Neurocomputing 242 (2017) 195–205

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS

Yuankai Ying, Ling Chen*, Gencai Chen

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 9 May 2016
Revised 6 February 2017
Accepted 24 February 2017
Available online 6 March 2017

Communicated by Peng Cui

Keywords:
LBSN
POI recommendation
Tensor decomposition
HITS
User preferences

ABSTRACT

The popularity of location-based social networks (LBSN) provides us with a new perspective for understanding people's travel behaviours and enables a lot of location-based services, such as point of interest (POI) recommendation. However, personalized POI recommendation is very challenging, as the user-location matrix is very sparse for traditional collaborative filtering (CF)-based POI recommendation approaches. The problem becomes even more challenging when people travel to a new city. In addition, temporal influence plays an important role in POI recommendation, for most users tend to visit different kinds of POIs at different time in a day, e.g., visiting a food-related POI at noon and visiting a nightlife spot at night. To the end, we propose a novel POI recommendation system, which consists of two components: context-aware tensor decomposition (CTD) for user preferences modelling and weighted HITS (Hypertext Induced Topic Search)-based POI rating (WHBPR). We model user preferences with a three-dimension tensor (user-category-time). Supplementing the missing entries of the tensor through CTD with the aid of other three matrices, we recover user preferences of different time slots. WHBPR incorporates the impacts of user preferences and social opinions on POI rating. We evaluated our method using the real Foursquare datasets, verifying the advantages of our method beyond other baselines.

1. Introduction

The increasing adoption of GPS-equipped smart devices fosters a bunch of location-based social networking services (LBSNs) [1], e.g., Foursquare, Facebook Places, Google Places, etc., where users can easily check in at point of interests (POIs), e.g., stores, restaurants, and share their life experiences in the physical world via smart devices. It is crucial to utilize user location history to make personalized recommendation, which helps users know new POIs and explore new regions, facilitates advertisers to launch mobile advertisements to targeted users.

Under such a circumstance, POI recommendations provide a user with some POIs (e.g., a Chinese restaurant) that match his personal preferences. Nevertheless, a high-quality POI recommendation has to simultaneously consider the following three factors. (1) User preferences: Music lovers may be more interested in concerts while shopaholics would pay more attention to shopping malls. (2) Temporal influence: user preferences change over different time slots of day. For instance, Fig. 1 plots an illustrative example of a user's location records on his top 5 most visited POI categories over 24 h. In the figure, we can observe that this user often went to a cocktail bar at 3 o' clock and had a lunch in a bakery at 13 o' clock. This example suggests a strong correlation between a user's check-in time and the corresponding check-in preference. (3) Social opinions: people's preferences or behavioural patterns may change when they travel in different cities, especially in cities that are new to them. For example, a user living in a small city likes food very much, but does not like shopping. When he travels in Hong Kong, especially for the first time, the user is very likely to visit local shopping centres. Thus, the location histories left by other people in the querying city are valuable resources for making a recommendation.

However, inferring user preferences for POIs is very challenging by exploring users' location history. Firstly, an individual can only visit a limited number of locations, while the number of POIs in a city is very large. This results in a very sparse user-location matrix for most existing collaborative filtering-based POI recommendation systems. Secondly, the observation of travel locality makes the task more challenging if a user travels to a new place. The observation of travel locality shows that users tend to travel a limited distance when visiting locations in practice. For example, Fig. 2 presents the location history distribution of a local resident in New York City (NYC). Obviously, the location records generated by the user are locally crowed, for 56.8% of the location history were generated in NYC but 74.7% of those records concentrate upon an area

* Corresponding author.
E-mail addresses: yingyuankai@zju.edu.cn (Y. Ying), lingchen@zju.edu.cn (L. Chen), chengc@zju.edu.cn (G. Chen).

http://dx.doi.org/10.1016/j.neucom.2017.02.067
0925-2312/

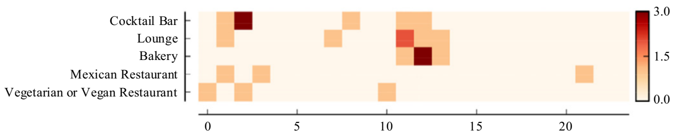

Fig. 1. User preference changes over time of day.

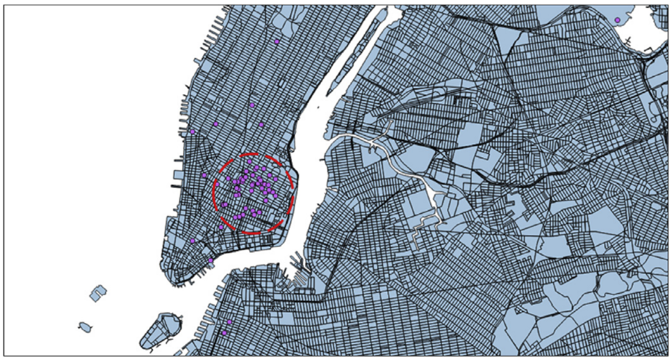

Fig. 2. A user's location history distribution in New York City.

of only about 4 km^2 (marked by the red dashed circle in Fig. 2). This phenomenon is quite common in the real world, aggravating the data sparsity problem with personalized POI recommendations. Although some CF-based methods can achieve better recommendation performance by exploiting social or geographical influences, these methods are not feasible any more, especially when dealing with the new city problem, because a querying user usually just has a few or even no overlap in the querying city that is new to him.

To the end, we propose a temporal-aware POI recommendation system with the consideration of the above mentioned three factors, consisting of two primary components: Firstly, we model user preferences with a three-dimension tensor, where the three dimensions stand for users, POI categories, and time slots, respectively. Supplementing the missing entries of the tensor through context-aware tensor decomposition (CTD) with the aid of other three matrices, we recover user preferences of different time slots. by modelling a user's preferences based on the category information of his location history (instead of physical locations), our recommender system not only can cope with the data sparsity problem to some extent but also can facilitate people's travel to a new city. Karatzoglou et al. [17] and Yao et al. [18] also proposed context-aware tensor (a user-location-time tensor) decomposition-based recommendation methods. However, they cannot deal with the data sparsity problem very well because of the large scale of location dimension. Secondly, we stand on the effective popularity ranking function of HITS algorithm and propose a weighed HITS-based POI rating approach (WHBPR) to infer the score of a POI by assigning the querying user's preferences values to corresponding links in LBSN graph (*Definition 5*), which incorporates the impacts of user preferences and POI popularity in the querying city.

In summary, our work has following three primary contributions:

- Propose a system that recommends POIs by assembling CTD and WHBPR, considering user preferences, temporal influence, and social opinions.
- Propose a user preferences modelling method, which learns temporal-aware user preferences from user's location history and complement user preferences through CTD with the aid of other three matrices, alleviating the data sparseness problem and facilitating people's travel to a new city by using the POI categories.
- Propose a POI scoring method, which infers the score of a POI with the weighted HITS-based POI scoring method, considering user preferences and social opinions in the querying city.

The rest of this paper is organized as follows: Section 2 summarizes the related work. Section 3 overviews the framework of our system. Sections 4 and 5 introduce two major parts of our method: (1) CTD for user preferences modelling; and (2) weighted HITS-based POI rating. Extensive experimental results based on real datasets are presented in Section 6 with some discussions, followed by the conclusions and future work in the last section.

2. Related work

The POI recommendation based on location history has attracted a number of researchers in recent years. In this section, we categorize the major methodologies used by recommendation systems in LBSNs as being based on: (1) collaborative filtering or (2) link analysis.

2.1. Collaborative filtering-based POI recommendations

CF is widely used in traditional recommendation systems [19]. CF-based systems can be divided into two categories, i.e., memory-based CF systems and model-based CF systems. The memory-based systems also can be divided into two subgroups: (1) user-based systems, such as [3], that use the similarity between each pair of users; and (2) item-based systems, such as [2,4,20], that use the similarity between each pair of items. For example, Zhang et al. [5] introduce a user-based CF method that considers both the temporal influence and the spatial influence to make the time-aware POI recommendation. However, it should be pointed out that solely using a CF method, either the user-based or the item-based CF method, cannot handle the data sparsity problem very well.

In contrast, model-based CF builds recommendation models using data mining techniques, such as matrix factorization (MF) [6,7,21] and probabilistic topic model [22]. Cheng et al. [6] proposed a method that fuses MF with geographical and social influence for POI recommendation in LBSNs. Gao et al. [8] proposed a method that utilizes the social network information for solving the cold start location prediction problem. Gao et al. [10] also proposed a location recommendation framework, based on the temporal properties of user movement observed from a real-world LBSN dataset. Ye et al. [9] proposed a unified POI recommendation framework, which fuses user preferences to a POI with social influence and geographical influence. In a very recent work, Yao et al. [18] proposed a CF method based on nonnegative tensor factorization that exploits a high-order tensor instead of the traditional user-location matrix to model multi-dimensional contextual information. Although above-mentioned methods achieve relatively good recommendation performance by reducing dimensions and exploiting social, geographical, and temporal influences, these methods still cannot deal with the serious data sparsity issue when people travel to a new city, for there are a few or even no overlap between users' location histories in the querying city. Instead of using pure CF-based methods, Bao et al. [23] proposed a hybrid method to alleviate the sparsity problem arisen from the travel locality by projecting user's location history into a well-designed weighted category hierarchy (WCH). After finding a set of local users as experts and then constructing a user-item matrix, it employed a traditional user-based CF model over this matrix to infer the rating of a candidate item. However, this method would have a poor performance when querying users just have a few location records.

Our recommendation system also is a hybrid method but differs from the above-mentioned works in the aspect of user preferences modelling. We project users' location histories into the category space time-slot by time-slot and then complement it by using CTD, which has the aid of other three matrices alleviating the data sparseness problem for user preferences modelling. Due to user preferences complementing, our method could also achieve a good performance when a user is lack of location history. Furthermore, we exploit the temporal influence to model the temporal-aware preferences for all users. In addition, there are many other tensor decomposition methods in different fields. For example, Jiang et al. [33] presented a novel tensor factorization based Flexible Evolutionary Multi-faceted Analysis framework (FEMA) for temporal multi-faceted behaviour prediction and behavioural pattern mining. Rendle et al. [34] proposed a Pairwise Interaction Tensor Factorization model (PITF) for personalized tag recommendation which models the pairwise interactions between users, items and tags with linear runtime. Karatzoglou et al. [35] introduced a CF method based on Tensor decomposition, allowing for a flexible and generic integration of contextual information by modelling the data as a user-item-context N-dimensional tensor. However, these existing tensor decomposition methods cannot alleviate the data sparsity issue very well, while our CTD supplements the missing entries with the aid of additional information, e.g., user features, temporal features, and the correlation between different categories.

2.2. Link analysis-based POI recommendations

Link analysis algorithms (e.g., PageRank [24] and HITS [12,13]) are widely used to rank the web pages. These algorithms extract high quality nodes from a complex graph by analysing the structure. In LBSNs, there are graphs of different types, e.g., user-user, user-location, location-location, and the combinations of above two or three kind of graphs. For example, Zheng et al. [25] proposed a method of mining interesting locations and travel sequences from GPS trajectories. They first modelled multiple individuals' location histories with a tree-based hierarchical graph (TBHG), and then propose a HITS-based inference model based TBHG, which regards an individual's access on a location as a directed link from the user to that location. Although above method can make recommendations by taking users' experiences into account and be robust against the cold start problem, these methods only provide generic recommendations for all users without considering user's personal preferences.

However, there are some personalized link analysis-based location recommendation methods only based on the LBSN graph structure, which are highly dependent on the scale of a querying user's location history. For instance, Long et al. [14] proposed a HITS-based POI recommendation method that can incorporate the impact of the social relationships on recommendations. Noulas et al. [15] proposed a new model based on personalized random walks over a user-location graph by seamlessly combining social network and location visiting frequency data. Yuan et al. [16] proposed a graph-based POI recommendation method with geographical and temporal influences by using the geographical-temporal influences aware graph (GTAG). Raymond et al. [26] proposed a random walk-based link analysis algorithm for location recommendation using the link propagation principle to exploit the spatiotemporal correlations among the locations. Our recommendation system mainly differs from above-mentioned works in following one aspect: we project the querying user's preferences into user-location links as the weights. Thus, our method can provide personalized recommendations for a particular user based on not only the graph structure but also user preferences modelled by user's location history.

In addition, there are many other link analysis-based methods. For example, Zhang et al. [36] proposed a learning-based random walk model over a heterogeneous network for the recommendation context. Jiang et al. [37] proposed a novel Hybrid Random Walk method (HRW) to select transferable items in auxiliary domains, bridge cross-domain knowledge with the social domain, and accurately predict user-item links in a target domain. However, the first method mainly focuses on non-personalized recommendation and it is difficult to obtain such rich cross-domain data for social recommendation with regard to the second method HRW. In contrast, based on the LBSN graph (*Definition 5*) built simply from the location histories of users and friendships among users in the querying city, our method provides personalized POI recommendation.

3. Overview

In this section, we first give the definitions of some basic concepts and terms, and then overview the framework of our system.

3.1. Preliminary

Definition 1 (POI). A point of interest (POI) is a specific location that someone may find useful or interesting in the physical world, having a ID, name, address, Longitude & Latitude, categories, etc. For example, a Chinese restaurant or a cocktail bar.

Definition 2 (POI categories). The POI categories indicating the functionalities of POIs have different granularities, which are usually represented by a category hierarchy. Foursquare also maintains a category system (i.e., a category hierarchy), which has eight predefined POI categories as the 1st-level categories, namely, "Arts & Entertainment", "Colleges & Universities", "Food", "Great Outdoors", "Nightlife Spots", "Travel Spots", "Shops", and "Home, Work and Others". In this category hierarchy, the category numbers of each level are 8, 241 and 103, respectively, and not all the 2nd-level categories have their sub-categories. Therefore, we focus on the second level categories as a vector to model user preferences in this paper, striking a balance between the granularity of user preferences modelling and the size of vector.

Definition 3 (Check-in). Many LBSNs, such as Foursquare, Google+ and Weibo, allow users to check in at a physical place, and share their experiences with their friends and leave tips for POIs. For example, Fig. 3 presents a check-in example created via Swarm for IOS (a check-in app by Foursquare), which has a user name, a POI name, an address, a created time, a geolocation, several POI categories, and usually associated with a short text or some photos.

Definition 4 (tips, to-do and done). Tips [27] are pieces of information containing recommendations or reviews, either positive or negative, about the POIs. For example, the best option of a menu in a restaurant, a great place to have lunch in an airport, or even a complaint about the service in a location. After reading a tip, a user may add it to his to-do list or even mark it as done. For instance, user A posts a tip at certain POI. User B searches for nearby locations, finds the POI tipped by previously by user A, reads it, and decides to add it to his to-do list. User B may also mark the tip as done, indicating that he followed user A's recommendation. Thus, the total number of times a tip was marked as done or added to users' to-do list can be seen as an estimate of the feedback from other users regarding that tip.

Definition 5 (LBSN graph). As shown in Fig. 4, we present the LBSN with a graph, where both users and POIs are represented by vertices, friendships among users and check-ins at POIs generated by users are marked by undirected and directed solid lines, respectively. A user can check-in at multiple POIs. Meanwhile, a POI may belong to several categories (shown as the directed dashed lines).

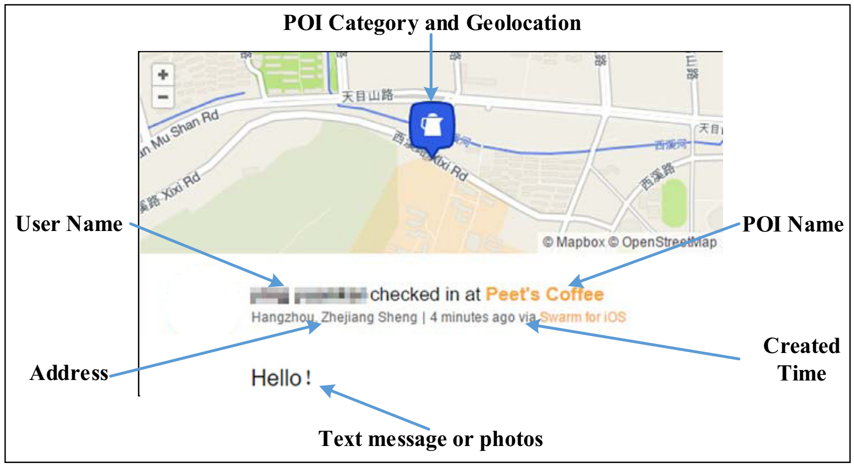

Fig. 3. An example of check-in.

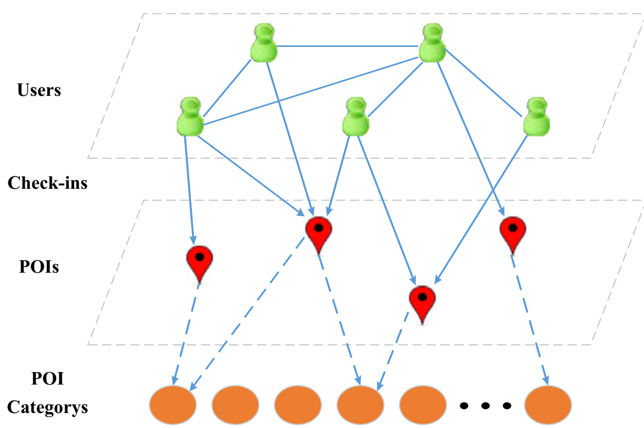

Fig. 4. The LBSN graph.

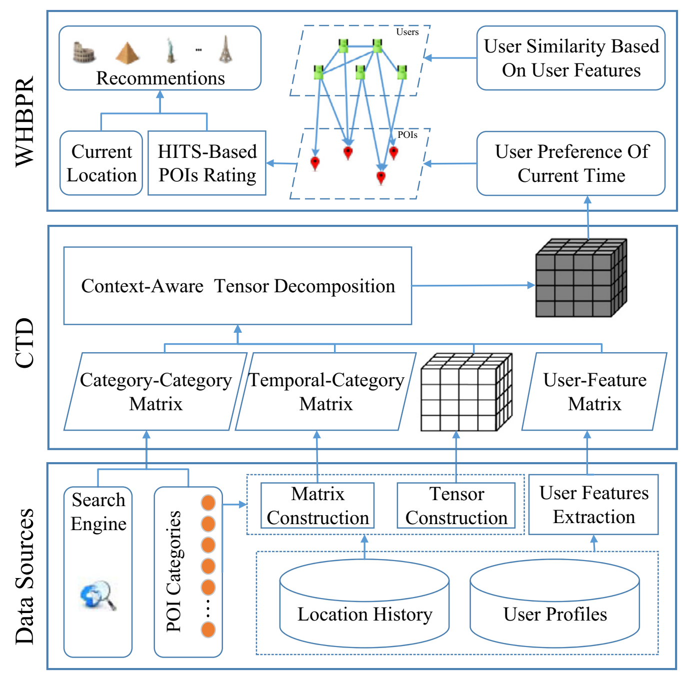

Fig. 5. Framework of our system.

3.2. System framework

Fig. 5 presents the framework of our personalized POIs recommendation system, consisting of two components: (1) context-aware tensor decomposition (CTD) for user preferences modelling and (2) weighted HITS-based POI rating (WHBPR). WHBPR incorporates the impacts of user preferences, which are modelled from user's location history through CTD, and social opinions on POI rating.

3.2.1. CTD for user preferences modelling

This part is comprised of two major procedures: 1) relevant data structures construction and 2) context-aware tensor completion. The first procedure constructs a tensor and three matrices, which would be used in context-aware tensor decomposition process later on. Given the pre-defined vector of POI categories, we map users' location histories into a three-dimension tensor, where the three dimensions stand for users, POI categories, and time slots, respectively. Each entry of the tensor stores the number of check-ins about a particular POI category generated by a certain user at a specific time slot. We extract users' features from users' profiles and location histories to build a user-feature matrix. Meanwhile, we model the temporal features using a temporal-category matrix, in which each entry denotes the number of visits of the corresponding category at a time slot. Finally, we model the correlation between each pair of categories using a category-category matrix, in which each entry indicates the number of pages responded from a search engine using the keywords consisting of two corresponding category names. Then, values of these data structures are further respectively normalized. The second procedure models user preferences using a vector of POI categories located by a user's ID and a time slot in the tensor. By taking advantage of the POI category, we alleviate the data sparseness problem for user preferences modelling to some extent. Furthermore, supplementing the missing entries of the tensor through CTD with the aid of above described three matrices, we improve the accuracy of user preferences modelling.

3.2.2. Weighted HITS-based POI rating

This part provides a user with a list of POIs, considering user preferences of current time and social opinions from the local citizens in the querying city. Thus, we propose a weighted HITS-based model for inferring the ratings of POIs. We improve the original HITS-based model by taking into account the friendships among users, taking the similarity between two users as the weight value

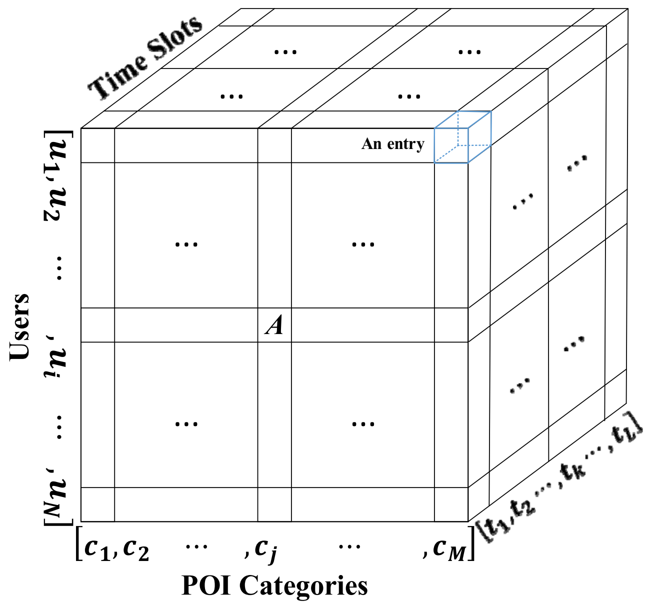

Fig. 6. Structure of user preference tensor.

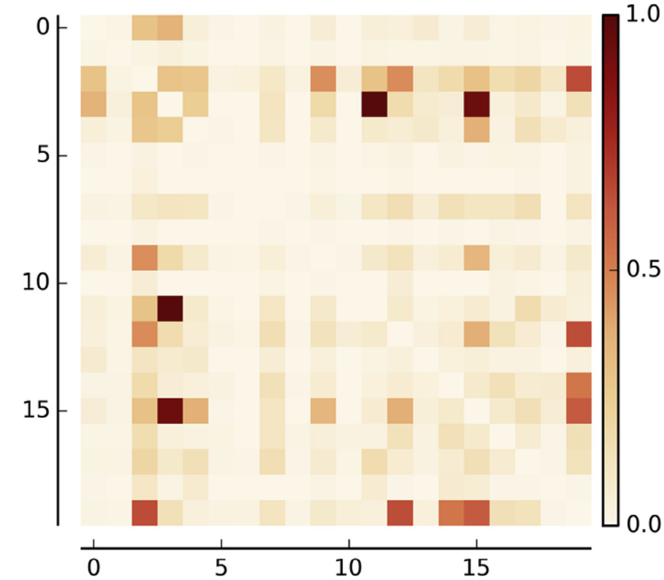

Fig. 7. Correlation between different POI categories.

of corresponding link in the LBSN graph. In addition, the weight value of a link denoting the check-in relationship (from a user to a POI) is assigned the value of a querying user's most interesting category of this POI in the vector of user preferences. Then, after applying the improved HITS-based algorithm, the POIs with relatively high ratings are returned as the POI recommendations within the querying range.

4. CTD for user preferences modelling

In this section, we present user preferences modelling part of our system, which consists of: (1) relevant data structures construction, which prepares a tensor and other three matrices for context-aware tensor decomposition and (2) context-aware tensor completion, which factorizes the tensor with the aid of other three matrices collaboratively, achieving a higher accuracy of user preferences modelling.

4.1. Relevant data structures construction

User preferences tensor: As illustrated in the Fig. 6, user preferences could be modelled by using a 3-dimensional tensor, $\mathcal{A} \in \mathbb{R}^{N \times M \times L}$ with three dimensions denoting N users, M POI categories, and L time slots, respectively. The first dimension denotes users $\boldsymbol{u} = [u_1, u_2, \ldots, u_i, \ldots, u_N]$, who generated the location histories using a GPS-equipped smart device. The second dimension denotes the 2nd-level categories in the category hierarchy $\boldsymbol{c} = [c_1, c_2, \ldots, c_j, \ldots, c_M]$. The last dimension denotes time slots $\boldsymbol{t} = [t_1, t_2, \ldots, t_k, \ldots, t_L]$, where each time slot lasts for an equal period, e.g. 10 am–11 am. Thus, an entry $\mathcal{A}(i, j, k)$ stores the number of visits of user u_i in category c_j and time slot t_k. The value of each entry in tensor $\mathcal{A}$ is then normalized to [0, 1] for decomposition in a later time.

User-feature matrix: User feature set is comprised of user's gender information F_g and other three parts: Location history features F_l, "to-do" features F_t, and "done" features F_d. Features F_l are extracted from a user's location history over the eight categories (*Definition 2* in Section 3.1) of the 1st-level in the category hierarchy, consisting of the numbers of visits f_l ($|f_l| = 8$) over the eight categories separately and the statistical characteristics of f_l (e.g., max value, min value, mean value, standard deviation, sum, median value, etc.). Features F_t and F_d are also the statistical characteristics, which are gained from the numbers of times a tip was added to other users' to-do lists and marked as done, respectively. By putting together the above features of a user into a vector, we then formulate a matrix $X \in \mathbb{R}^{N \times P}$ (P denotes the dimension of user features). We can catch the similarity between two users from Matrix X in terms of their features. Intuitively, users with similar features could have a similar preference.

Temporal-category matrix: We model the temporal features by using a matrix $Y \in \mathbb{R}^{L \times M}$, where each row denotes a time slot and each column denotes a POI category. An entry Y_{kj} contains the number of visits in time slot t_k and category c_j. Matrix Y can reveal the correlation between different time slots in terms of the distribution of visits over different categories. A user may visit some similar category places in t_i or t_j, for these two time slots sharing a similar user visit pattern. For example, the check-in behaviours of a user might be similar at 9–11 am and 2–5 pm, for he is likely to stay at workplace and make check-ins around it, and 12 am and 7 pm are the dinnertime, people might visit some restaurant to have a dinner.

Category-category matrix: The correlation $Cor(c_i,\ c_j)$ $(0 \le i, j \le M, i \neq j)$ between different POI categories can be learned from the search engine using the keywords, which combine two corresponding category names. For example, in order to get the correlation between "Café" and "Chinese Restaurant", we search in a search engine using the keywords, i.e., ⟨ "Café" "Chinese Restaurant" ⟩. Then, the number of results responded as the correlation between these two POI categories. Once the correlation is determined, we can infer the ratings of other categories in the user preferences given the observed POI categories. For instance, as shown in Fig. 7, we present the correlation between each pair of twenty POI categories. In the figure, we can see "Casino" (c_3) has a strong correlation with "Racetrack" (c_{11}) and "Water Park" (c_{15}) as well, which can be interpreted in term of common sense. By putting together $Cor(c_i,\ c_j)$, we formulate matrix $Z \in \mathbb{R}^{M \times M}$. Though tensor $\mathcal{A}$ can capture the correlation between different POI categories to some extent, matrix Z can further intensify the correlation.

4.2. Context-aware tensor completion

We formulate user preferences modelling problem as a collaborative filtering problem, i.e., to estimate the missed entries of $\mathcal{A}$.

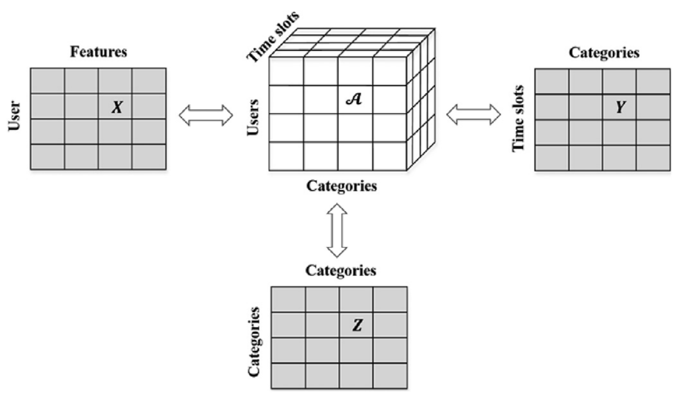

Fig. 8. An illustration of our method based on context-aware tensor decomposition.

Although we can complete the tensor $\mathcal{A}$ by decomposing it solely based on the non-zero entries of $\mathcal{A}$, the results would not be accurate enough. The reason is that the tensor is over sparse. For example, if setting 8 h as a time slot, only 5.17% entries of $\mathcal{A}$ have values.

Thus, to deal with the data sparsity issue and then to achieve a higher accuracy of filling in the missing entries of $\mathcal{A}$, we factorize $\mathcal{A}$ with the aid of X, Y, and Z collaboratively, using a tucker decomposition model [28], as illustrated in Fig. 8. Tensor $\mathcal{A}$ can be decomposed into the multiplication of a core tensor and three matrices, $\mathcal{A} = S \times_U U \times_C C \times_T T$, where the entries of the core tensor $S \in \mathbb{R}^{d_U \times d_C \times d_T}$ show the level of interaction between the different components. Likewise, matrix X can be factorized into the multiplication of two matrices, $X = U \times V$, where $U \in \mathbb{R}^{N \times d_U}$ and $V \in \mathbb{R}^{d_U \times P}$ are low rank latent factors for users and user feature, respectively. In addition, matrix Y can be factorized into the multiplication of two matrices, $Y = T \times C^T$, where $T \in \mathbb{R}^{L \times d_T}$, $C \in \mathbb{R}^{M \times d_C}$ are the low rank latent space of time slots and categories, respectively. Here, matrices Y and Z share the same dimension of category with tensor $\mathcal{A}$. Tensor $\mathcal{A}$ has a common dimension of time with Y and a shared dimension of user with X. Thus, the knowledge from user features, temporal features, and the correlations between different POI categories are propagated into tensor $\mathcal{A}$ by the context-aware tensor decomposition.

$$\mathcal{L}(S,U,C,T,V) = \frac{1}{2}\|\mathcal{A} - S \times_U U \times_C C \times_T T\|^2 + \frac{\lambda_1}{2}\| X - UV\|^2 + \frac{\lambda_2}{2} tr\left(C^T L_Y C\right) + \frac{\lambda_3}{2}\left\|Y - TC^T\right\|^2 + \frac{\lambda_4}{2}\left(\|S\|^2 + \|U\|^2 + \|C\|^2 + \|T\|^2 + \|V\|^2\right) \quad (1)$$

The objective function of our method as Eq. (1), where $\| \cdot \|^2$ denotes the Frobenius norm; $\| \mathcal{A} - S \times_U U \times_C C \times_T T\|^2$ is to control the error of decomposition of tensor; $\| X - UV\|^2$ is to control the error of factorization of X; $\|Y - TC^T\|^2$ is to control the error of factorization of Y; $\|S\|^2 + \|U\|^2 + \|C\|^2 + \|T\|^2 + \|V\|^2$ is a regularization penalty to prevent over fitting; $\lambda_1, \lambda_2, \lambda_3$, and λ_4 are the parameters controlling the contribution of each part during the collaborative decomposition. In addition, $tr(C^T L_Y C)$ is derived from Eq. (2) of the manifold alignment [29,30], where $tr(\cdot)$ denotes the matrix trace; D ($D_{ii} = \Sigma_i Z_{ij}$) is a diagonal matrix, and $L_Z = D - Z$ is the Laplacian matrix of the category correlation graph. This formula guarantees that two POI categories with a high similarity should also have a closer distance in the new latent space C.

$$\begin{aligned}\Sigma_{ij}\|C(i,\cdot) - (j,\cdot)\|^2 Z_{ij} &= \Sigma_k \Sigma_{ij}\|C(i,k) - C(j,k)\|^2 Z_{ij} \\ &= tr\left(C^T (D - Z) C\right) \\ &= tr\left(C^T L_Y C\right) \end{aligned} \quad (2)$$

Algorithm 1: Context-aware Tensor Decomposition (CTD)

Input: Incomplete tensor $\mathcal{A}$, matrices X, Y, and Z, iteration precision ε
Output: Complete tensor $\mathcal{A}_{rec}$
Initialize S, U, C, T, and V with small random values
Set step size η with a value, $D_{ii} = \sum_i Z_{ij}$, and $L_Z = D - Z$
while $Loss_r - Loss_{r+1} > \varepsilon$: ($r$ and $r+1$ denote continuous two iterations)
  foreach $\mathcal{A}_{ijk} \neq 0$:
    $Y_{ijk} = S \times_U U_{i*} \times_C C_{j*} \times_T T_{k*}$
    $U_{i*} \leftarrow U_{i*} - \eta\lambda_4 U_{i*} - \eta(Y_{ijk} - \mathcal{A}_{ijk}) \times S \times_C C_{j*} \times_T T_{k*} - \eta\lambda_1(U_{i*} \times V - X_{i*}) \times V^T$
    $C_{j*} \leftarrow C_{j*} - \eta\lambda_4 C_{j*} - \eta(Y_{ijk} - \mathcal{A}_{ijk}) \times S \times_U U_{i*} \times_T T_{k*} - \eta\lambda_2(L_Z \times C)_{j*} - \eta\lambda_3(C_{j*} \times T^T - Y_{*j}^T) \times T$
    $T_{k*} \leftarrow T_{k*} - \eta\lambda_4 T_{k*} - \eta(Y_{ijk} - \mathcal{A}_{ijk}) \times S \times_U U_{i*} \times_C C_{j*} - \eta\lambda_3(T_{k*} \times C^T - Y_{k*}) \times C$
    $S \leftarrow S - \eta\lambda_4 S - \eta(Y_{ijk} - \mathcal{A}_{ijk}) \times U_{i*} \otimes C_{j*} \otimes T_{k*}$
    $V \leftarrow V - \eta\lambda_4 V - \eta\lambda_1((U_{i*}V - X_{i*})^T \times U_{i*})^T$
  end
end
Result: $\mathcal{A}_{rec} = S \times_U U \times_C C \times_T T$

Fig. 9. Algorithm for the tensor decomposition.

As shown in Fig. 9, to minimize the objective function, we use the gradient descent algorithm. Firstly, we set $\mathcal{A}, X$, Y, and Z as inputs and then initialize S, U, C, T, and V with some small random values. Secondly, after setting the step size η, diagonal matrix D, and the Laplacian matrix L_Z, we apply the gradient descent algorithm to minimize the objective function in an iterative way based on the non-zero entries of $\mathcal{A}$. Finally, we can recover the missing entries of $\mathcal{A}$ by Eq. (3). Given $\mathcal{A}_{rec}$, we can easily obtain the preferences of a user u_i in a certain time slot t_k by retrieving the vector $\mathcal{A}_{rec}(i, :, k)$. Then, user preferences can be used for the weighted HITS-based POI rating in a later time.

$$\mathcal{A}_{rec} = S \times_U U \times_C C \times_T T \quad (3)$$

5. Weighted HITS-based POI rating

In this section, we present the weighted HITS-based POI rating part of our system, which infers the ratings of POIs by considering user preferences of current time and social opinions. User preferences of current time determine which kinds of POIs a user has desire to visit at current time. For example, to have a lunch in a restaurant at noon and to have a good time in a bar at midnight. Intuitively, local citizens could find high quality locations in the querying city as compared with outsiders, resulting in more valuable location records as references. In addition, social relations can influence people's visit behaviours. For example, a user may visit some interesting POIs, which were recommended by his friends. Therefore, we have reason to believe that the recommendations from local citizens are fantastic for both local users and users from out of town by providing their preferences of current time.

In our method, we extract the LBSN graph in the querying city from local users' location histories. As illustrated in Fig. 4, both users and POIs are represented by vertices, the friendships among users are represented by undirected links, and we regard a user's check-in as a directed link from the user to that POI. Each user has a hub value denoting its knowledge and each POI is associated with an authority value indicating its interest level. Intuitively, people who have visited many high quality POIs in a city are more likely to have rich knowledge about this city, and a POI visited by many people with rich knowledge is more likely to be a high-quality POI. Authority and hub values are defined in terms of one another in a mutual recursion. As shown in Eq. (4), a POI's authority value α_{POI} can be computed as the sum of the hub values of the users who have visited this POI, and a user's hub value h_U can be represented by adding the sum of the hub values of this user's friends and the sum of the authority values of the POIs visited by the user. Therefore, users and POIs exhibit a mutually reinforcing relationship and the POIs with relatively high authority values are recommended to a user as recommendations in the querying range.

$$\begin{cases} a_{POI} = W_{U-POI}^T h_U \\ h_U = \beta W_U h_U + (1-\beta) W_{POI-U}^T a_{POI} \end{cases} \tag{4}$$

where $0 < \beta < 1$; W_U is a user-user adjacency matrix defined in Eq. (5); W_{U-POI} is a user-POI adjacency matrix defined in Eq. (6); and W_{POI-U} is a POI-user adjacency matrix defined in Eq. (7).

$$W_U(i,j) = \begin{cases} \dfrac{\lambda sim_{ij} \log sim_{ij}}{\sum_j sim_{ij} \log sim_{ij}}, & \text{if } e_{ij} \in E^f \text{ and } N_i^c > 0 \\ \dfrac{sim_{ij} \log sim_{ij}}{\sum_j sim_{ij} \log sim_{ij}}, & \text{if } e_{ij} \in E^f \text{ and } N_i^c = 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $0 < \lambda < 1$; E^f is the link set among users; $e_{ij} \in E^f$ is a link between user u_i and user u_j; $W_U(i, j)$ denotes an entry of W_U; sim_{ij} is the Cosine similarity between user u_i and user u_j based on the users' feature vectors; and N_i^c denotes the number of POIs that user u_i has checked in at.

$$W_{U-POI}(i,k) = \begin{cases} \dfrac{(1-\lambda) up_{ik} \log up_{ik}}{\sum_i up_{ik} \log up_{ik}}, & \text{if } e_{ij} \in E^c \text{ and } N_i^f > 0 \\ \dfrac{up_{ik} \log up_{ik}}{\sum_i up_{ik} \log up_{ik}}, & \text{if } e_{ij} \in E^c \text{ and } N_i^f > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where E^c is the link set between users and POIs; $e_{ik} \in E^c$ is a directed link that denotes user u_i checks in at POI p_k; $W_{U-POI}(i, k)$ denotes an entry of W_{U-POI}; up_{ij} is the weight value of the directed link from user u_i to POI p_k indicating the querying user's preference about the category of p_k; and N_i^f denotes the number of user u_i's friends.

$$W_{POI-U}(k,i) = \frac{P_{ki} \log P_{ki}}{\sum_i P_{ki} \log P_{ki}} \tag{7}$$

where P_{ki} is the probability of POI p_k checked in by user u_i given all users' check-ins at POI p_k.

6. Experiments

In this section, we first describe the settings of experiments including datasets, baseline approaches, evaluation methods, and parameters in WHBPR. We then report major experiment results on both the effectiveness of user preferences modelling and the performance of POI recommendation.

6.1. Experiment setting

6.1.1. Datasets

Table 1 summarizes the information of four data sets obtained from Foursquare. These 637,700 tips are generated by 78,837 users around the world from May 2008 to July 2011 and each tip is associated with a POI ID, comments, and a timestamp. We use tips as the location records instead of check-ins, for Foursquare blocked the API for crawling a user's check-in data due to the privacy concern. We also obtain 4118,477 friendship links among 75,736 users to build the social relations in LBSN graph. A user's profile, including user ID, name, gender, and home city, could be used to extract a part of user's features. Foursquare defines a category hierarchy for classifying POIs. POI dataset contains 370,159 POIs over those categories.

Table 1
Description on datasets.

Data sets	Properties	Values
Tips	Number of instances	637,700
	Number of users	78,837
(5/6/2008–7/12/2011)	Number of POIs	298,711
	Total duration(h)	27,888
	Average of tips per users	8.1
Friendships	Number of friendship links	4118,477
	Number of users	75,736
	Average of friends per user	54.4
Users	Number of instances	980,325
POIs	Number of instances	370,159
	Number of all categories	352
	Number of 1st-level categories	8
	Number of 2nd-level categories	241
	Number of 3rd-level categories	103

From the tips dataset, we choose the local citizens and out-of-towners in NYC and study the POI recommendation made for these users in NYC, respectively. To guarantee the validity of experimental results, we select the users who have over 24 tips totally and over 8 tips in the querying city. Table 2 shows the details about these users, where the ground truth density denotes the value of the number of ground truth POIs divided by the number of candidate POIs in a user's querying range.

6.1.2. Baseline approaches

As shown in Table 3, we compare our method with following five baseline approaches, where the first three baseline approaches are existing recommender methods, the last method (TAP-F) represents our method, the fifth method (TAP) means our method without using friendships among users, and the fourth (TAP w/o TI) means the method TAP without considering temporal influence.

Tensor-decomposition-based collaborative filtering (TD-CF): This method first projects users' location histories into a 3- dimension tensor, where the three dimensions stand for users, POIs, and time slots, respectively, and then completes the missing entries of the tensor through tucker decomposition model [28]. Then, in the querying geospatial range and current time, the top-N ranked POIs are returned as the recommendations for the user.

Location-based and preference-aware collaborative filtering (LP-CF) [23]: This baseline consists of two parts: offline modelling and online recommendation. The offline modelling part models user preferences using a weighted category hierarchy (WCH) and infers the expertise of each user in a city with respect to different categories. The online recommendation part selects candidate local experts in a geospatial range that matches the user's preference and then applies a user-based CF method to infer the rating of an unvisited POI.

Hypertext-induced-topic-search-based recommendation (HITS) [11]: This baseline first builds a user-POI network representing the users' location history and then applies HITS algorithm for discovering experienced users and interesting POIs in a LBSN. Finally, in the querying geospatial range, the top-N ranked POIs are returned as the recommendations for a querying user.

6.1.3. Evaluation methods

We evaluate both the effectiveness of user preferences modelling and the performance of POI recommendation.

User preferences modelling effectiveness: We set 8 h as a time slot for alleviating the data sparseness problem. Then the size of the tensor $\mathcal{A}$ is $1774 \times 241 \times 3$. To make the effectiveness evaluation, we randomly remove a% non-zero entries from the tensor and supplement these entries using CTD model. We then use the

Table 2
Statistics of the selected users.

Status	Total users	Tips in NYC	All tips	Average tips	Average tips in NYC	Ground truth density
Home-town	1227	46,648	60,156	49.03	38.02	0.0051
Out-of-town	547	10,691	48,347	88.39	19.54	0.0039

Table 3
Comparison between baseline methods and ours.

Methods	HITS based	POI category	Preference completion	Temporal influence	Friendship
TD-CF				√	
LP-CF		√			
HITS	√				
TAP w/o TI	√	√	√		
TAP	√	√	√	√	
TAP-F	√	√	√	√	√

original values of the removed entries as ground truth to evaluate the inferred values. We compare our method with following two baselines: (1) ABT complements a missing entry with the average of all non-zero entries that belong to the corresponding time slot; (2) MF fills the missing entries by factorizing the user-category matrix time-slot by time-slot. We also study the contribution of matrices X, Y and Z in helping supplement the missing entries. As defined in Eqs. (8) and (9), RMSE (i.e., root mean square error) and MAE (i.e., mean absolute error) are used as evaluation metrics, where n is the number of the estimated entries; y_i' is an inference and y_i is the ground truth.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - y_i'\right)^2}{n}} \tag{8}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|y_i - y_i'\right|}{n} \tag{9}$$

POI recommendation performance: Since our system is designed for both home-town recommendation and out-of-town recommendation, we evaluate POI recommendation performance of our system under these two scenarios, respectively. For each user, we randomly mark off 10% of his location records in the querying city (i.e., NYC) as development data to tune parameters, mark off 20% of his location records in the querying city as testing data to evaluate the recommendation effectiveness, and use the remaining 70% location records as training data to learn user preferences and to help building the LBSN graph.

As shown in Fig. 10, the black dots denote the POIs that a user visited in the querying city, and the corresponding minimum bounding rectangle (MBR) of these POIs determines the range to produce candidate POIs. To simulate a real scenario, we expand the MBR to a new bigger spatial range (i.e., the querying range), which can be specified by a user or automatically determined based on user's current location in the user's recommendation request in the real world. Given user preferences of current time slot and the querying range, our system can return the top-N POIs as POI recommendations indicated by the striped dots in Fig. 10. Then we evaluate the performance of our system by using following two measures, i.e., precision and recall, according to Eqs. (10) and (11), respectively.

$$Precision@N = \frac{|recovered\ ground\ truths|}{|recommendations|} \tag{10}$$

$$Recall@N = \frac{|recovered\ ground\ truths|}{|ground\ truths|} \tag{11}$$

where *Precision@N* denotes the precision metric when the number of requested recommendations is assigned to *N*; likewise; *Recall@N* indicates the recall metric when the number of requested recommendations is assigned to *N*.

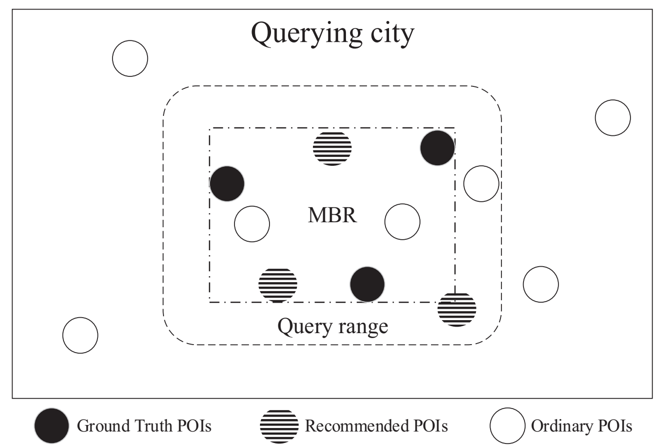

Fig. 10. POI recommendation effectiveness evaluation method.

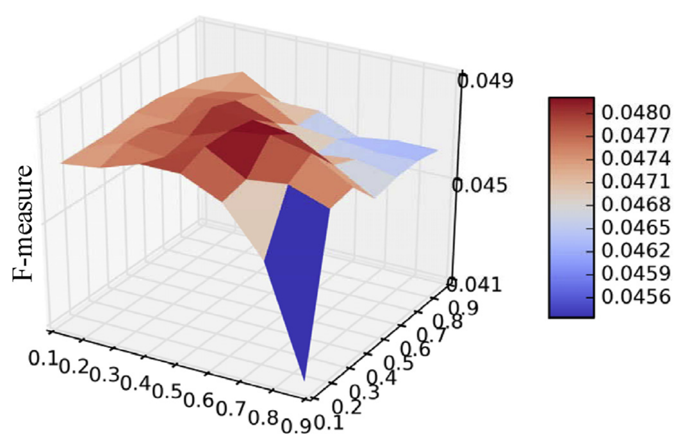

Fig. 11. Impact of the two parameters.

6.1.4. Impact of the parameters

To study the impact of the two parameters (i.e., λ and β) in WHBPR and further to understand the roles of user preferences, social influence from friends, and the influence from POIs in achieving the optimal WHBPR performance, we tried different setups for these two parameters. We only show the F-measure@10 results for the out-of-town recommendation on the development data since the experiment results for the home-town recommendation are similar and the space constraint. As shown in Fig. 11, we

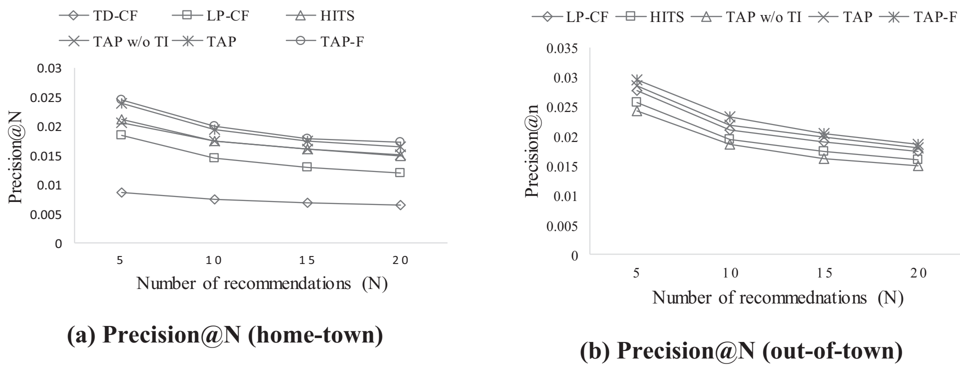

Fig. 12. Precision@N w.r.t Recommendation Numbers.

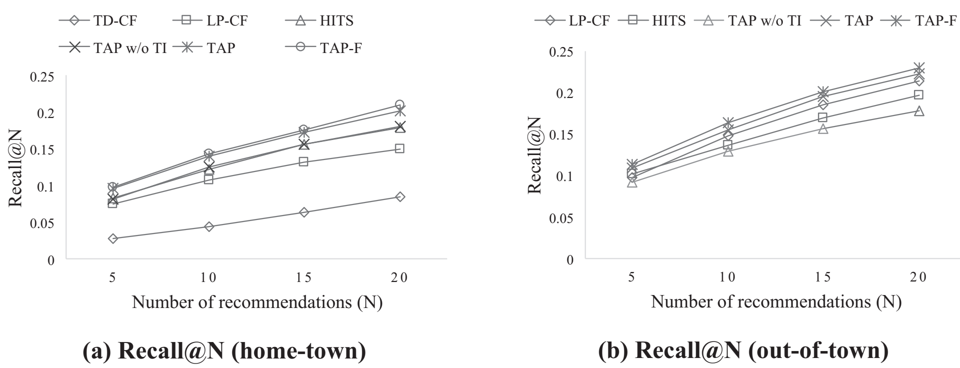

Fig. 13. Recall@N w.r.t Recommendation Numbers.

Table 4
Effectiveness of comparison of different methods for user preferences complement.

Methods	MAE	RMSE
ABT	1.0303	2.9812
MF	2.3742	3.6593
TD	1.4892	3.3356
TD+X	1.2324	3.1051
TD+Y	1.4705	3.3273
TD+Z	1.0674	3.0687
TD+X+Y	1.3115	3.1806
TD+Y+Z	1.0628	2.9328
TD+X+Z	1.0399	3.0398
TD+X+Y+Z (CTD)	1.0065	2.9134

observe that the F-measure results are unideal when both λ and β are assigned relatively large values. The possible reason is that increasing both these parameters undermines the influences of user preferences and POIs, simultaneously. Further-more, we also observe that the F-measure results are good when λ is assigned a relatively small value, while β is assigned an arbitrary value. Thus, we have reasons to believe that user preferences play a relatively important role in contributing to the optimal recommendation. As shown in Fig. 11, we choose the optimal point ($\lambda = 0.5,\ \beta = 0.375$) as the best parameter setting.

6.2. Results

6.2.1. Evaluation on the CTD

We set $a = 20$, and adopt 5-fold cross validation to generate test cases. Table 4 presents evaluation results, in which CTD has higher performance than other methods. In addition, the user preferences modelling effectiveness of TD is lower than ABT through decomposing $\mathcal{A}$ solely based on its own non-zero entries. However, by taking user features, temporal features, and correlation between different POI categories into account, we can obtain more accurate results. Due to the sparsity issue of tensor $\mathcal{A}$, it is difficult for CF to catch the correlations (or similarities) between users, between categories, and between time slots, respectively. Thus, with the aid of matrices X, Y and Z, which have rich information with respect to user, category, and time slot, we can achieve a more accurate user preferences modelling.

6.2.2. Evaluation on the WHBPR

In this part, we present the experimental results of comparing recommendation methods with well-tuned parameters. Figs. 12 and 13 show the average recall and precision of different methods varying in the number of recommended POIs (N). From the figures, we can observe that our method always exhibits the best performance in terms of precision and recall under all values of Ns, showing the strength of combining the three factors of user preferences, social influence and temporal influence. First, TAP-F has better performance than TAP, showing the advantage of taking friendships among users into account. Second, TAP outperforms TAP w/o TI, justifying the benefit brought by considering temporal influence. Third, in the scenario of home-of-town recommendation, TAP w/o TI exceeds LP-CF due to the advantage of user preferences complement when users have a few location records, which helps us to understand users better. Finally, TD-CF drops behind other methods, showing the advantage of using POI categories to model user preferences and to deal with data sparseness problem.

Fig. 14 plots the F-measure of our method in the two scenarios changing over the ground truth density ranges, where F-measure $= \frac{2\times precision\times recall}{precison+recall}$. The results match our intuition that the denser ground truths are in a querying range, the better recommendation performance will be. However, it is somehow surprising that the F-measure of out-of-town is higher than that of home-town though the average of ground truth density of home-town ($=0.0051$) exceeds that of out-of-town ($=0.0039$). There are

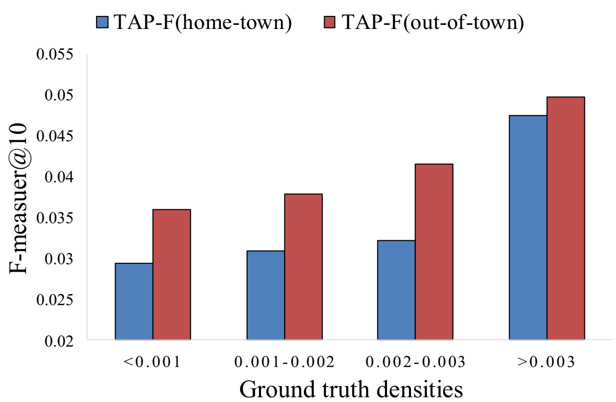

Fig. 14. F-measure@10 w.r.t ground truth densities.

two reasons following can explain this phenomenon. On the one hand, the possible reason is that the average of location records generated by the out-of-town users is almost double that generated by home-town users. Thus, the more location records, the more accurate of user preferences that we can model, thereby the better recommendation performance. Thus, we can also observe that LP-CF has a better performance than HITS in the scenario of out-of-town recommendation for this reason. On the other hand, the POIs to be visited by a user are more predicable when a user travels to a new city. Intuitively, people usually visit some well-known places (e.g., places of historic interest and scenic beauty) in a new city to them, while would travel to any locations in a city they are very familiar with. As a result, we can observe that HITS overtakes TAP w/o TI in the scenario of out-of-town recommendation.

7. Conclusions and future work

In this paper, we present a personalized POI recommendation system for providing people with POI recommendations within a specified geospatial range based on (1) the temporal-aware user preferences learnt from his tips records and (2) the social opinions from the users in the querying city. Our method effectively overcomes the challenges arising from data sparsity (especially that arisen from travel locality) by modelling user preferences from the POI categories and inferring ratings of POIs through a weighted HITS-based model. Therefore, our system can facilitate people travel not only near their living and working areas but also to a new city. We conduct extensive experiments to evaluate the effectiveness of user preferences modelling and validate its advantages beyond other baseline methods by having the aid of external information (i.e., user-feature matrix, temporal-category matrix, and category-category matrix). We also evaluate the performance of POI recommendation using extensive experiments based on four real datasets collected from Foursquare. The results revealed the advantages of our method over other POI recommendation methods for both home-town and out-of-town recommendations. In the future, we are going to deal with cold start issue [31], interest drift problem [32], and incorporate more context information, e.g., weather.

Acknowledgements

This work was funded by China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5), the Natural Science Foundation of China (No. 61332017), the Science and Technology Department of Zhejiang Province (No. 2015C33002).

References

[1] Y. Zheng, Location-based social networks: Users, Computing with Spatial Trajectories, Springer, 2011, pp. 243–276.
[2] C. Desrosiers, G. Karypis, A Comprehensive Survey of Neighborhood-Based Recommendation methods, in: Recomm. Syst. Handb., Springer, 2011, pp. 107–144.
[3] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of Twenty-Second Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1999, pp. 230–237.
[4] D. Lemire, A. Maclachlan, Slope one predictors for online rating-based collaborative filtering, in: SDM, SIAM, 2005: pp. 1–5.
[5] H. Zhang, Y. Yang, Z. Zhang, CTS: Combine temporal influence and spatial influence for time-aware POI recommendation, in: International Conference of Young Computer Scientists, Engineers and Educators, Springer, 2016, pp. 272–286.
[6] C. Cheng, H. Yang, I. King, M.R. Lyu, Fused matrix factorization with geographical and social influence in location-based social networks, in: Twenty-Sixth AAAI Conference on Artificial Intelligent, 2012.
[7] S. Rendle, Factorization machines with libFM, ACM Trans. Intell. Syst. Technol 3 (2012) 57.
[8] H. Gao, J. Tang, H. Liu, gSCorr: modeling geo-social correlations for new check--ins on location-based social networks, in: Proceeding of Twenty-First ACM International Conference on Information and Knowledge Management, ACM, 2012, pp. 1582–1586.
[9] M. Ye, P. Yin, W.-C. Lee, D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, in: Proceeding of Thirty-Fourth International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 325–334.
[10] H. Gao, J. Tang, X. Hu, H. Liu, Exploring temporal effects for location recommendation on location-based social networks, in: Proceeding of Seventh ACM Conference Recommender System, ACM, 2013, pp. 93–100.
[11] J. Bao, Y. Zheng, D. Wilkie, M. Mokbel, Recommendations in location-based social networks: a survey, Geoinformatica 19 (2015) 525–565.
[12] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, J. Kleinberg, Automatic resource compilation by analyzing hyperlink structure and associated text, Comput. Networks ISDN Syst. 30 (1998) 65–74.
[13] J.M. Kleinberg, Authoritative sources in a hyperlinked environment, J. ACM. 46 (1999) 604–632.
[14] X. Long, J. Joshi, A HITS-based POI recommendation algorithm for location-based social networks, Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ACM2013, pp. 642–647.
[15] A. Noulas, S. Scellato, N. Lathia, C. Mascolo, A random walk around the city: New venue recommendation in location-based social networks, in: Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on Social Computing, IEEE, 2012, pp. 144–153.
[16] Q. Yuan, G. Cong, A. Sun, Graph-based point-of-interest recommendation with geographical and temporal influences, in: Proceeding of Twenty-Third ACM International Conference on Information Knowledge Management, ACM, 2014, pp. 659–668.
[17] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, in: Proceeding of Fourth ACM Conference Recommender System, ACM, 2010, pp. 79–86.
[18] L. Yao, Q.Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, Q. He, Context-aware point-of-interest recommendation using tensor factorization with social regularization, in: Proceedings of Thirty-Eighth International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 1007–1010.
[19] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, Knowl. Data Eng. IEEE Trans. 17 (2005) 734–749.
[20] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of Tenth International Conference of World Wide Web, ACM, 2001, pp. 285–295.
[21] K. Ji, H. Shen, Making recommendations from top-N user-item subgroups, Neurocomputing 165 (2015) 228–237. http://dx.doi.org/10.1016/j.neucom.2015.03.013.
[22] D. Agarwal, B.-C. Chen, fLDA: matrix factorization through latent Dirichlet allocation, in: Proceedings of Third ACM International Conference on Web Search Data Mining, ACM, 2010, pp. 91–100.
[23] J. Bao, Y. Zheng, M.F. Mokbel, Location-based and preference-aware recommendation using sparse geo-social networking data, in: Proceeding of Twentieth International Conference on Advance Geographical Information System, ACM, 2012, pp. 199–208.
[24] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: bringing order to the web, (1999).
[25] Y. Zheng, L. Zhang, X. Xie, W.-Y. Ma, Mining interesting locations and travel sequences from GPS trajectories, in: Proceedings of Eighteenth International Conference on World Wide Web, ACM, 2009, pp. 791–800.
[26] R. Raymond, T. Sugiura, K. Tsubouchi, Location recommendation based on location history and spatio-temporal correlations for an on-demand bus system, in: Proceedings of Nineteenth ACM SIGSPATIAL International Conference of Advance Geographical Information System, ACM, 2011, pp. 377–380.

[27] M.A. Vasconcelos, S. Ricci, J. Almeida, F. Benevenuto, V. Almeida, Tips, dones and todos: uncovering user profiles in foursquare, in: Proceedings of Fifth ACM International Conference on Web Search Data Mining, ACM, 2012, pp. 653–662.
[28] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (2009) 455–500.
[29] Y. Ma, Y. Fu, Manifold Learning Theory and Applications, CRC Press, 2011.
[30] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, E. Chang, Diagnosing New York city's noises with ubiquitous data, in: Proceeding of 2014 ACM International Joint Conference Pervasive Ubiquitous Computing, ACM, 2014, pp. 715–725.
[31] F. Peng, J. Lu, Y. Wang, R. Yi-Da Xu, C. Ma, J. Yang, N-dimensional Markov random field prior for cold-start recommendation, Neurocomputing 191 (2016) 187–199.
[32] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, X. Zhou, Geo-SAGE: a geographical sparse additive generative model for spatial item recommendation, in: Proceeding of Twenty-First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1255–1264.
[33] M. Jiang, P. Cui, F. Wang, X. Xu, W. Zhu, S. Yang, Fema: flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery, in: Proceedings of the Twentieth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (ACM2014), pp. 1186–1195.
[34] S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, 2010, pp. 81–90.
[35] A. Karatzoglou, X. Amatriain, L. Baltrunas, N. Oliver, Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering, in: ACM Conference on Recommender Systems, Barcelona, Spain, Recsys, 2010, pp. 79–86.
[36] J. Zhang, J. Tang, B. Liang, Z. Yang, Recommendation over a heterogeneous social network, in: International Conference on Web-Age Information Management, 2008, pp. 309–316.
[37] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, S. Yang, Social recommendation with cross-domain transferable knowledge, IEEE Trans. Knowl. Data Eng. 27 (2015) 3084–3097.

Yuankai Ying is currently a master candidate in the College of Computer Science and Technology at Zhejiang University, Hangzhou, China. He is interested in POI recommendation using check-in data.

Ling Chen received a Ph.D. degree from Zhejiang University, Hangzhou, China. He is Professor of Computer Science at Zhejiang University. His research interests pervasive computing and database.

Gencai Chen received a M.Sc. degree from Hangzhou University, Hangzhou, China. He is Professor of Computer Science at Zhejiang University. His research interests AI, computer supported collaborative work system and database.