different users demonstrate distinct behavioral patterns in the network. To address these issues, this paper proposes a cooperation-based personalized link prediction algorithm. For capturing user personalized features, we employ kernel density estimation to model users in both temporal and spatial dimensions. Users are partitioned into overlapping communities based on interest groups, and personalized user link prediction is performed through community, friendship, and check-in relationships. Building upon the personalized user link prediction results, we utilize community-restarted random walks to predict personalized location links for users. The cooperation-based personalized link prediction algorithm iteratively enhances the performance of both user link prediction and location link prediction. Experimental results demonstrate that the proposed algorithm achieves superior prediction performance compared to existing algorithms.

Keywords: link prediction; location-based social network; kernel density estimation; personalization; random walk

0 Introduction

With the rapid development of social networks and the continuous proliferation of mobile smart devices, location-based social networks (LBSNs) have gradually become an ideal platform for people to maintain social relationships and share location information [1]. An increasing number of users are accustomed to using smart devices for location check-ins on social network platforms. However, as the user base grows, the volume of network data has exploded, leading to network information overload [2]. Research on link prediction in LBSNs can help users discover potential user link relationships from massive data and recommend other users or location information of interest, which holds significant research importance and application value for understanding the evolution patterns of LBSN structures and enhancing user loyalty to LBSN platforms [3].

Link prediction aims to discover and restore missing information in networks or predict potential relationships between nodes in the future based on network structure and existing information. Its research has important practical significance for applications such as friend recommendation and point-of-interest recommendation [4].

Currently, link prediction in LBSNs mainly falls into two categories: one predicts links between users, and the other predicts links between users and locations. For user-user link prediction, Valverde-Rebaza et al. [5] considered relationship strength between users and user location information, combining users' social patterns and mobility patterns to improve link prediction accuracy. Ding Yong et al. [6] proposed constructing a friend recommendation model from three attributes: interest, distance, and familiarity, additionally considering users' friend-making preference attributes. Bayrak et al. [7] argued that different categories of locations have varying degrees of influence on link establishment and proposed two new category-based features to enhance user link prediction performance. For user-location link prediction, Pavlos et al. [8] considered the social influence of user comments and the spatial influence of user check-ins, predicting user-location link relationships by incorporating these two features. Li Xin et al. [9] proposed a social relationship model based on interest circles in LBSNs, using social relationships including friend relationships and expert users as regularization terms in the matrix decomposition objective function to improve performance in predicting user-location links. Hosseini et al. [10] believed that a correspondence should exist between users and locations —if a user prefers activities on weekdays, they should be recommended locations popular on weekdays; similarly, for users who prefer weekend activities, weekend-popular locations should be recommended.

The aforementioned methods independently predict user-user link relationships and user-location link relationships for separate purposes, without considering the correlation between them. However, in reality, the two are not unrelated. For example, if a link exists between two users, they are likely to check in at the same locations. Conversely, if two users frequently check in at the same locations, they are likely to have a link relationship. Therefore, strong correlations exist between user links and location links, and they can mutually enhance each other' s prediction performance. Currently, few works jointly address these two problems. Zhang et al. [11] proposed a novel link prediction method called TRAIL, which obtains optimal user links and location links by maximizing the product of user link prediction and location link prediction. Reference [12] proposed an anchor link prediction method that incorporates users' social, spatial, and textual information.

In real life, geographic proximity significantly influences user check-in behavior. Although some of the aforementioned studies have considered the collaborative nature of user-location link prediction, they have not effectively integrated the impact of geographic location information on prediction performance. Currently, there are two approaches to considering location information: the first filters out locations far from users based on distance [13]; the second models user check-in data as a probability distribution function [14]. The second approach provides more rigorous consideration of geographic location information and thus yields better link prediction performance. However, different users have varying tolerances for spatial distances, and establishing a unified probability distribution model obscures users' personalized features, leading to loss of personalized information and affecting prediction accuracy. Additionally, users have different check-in habits and prefer going out at different times. Better matching user habits will further improve algorithm prediction performance. Therefore, this paper performs personalized modeling for each user to more accurately capture their personalized features in spatial location and behavioral habits.

First, we consider the collaborative problem of user link prediction and location link prediction. Second, we consider the personalized features of different users. This paper proposes a Cooperation-based Personalized Link Prediction (CPP) algorithm in LBSNs to improve link prediction performance from a novel perspective.

1 Problem Description

A location-based social network can be viewed as a heterogeneous network composed of different types of nodes and edges. This paper uses a triple to represent the node set, the edge set, and the node type set. We use several related definitions to better illustrate the problem.

We use to represent, where,

Definition 1 (Interest Group). If user and user both check in at locations of category , then users and are defined as belonging to the same interest group .

Definition 2 (Local Location Importance, LLI). Given user , let denote the set of locations visited by user , denote the total number of check-ins by user , and denote the number of check-ins by user at location . The local location importance represents the importance of location relative to all locations visited by user , with the formula . Let denote the set of all users.

Definition 3 (Global Location Importance, GLI). Let denote the total number of check-ins by all users at location , with other symbols defined as above. The global location importance represents the importance of user compared to other users for location , with the formula . Let denote the set of edges in the network, where edges described in this paper are directed edges. Let denote the set of edge weights in the network.

First, we partition user groups according to relevant definitions and construct the initial prediction space . Then, through the link prediction algorithm proposed in this paper, we predict potential user links and location links in the network. The related problem definitions are expressed as:

1.1 Problem Input

Based on the above definitions, the inputs for this research are: a) Weighted heterogeneous network b) Initial prediction space .

1.2 Problem Output

Given the weighted heterogeneous network in the location-based social network and the initial prediction space , we address the following problems: a) How to perform personalized user modeling? We use kernel density estimation from nonparametric estimation methods to model each user’ s check-in time and spatial location tolerance. The advantage is that it does not require prior assumptions about sample distribution characteristics, making it suitable for small-sample datasets. Temporal modeling uses one-dimensional kernel density estimation for time scalars, while spatial modeling uses two-dimensional kernel density estimation for latitude-longitude coordinate vectors. b) How to design a personalized user link prediction algorithm that simultaneously solves user link prediction and location link prediction problems in LBSNs? Based on users’ historical check-in locations, we obtain user interest groups and partition users into communities accordingly. We calculate the probability of a link between two users using a user link prediction method and update the user link probability using personalized temporal features. We identify important locations in each community through community-restarted random walks and update each user’ s location link probability using personalized spatial features. Finally, through iterative refinement of user links and location links, the performance of both is mutually enhanced to obtain the final prediction results.

---

2 User Personalized Modeling

Most existing research uses global user data for modeling, rarely considering users’ personalized behavioral habits and individual preferences, resulting in loss of personalized information. This paper performs personalized modeling for individual users from both temporal and spatial perspectives based on user check-in data to capture personalized features and further improve the accuracy of user link prediction and location link prediction. We adopt the kernel density estimation method for personalized user modeling.

2.1 Kernel Density Estimation

Kernel density estimation is a non-parametric estimation method whose advantage is that it does not require prior assumptions about sample distribution and can discover distribution characteristics from the samples themselves. Compared to parametric estimation methods, it avoids complex distribution assumptions and parameter regression processes, making sample distribution estimation simple and efficient. Therefore, kernel density estimation is well-suited for personalized modeling of individual users. Based on the dimensionality of sample object data, kernel density estimation can be divided into one-dimensional and multi-dimensional kernel density estimation.

2.1.1 One-dimensional Kernel Density Estimation Let be a set of independent and identically distributed random variables drawn from a sample, and denote its unknown probability density function. Its kernel density estimation formula is:

where: is the window width, is the kernel function, and the kernel function must satisfy and .

2.1.2 Multi-dimensional Kernel Density Estimation When sample objects transform from scalar form to -dimensional vectors, the sample’ s density distribution function becomes multi-dimensional kernel density estimation. Assume there are -dimensional random variables that follow an independent and identically distributed probability density function . Its multi-dimensional kernel density estimation is:

where: the multi-dimensional kernel density estimation' s kernel function is constructed from the product of one-dimensional kernel functions. When , it becomes the two-dimensional kernel density estimation function.

2.2 User Personalized Time Modeling

In real life, different users have different lifestyle habits. Some users prefer to go out and check in during the day, while others prefer nighttime activities. These check-in behaviors often reflect their personal preferences and lifestyle habits. Therefore, the more similar the check-in behavior distributions of two users, the more likely they share the same personal hobbies and behavioral habits. According to homophily theory [15], they are more likely to become friends.

First, users may check in at any time during the 24 hours of a day. To avoid severe bias in the probability function, this paper divides user check-in time into 24 equal parts, corresponding to 24 time slots in a day, and then counts user check-in frequencies in these 24 time slots. Specifically, this paper counts the check-in time frequencies of user and user , with results shown in [Figure 1: see original paper]. The histograms represent the check-in time frequency distributions of users and respectively. The figure shows that users and have relatively obvious differences in check-in habits, suggesting low similarity in their check-in behavioral patterns.

Although the above method can judge similarity between users, it is difficult to achieve ideal results because dividing a day into 24 time slots leads to identical check-in frequencies for different time points within a slot, which is illogical. Therefore, we can use a Gaussian kernel function to establish a kernel density estimation distribution based on check-in time, as shown by the curves in [Figure 1: see original paper]. The advantages of using continuous distribution are: a) it can accurately reflect users' check-in behavior distributions across continuous time in a day; b) for users with sparse check-ins who have no check-in records in most time slots, using discrete statistics would cause severe bias in calculating similarity between users, which can be effectively alleviated through continuous kernel density estimation distributions.

Assume the check-in probability of user in time period is . Under one-dimensional kernel density estimation, the similarity between two users is obtained through the cosine similarity function as follows:

where: is the set of check-in times.

2.3 User Personalized Space Modeling

This paper adopts two-dimensional kernel density estimation to mine the probability of a single user checking in at a new location. Let denote the set of locations visited by user . Using two-dimensional kernel density estimation, we obtain the probability of user visiting a new location :

where: represents the two-dimensional spatial vector coordinates of location , with representing longitude and representing latitude. denotes the kernel function, and represents the smoothing window, also called the window width.

In equation (9), the kernel function selected is the standard Gaussian kernel function, expressed as:

The optimal window width is set as , where and represent the mean and variance of longitude and latitude values in the set respectively, calculated as:

---

3 Community Division and Link Prediction

3.1 Community Division

Traditional community division in social networks adopts network structure-based partitioning methods. To better mine users with similar interests, this paper utilizes the above interest group definition to partition communities. As shown in [Figure 2: see original paper], since a user may visit multiple locations and a single location may belong to multiple categories, the communities partitioned based on interest groups form overlapping communities—meaning one user may belong to multiple communities. Based on interest group partitioning, we construct a user-community matrix of size , denoted as , where indicates user does not belong to community , and conversely, indicates user belongs to community .

3.2 User Link Prediction

If users and belong to the same community, it indicates they share similar interests. The more communities they have in common, the greater their similarity and the higher the likelihood of forming a link. Following the method in reference [16], we calculate the link probability between two users , in community as:

If one of users , does not belong to community , then . Since a user may belong to multiple communities, the probability that , have no link can be expressed as:

where: represents the probability that , have a link.

The number of co-checked-in locations and the number of common friends between users also affect link existence probability. Therefore, equation (15) can be modified as:

Since there are two types of edges, and this paper assumes that transition probabilities between the same type of edges are identical, the transition probability between two user nodes can be set as follows:

where: represents the user-user relationship matrix, indicating whether users and have a friendship; represents the user-location relationship matrix, indicating whether user has checked in at location . Through this approach, we obtain the probability of each user forming links with other users in the network. On this basis, considering the similarity of check-in time distributions between users , we match users' personalized behavioral habits and update the link existence probability between each user and other non-friend users as follows:

where: represents the probability of user ' s existence, and equals when user exists; otherwise, equals .

Since different locations have varying importance to users, we need to quantify the importance of location to user . This paper primarily uses the defined local location importance and global location importance to quantify this metric, i.e.,

3.3 Location Link Prediction

To reasonably predict the link relationship between users and locations, this paper first makes the following assumption: given a target user , if user belongs to community , then user will be more willing to visit locations important to community . If user belongs to multiple communities, the locations user is more willing to visit are determined comprehensively by multiple communities.

To find important locations in community , this paper adopts a random walk with restart from community , expressed as follows:

where: represents the arrival probability of user nodes, represents the arrival probability of location nodes, denotes the iteration count, represents the neighbor location set of user , represents the neighbor user set of user , and represents the neighbor user set of location . represents the transition probability from user to location , represents the transition probability from location to user , and is the restart probability of the random walk. If belongs to community , ; otherwise, .

Since location nodes are only connected to users, the transition probability between location nodes and user nodes can be expressed as:

User nodes can be connected to either location nodes or user nodes. To coordinate the weight relationships between user nodes and location/user nodes, we introduce a 调节参数 (adjustment parameter) . The transition probability from user to location can then be expressed as:

When the community-restarted random walk converges, the arrival probability of each location node under community can be understood as the importance of each location in community , as follows:

Since there are communities in the network, we can obtain the community-location matrix as:

Using the product of the community-location matrix and the user-community relationship to represent the probability of user visiting location driven by community interests, the calculation formula is:

where: represents all communities in the network, and represents the importance of location in community .

The important locations selected through community relationships are summarized from sample user sets with similar interests to the target user, obtained from the interest perspective. However, due to geographic location influence, users may not necessarily visit these locations. For example, user is a homebody who rarely visits distant locations, while user enjoys traveling and frequently tours different countries. If both users and are found to be interested in location from the interest perspective, and location is far from both users and , considering that user ' s tolerance for distance may be lower than user ' s, we can consider that user ' s probability of visiting location will be less than user ' s. Therefore, predicting whether a user will visit a certain location requires considering both the interest perspective and each user' s spatial location tolerance.

Through personalized spatial modeling of users, we obtain the probability of user visiting location under their own spatial tolerance . We then update the access probability matrix obtained from the interest perspective using the following formula:

where: represents the probability that user may visit . We update the matrix based on as follows:

where: represents the probability of user visiting location after iterations. After updating the matrix, we can begin a new round of user link prediction and location link prediction. The algorithm description is presented as Algorithm 1.

The complexity of the user link prediction algorithm in this paper is , where is the number of co-neighbor users, is the number of communities, is the number of co-neighbor locations, and is the number of users. The complexity of the location link prediction algorithm is , where is the number of random walk convergence iterations. Assuming the mutual iteration count is , the time complexity of the CPP algorithm is .

Algorithm 1: CPP Algorithm Input: Location-based social network ; initial prediction space . **Output:** Potential user links and location links in the network.

1. Based on individual user check-in times, model user check-in behavior probability distribution using one-dimensional kernel density estimation to obtain similarity between users .
2. Based on latitude and longitude information of users' historically visited locations, model users'spatial tolerance information using two-dimensional kernel density estimation to obtain the probability of each user visiting new locations .

3. Partition users into overlapping communities based on interest group definitions and construct user-community matrix .
4. repeat
5. // User link prediction
6. Calculate edge weights in the network using equations (17)~(20).
7. Update user link probability using equation (14) to obtain personalized user link probability .
8. // Location link prediction
9. Calculate arrival probabilities of each location in each community using equations (15) and (16).
10. Obtain community-location matrix .
11. Calculate user-location probability matrix under community drive using equation (24).
12. Update user-location matrix using equation (25).
13. until reaching the specified number of iterations

4 Experiments

4.1 Dataset Description

This paper conducts experiments using a dataset crawled from Gowalla, sourced from reference [17], containing user tables, location tables, friend relationship tables, and check-in tables. The user table includes information such as the number of user check-ins and the number of location categories checked in; the location table includes location latitude and longitude information, city, and category; the friend relationship table includes friendship edges between users; the check-in table includes user check-in locations and corresponding check-in times, accurate to the hour.

In the experiments, this paper selects check-in records from Berlin and Houston from the raw data as experimental datasets. We remove users with total check-ins fewer than 10 or check-in locations no more than 2, and delete locations with fewer than 5 check-ins or checked in by no more than 2 users to reduce the impact of data sparsity on experiments. Table 1 provides detailed descriptions of the processed datasets.

4.2 Experimental Results Analysis

In the experiments, we first set the maximum iteration count for user link prediction and location link prediction to 30. The restart probability for the random walk with restart is . The comparison of parameter variations is shown in [Figure 3: see original paper]. Different adjustment parameters affect location link prediction accuracy, with location link prediction accuracy being optimal at around . The user link prediction parameter adjusts the weights of users' common communities, common friends, and common check-in locations on user link prediction. As shown in Table 2, we verify the impact of parameter on user link prediction in real datasets. When is relatively large, link prediction accuracy improves, with optimal user link prediction accuracy at around . Therefore, we set for the Berlin dataset and for the Houston dataset. The simulations primarily consider two factors affecting experimental results: first, the impact of different proportions of training samples on different methods; second, the impact of iteration count on the proposed CPP algorithm.

We select typical link prediction methods for comparison with the CPP algorithm. The comparison algorithms include: Common Neighbors of Places (CNP) [18], a user link prediction algorithm that assumes users with more common friends who have visited places checked in by one user have higher probability of forming links; Friend++ [19], an improved random walk with restart algorithm for user link prediction that integrates a weighted-average method into the random walk framework; Rank-GeoFM [20], a location link prediction algorithm that optimizes location ranking functions by considering user preferences, check-in locations, and spatiotemporal context; and TRAIL [11], an algorithm that simultaneously predicts user links and location links by maximizing the product of user link probability functions and location link probability functions.

First, we compare the impact of different proportions of training samples on algorithm performance. This paper divides the dataset using 9 standards. To ensure

The user link prediction results are shown in Figures 4 and 5. Figure 4 shows user link prediction accuracy, where the horizontal axis represents the training set proportion and the vertical axis represents algorithm accuracy. Figure 5 shows user link prediction AUC values, where the horizontal axis represents the training set proportion and the vertical axis represents AUC values.

[Figure 4: see original paper]

[Figure 5: see original paper]

The direct physical meaning of AUC is the area under the ROC curve. The ROC curve is a commonly used performance evaluation metric for classifiers. Since a binary classifier’ s output of 1 or 0 often depends on the output probability and a preset probability threshold, and the selection of the probability threshold affects classifier performance to some extent, the ROC curve is adopted as a metric that remains correct regardless of threshold selection. However, the ROC curve only reflects the classifier’ s capability, while the AUC value can quantify the ROC curve to visually present the classifier’ s ability. Larger AUC values indicate better classification performance, with an ideal AUC value of 1.

From Figures 4 and 5, we can see that as the training set proportion increases, the accuracy and AUC values of all methods show an upward trend. The CPP algorithm consistently outperforms the TRAIL algorithm because TRAIL fails to effectively capture users’ personalized features, neglecting consideration of similarity in behavioral habits between users and users’ spatial location tolerance. In contrast, CPP fully integrates users' personalized features in both temporal and spatial dimensions on top of algorithmic iteration, resulting in improved prediction performance. The CPP and TRAIL algorithms significantly outperform CNP and Friend++ because Friend++ only utilizes users' social relationships without considering check-in relationships, resulting in incomplete information utilization. Meanwhile, CNP and Friend++ fail to incorporate iterative thinking, leading to less satisfactory algorithm performance.

The location link prediction results are shown in Table 3. From the table, we can see that all algorithms have lower accuracy and AUC values in the Houston dataset because the Houston dataset is larger with stronger data sparsity in the network, leading to decreased algorithm prediction performance. Additionally, the CPP algorithm consistently achieves the best prediction performance because it integrates community knowledge and personalized selection, enabling it to reasonably mine user interests and accurately predict potential location links in the network.

Finally, to verify that algorithm iteration can effectively improve link prediction performance, we set the training set proportion to 0.9 and iteration counts to . We repeat each iteration count 10 times and average the results as the final prediction. The experimental results are shown in [Figure 6: see original paper], clearly demonstrating that as iteration count increases, the AUC values for both user link prediction and location link prediction continuously improve. When the iteration count reaches around 30, the AUC values stabilize. The experiments prove that the iterative process of the CPP algorithm can effectively enhance link prediction performance.

[Figure 6: see original paper]

---

5 Conclusion

This paper addresses the problems of traditional link prediction methods that solve user links and location links independently while failing to capture users' personalization in temporal and spatial dimensions. We propose a Cooperation-based Personalized Link Prediction (CPP) algorithm in LBSNs. Through one-dimensional and two-dimensional kernel density estimation, we model each user' s check-in time and check-in space to mine users' check-in behavioral habits and spatial location tolerance. Simultaneously, based on overlapping community-restarted random walks, we effectively integrate users' personalized features into the model. Through a limited iterative process, the performance of both predictions continuously improves and simultaneously reaches optimality. Our method simultaneously completes user link prediction and location link prediction tasks from a novel perspective, effectively mining and utilizing the correlation between the two tasks. The limitation lies in the relatively high time complexity of the algorithm. Future work will focus on algorithm optimization to address the high time complexity issue in CPP.

References

[1] Stepan T, Morawski J M, Dick S, et al. Incorporating spatial, temporal, and social context in recommendations for location-based social networks [J]. IEEE Trans on Computational Social Systems, 2016, 3 (4): 1-24. [2] 文馨, 陈能成, 肖长江. 基于 Spark GraphX 和社交网络大数据的用户影响力分析 [J]. 计算机应用研究, 2018, 35 (3): 830-834. (Wen Xin, Chen Nengcheng, Xiao Changjiang. Analysis of user influence based on social network big data and Spark GraphX [J]. Application Research of Computers, 2018, 35 (03): 830-834.) [3] Zhou Ningnan, Zhao Waynexin, Zhang Xiao, et al. A general multi-context embedding model for mining human trajectory data [J]. IEEE Trans on Knowledge and Data Engineering, 2016, 28 (8): 189-202. [4] Xiao Yunpeng, Li Xixi, Wang Haoxun, et al. 3-HBP: a three-level hidden bayesian link prediction model in social networks [J]. IEEE Trans on Computational Social Systems, 2018, 5 (2): 430-443. [5] Valverda-rebaza J, Roche M, Poncelet P, et al. The role of location and social strength for friendship prediction in location-based social networks [J]. Information Processing & Management, 2018, 54 (4): 475-489. [6] 丁勇, 刘菁, 蒋翠清, 等. LBSN 中考虑用户交友偏好的好友推荐方法研究 [J]. 系统工程理论与实践, 2017, 37 (11): 2975-2982. (Ding Yong, Liu Jing, Jiang Cuiqing, et al. A study of friends recommendation algorithm considering users’ preference of making friends in the LBSN [J]. Systems Engineering Theory & Practice, 2017, 37 (11): 2975-2982.) [7] Bayrak A E, Polat F. Examining place categories for link prediction in location based social networks [C]// Proc of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. New York: ACM Press, 2016: 976-979. [8] Pavlos K, Yannis M. A time-aware Spatio-textual Recommender system [J]. Expert Systems with Applications, 2017, 78: 396-406. [9] 李鑫, 刘贵全, 李琳, 等. LBSN 上基于兴趣圈中社会关系挖掘的推荐算法 [J]. 计算机研究与发展, 2017, 54 (2): 394-404. (Li Xin, Liu Guiquan, Li Lin, et al. Circle-based and social connection embedded recommendation in LBSN [J]. Journal of Computer Research and Development, 2017, 54 (2): 394-404.) [10] Hosseini S, Li L T. Point-of-interest recommendation using temporal orientations of users and locations [C]//Proc of International Conference on Database Systems for Advanced Applications. Dallas: Springer International Publishing, 2016: 330-347. [11] Zhang Jiawei, Kong Xiangnan, Yu P S. Transferring Heterogeneous Links Across Location-based Social Networks [C]// Proc of the 7th ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2014: 303-312. [12] Zhan Qianyi, Zhang Jiawei, Yu P S. Integrated anchor and social link predictions across multiple social networks [J]. Knowledge and Information Systems, 2018, 6: 1-24. [13] Wang Hao, Terrovitis M, Mamoulis N. Location recommendation in Location-based Social Networks using user Check-in Data [C]// Proc of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM Press, 2013: 334-343. [14] Yao Lina, Sheng Quan, Wang Xianzhi, et al. Collaborative location recommendation integrating multi-dimensional contextual information [J]. ACM Trans on Internet

Technology, 2018, 18 (3): 1-24. [15] Timothy L F, Neville J. Randomization tests for distinguishing social influence and homophily effects [C]// Proc of the 19th International Conference on World Wide Web. New York: ACM Press, 2010: 601-610. [16] Yang J, Mcauley J, Leskovec J. Community Detection in Networks with Node Attributes [C]//Proc of the 13th IEEE International Conference on Data Mining. Piscataway,NJ: IEEE Press, 2013: 1151-1156. [17] Liu Yong, Wei Wei, Sun Aixin, et al. Exploiting geographical neighborhood characteristics for location recommendation [C]// Proc of the 23rd ACM International Conference on Conference on Information and Knowledge Management. New York: ACM Press, 2014: 739-748. [18] Alverda-rebaza J, Roche M, Poncelet P, et al. Exploiting social and mobility patterns for friendship prediction in location-based social networks [C]// Proc of the 23rd International Conference on Pattern Recognition. Piscataway,Nj: IEEE, 2016: 2526-2531. [19] Gong Jibing, Gao Xiaoxia, Cheng Hong, et al. Integrating a weighted-average method into the random walk framework to generate individual friend recommendations [J]. Science China Information Sciences, 2017, 60 (11): 110104. [20] Li Xutao, Gao Cong, Li Xiaoli, et al. Rank-geofm: a ranking based geographical factorization method for point of interest recommendation [C]// Proc of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2015: 433-442.

Note: Figure translations are in progress. See original paper for figures.