Collaborative Location Recommendation by Integrating Multi-dimensional Contextual Information

LINA YAO, University of New South Wales
QUAN Z. SHENG, Macquarie University
XIANZHI WANG, Singapore Management University
WEI EMMA ZHANG, Macquarie University
YONGRUI QIN, The University of Huddersfield

Point-of-Interest (POI) recommendation is a new type of recommendation task that comes along with the prevalence of location-based social networks and services in recent years. Compared with traditional recommendation tasks, POI recommendation focuses more on making personalized and context-aware recommendations to improve user experience. Traditionally, the most commonly used contextual information includes geographical and social context information. However, the increasing availability of check-in data makes it possible to design more effective location recommendation applications by modeling and integrating comprehensive types of contextual information, especially the temporal information. In this paper, we propose a collaborative filtering method based on Tensor Factorization, a generalization of the Matrix Factorization approach, to model the multi-dimensional contextual information. Tensor Factorization naturally extends Matrix Factorization by increasing the dimensionality of concerns, within which the three-dimensional model is the one most popularly used. Our method exploits a high-order tensor to fuse heterogeneous contextual information about users' check-ins instead of the traditional two dimensional user-location matrix. The factorization of this tensor leads to a more *compact* model of the data which is naturally suitable for integrating contextual information to make POI recommendations. Based on the model, we further improve the recommendation accuracy by utilizing the internal relations within users and locations to regularize the latent factors. Experimental results on a large real-world dataset demonstrate the effectiveness of our approach.

1 INTRODUCTION

Over the past few years, location-based social network (LBSN) applications have attracted thousands of millions of users with the ever growing popularization of mobile devices (especially smart phones) and the advances in wireless communication technologies. Nowadays, people are increasingly using LBSN services to contact friends, explore places (e.g., restaurants, shops, cinemas), and share information about their everyday lives. LBSNs not only provide users with the platforms for social interaction but also recommend new items, typically places, using the rich contextual location information (e.g., check-in data). By mining users' preferences on locations, the location-based recommenders can suggest places for recreation and healthcare services, as well as be applied in smart city scenarios. For example, after a patient visiting an orthopaedic clinic, the services for physical therapy might be recommended. In a smart home, when a coffee pot is detected being used, a nearby sugar jar might be recommended to the user.

Currently, most commonly used LBSN services, such as Foursquare, Gowalla, and Brightkite, obtain user locations through their check-in activities. These check-in data contain rich contextual information about users' preferences. The availability of check-in data makes it possible to design effective location recommendation applications by modeling and integrating such rich contextual information. From this point of view, many significant research efforts have been conducted on the Point-of-Interest (POI) recommendation, which recommends users with places of potential interest in the near future [8, 12, 32, 36, 44, 45, 49]. By analyzing users' check-in history, these approaches identify the informative patterns that are useful for POI recommendation by exploiting the social constraints, and the geographical and temporal influences. Generally, the more clues considered, the more accurate is the recommendation. For this reason, some recent research efforts are dedicated to integrating multiple types of contextual information for the POI recommendation. For instance, Cheng et al. [6] develop a multi-center Gaussian model that makes recommendation by fusing social influence and geographical influence. Bao et al. [2] propose a location-based preference-aware recommender system by using the sparse geo-social data. Yuan et al. [46] develop a graph-based POI recommendation framework, which considers the geographical and temporal influences of users' visits to places to make more accurate recommendation. Gao et al. [14] further consider the social influence, and incorporate the social context with the temporal and spatial influences under a unified matrix factorization framework.

To fully exploit the multi-dimensional contextual information in the check-in data, we propose a *tensor-based* unified framework for effective POI recommendation. Instead of modeling the check-in data as a two dimensional user-location matrix (as done by the existing approaches [1, 19, 30, 42]), we address the POI recommendation problem by modeling the multi-dimensional contextual information of the check-in data as a three-order tensor, which represents the relations across users, locations and time frames, respectively. In particular, we propose a regularization based tensor optimization framework, where we further incorporate users' social connections and spatial proximity as additional regularization terms.

Compared with existing recommendation frameworks, our proposed tensor based factorization approach has several advantages. Firstly, tensor naturally models users' check-in activities by incorporating temporal dimension, along with user-location check-in matrix, in a compact representation. Secondly, tensor based factorization can be considered as a generalization of the traditional two-dimensional matrix factorization, which inherits the advantages, such as computational simplicity and robustness to the sparse check-in data, of the matrix factorization-based methods. Finally, the proposed model can be easily extended to embrace more contextual information with higher order tensor by integrating more relations of check-in data. It should be noted that in a very similar work, Karatzoglou et al. [16] also propose a tensor-based context-aware recommendation framework using Tucker's decomposition. However, they do not consider the internal correlations among the contextual entities. In contrast, we employ a three-order tensor to uncover the hidden dependencies within the multi-dimensional contextual information. We further explore the diverse, multi-dimensional contextual information contained in the check-in data to achieve more accurate POI recommendations. To sum up, this paper makes the following main contributions:

- We elaborate how multi-dimensional contextual information can improve a POI recommendation system and propose to exploit the high order unified tensor to interpret the multi-dimensional contextual information, including spatial influence, temporal dependency, and users' social constraints of the check-in data, in a compact manner. Specially, the tensor construction can address the temporal dependency of users' dynamic checking-in activities [7, 41].
- We systematically illustrate the design and generalization of the tensor factorization objective function with social and spatial regularization terms. The proposed method is generic and flexible to integrate more auxiliary information.
- We extensively evaluate our proposed method using a public real-life dataset. The experimental results demonstrate the effectiveness of our approach. More specifically, our experimental results show that: i) compared to the six baseline two-dimensional matrix factorization based models, our tensor based method can significantly improve the overall recommendation accuracy, ii) our tensor-based method also outperforms other six baseline temporal based recommendation methods, and iii) the spatial proximity between locations has more significant impact than users' social connections, and utilizing both of these influences can boost the recommendation performance.

The remainder of this paper is organized as follows. Section 2 provides an overview of the major approaches closely related to our proposed method. Section 3 describes some preliminaries including empirical data analysis and tensor factorization. Section 4 gives the technical details on our proposed solution based on regularized low-rank tensor based matrix factorization. The results of an empirical analysis are presented in Section 5, followed by the conclusion and future work in Section 6.

2 RELATED WORK

In this section, we overview the major techniques that are relevant to our work.

2.1 Recommendation Techniques

Recommender systems have been widely adopted by electronic retailers and content providers as a means of matching customers with the most appropriate products by analyzing the patterns of user interests. Among all recommendation approaches, collaborative filtering (CF) is one of the most popular methods. The CF-based approaches rely only on past user behaviors without requiring the creation of explicit profiles. Therefore, when compared with the content filtering approach, CF is more suitable for dealing with recommendation problems in different domains. CF-based approaches can be divided into two categories, namely the *memory-based* CF (or the neighborhood methods) and the *model-based* methods (or latent factor models). The former utilizes user-item interaction data (e.g., ratings, check-ins) to calculate the similarity between users or items to make recommendations [35]. The latter tries to explain the interaction by characterizing both items and users with patterns. Matrix Factorization (MF) is a latest realization of the latent factor models that shows great potential in various recommendation application domains and gains popularity in recent years. MF-based approaches not only features high prediction accuracy, but also presents good scalability and flexibility for modeling various real-life situations.

2.2 Methods for POI Recommendation

As a special type of recommender systems, POI recommendation has become increasingly popular with the rise of social network services and the ubiquity of hand-held smart devices. Currently, most POI recommendation studies focus on exploiting the various and abundant contextual information to make more accurate recommendation. Geographical information and social influence are the two types of information mostly considered by the POI recommendation approaches. For example, Ye et al. [43] [42] harness user-based CF to explore both the social and geographical influences under a linear framework for POI recommendation. Cheng et al. [6] model the geographical information by a Gaussian mixture model (GMM) and incorporate this model into a matrix factorization based framework. Liu et al. [26] introduce a probabilistic recommendation framework that employs three types of data, the geographical influence on a user's check-in behavior, the user mobility pattern, and the user check-in count data, for location recommendation. The authors in [47] consider social and geographical influence from both user and location perspectives, and develop a recommendation framework by fusing Kernel density estimation. Hu et al. [15] propose to improve the recommendation accuracy by incorporating the geographical neighborhood information. In a very recent work, Liu et al. [29] systematically evaluate 12 state-of-the-art POI recommendation models.

The MF methods are especially suitable for POI recommendation for the following reasons. Firstly, the check-in data used for POI recommendation usually contains different types of inputs, which can be easily placed in a matrix with one dimension representing users and the other dimensions representing items of interest in a MF model. Secondly, MF allows the incorporation of additional information, including both explicit and implicit feedbacks for the recommendation. It is intuitive that considering more clues could generally improve the accuracy of POI recommendation. Our proposed method for POI recommendation belongs to the collaborative filtering recommendation frameworks and stems from the matrix factorization methods for CF-based recommendation. Some other MF-related works are described as follows. Koren et al. [22] propose to factorize the user-item rating matrix of Netflix movie ratings. The approach uses SVD models after the related Singular Value Decomposition. The key idea behind the SVD models is to factorize the rating matrix to a product of two lower rank matrices. Two approaches, namely *stochastic gradient descent* and *alternating least squares*, are used to optimize the objective function. The authors in [3] propose to only use user-location check-in data in regularized MF based POI recommendation. However, they use binary and binning preference definitions to derive pseudo ratings from check-in data due to the lack of explicit ratings.

2.3 POI Recommendation Considering Spatial Information

The spatial information has been exploited as an important means of achieving more accurate POI recommendation in previous research. Typically, the authors in [27] incorporate location-awareness in a topic-based POI recommender system and propose a Topic and Location-aware probabilistic matrix factorization (TL-PMF) method. The method shows superior performance to several baseline methods and demonstrates the significance of spatial information for POI recommendation. Later, Liu et al. [28] continue to propose a general geographical probabilistic factor model (Geo-PFM) framework which strategically takes into consideration the geographical influences on a user's check-in behavior together with several other factors to achieve better results. Both Lian et al. [25] and Li et al. [24] focus on the scarcity issue of the check-in data, but the former leverages the spatial clustering phenomenon to enhance a weighted matrix factorization model for POI recommendation, and the latter uses a ranking-based geographical factorization method for POI recommendation. More recently, Chen et al. [5] introduce information coverage to encode the location categories of POIs in a city and propose a top-K location category based POI recommendation approach. The results show that the performance can be improved through more effective organization and usage of the location information. Lian et. al [25] propose the GeoMF framework, wherein the clustering phenomenon in human mobilities is exploited by two-dimensional kernel density estimation to deal with data sparsity and boost performance. Liu et. al [28] design a general geographical probabilistic factor model for POI recommendation. They capture human check-in behaviors using Poisson distribution embedded into NMF-based factor model for better interpreting the geographical influence on the mobility model. Wang et. al [38] design a hybrid predictive model, which not only takes the regularity of human movements and social conformity into account, but also explores the mutual reinforcement of the both factors.

2.4 POI Recommendation Considering Temporal Information

Besides the geographical information and social influence, some other information, such as temporal information, has been exploited [20] [39]. Cho et al. [8] propose a generative model to predict users' locations by modeling periodical mobility patterns. Yuan et al. [45] propose a collaborative recommendation model by incorporating temporal information. Gao et al. [13] develop a temporal recommendation framework by exploiting different temporal patterns such as consecutiveness and non-uniformness. As a further step, more researchers have been exploring the integration of multiple contexts in more efficient manner to benefit the recommendation performance. For example, Yuan et al. [46] propose a graph-based recommendation framework by taking both geographical and temporal influences into account. Since tensor is a natural choice to model high order contextual information in POI recommendation, Gao et al. [14] propose a two dimensional matrix factorization framework to recommend point-of-interest by integrating social, temporal and spatial influence together. Karatzoglou et al. [16] also propose a tensor based recommendation framework, which takes the extra contextual information of time into consideration.

2.5 Summary

As a summary, traditional recommendation methods do not focus on incorporating rich contextual information, while most existing recommendation approaches for POI recommendation focus on only the geographical and social contexts. Compared to these approaches, our proposed approach employs high order tensor to represent the triadic relations among users, locations, and time within the check-in data. Compared with the traditional two-dimensional user-location matrix, our method incorporates richer contextual information, which ensures high recommendation accuracy. In addition, we specially present a unified, regularized tensor factorization method along with graph Laplacians induced from users' social constraints and spatial proximity to predict the check-in probability of a location, and to finally solve the POI recommendation problem. It should be noted that the work proposed in [33] is most similar to ours. In this work, the authors develop several types of tensor-based models for integrating within and cross objects relations. Compared to the above work, we propose a generic solution to solve a specific domain, i.e., the POI recommendation problem. In addition, we provide an informative optimization process and perform the complexity analysis. We also present extensive empirical studies to evaluate the proposed approach.

3 PRELIMINARIES

In this section, we first analyze the characteristics of the check-in data from social, spatial, and temporal dimensions' perspectives. We then briefly introduce the basic matrix factorization based recommendation framework before delving to our proposed regularized tensor-based framework.

In this paper, scalars will be denoted by lowercase letters (e.g., m), vectors by boldface lowercase letters (e.g., $\mathbf{u}$), matrices including tensor by boldface capital letters (e.g., $\mathbf{U}$ and matrix $\mathbf{R}$), and tensors by boldface Calligraphy (e.g., $\mathcal{R}$). We list the main notations used in this paper in Table 1.

Table 1. Notations

Notation	Explanation
m	The number of users
n	The number of locations
q	The number of check-ins
d	The lower dimension used in both matrix factorization and tensor factorization
$\mathbf{R} \in \mathbb{R}^{m \times n}$	The user-location matrix in matrix factorization
$\mathcal{R} \in \mathbb{R}^{m \times n \times q}$	The user-location matrix in tensor factorization
$\mathcal{R}_{ijk}$	A piece of check-in information in tensor factorization
$\mathbf{U} \in \mathbb{R}^{m \times d}$	The user feature matrix
$\mathbf{V} \in \mathbb{R}^{n \times d}$	The location feature matrix
$\mathbf{T} \in \mathbb{R}^{k \times d}$	The time feature matrix
$\mathbf{U}_i \in \mathbb{R}^{m \times d}$	User i's feature in the feature matrix
$\mathbf{V}_j \in \mathbb{R}^{n \times d}$	Location j's feature in the feature matrix
λ_U	The regularization parameter for $\mathbf{U}$
λ_V	The regularization parameter for $\mathbf{V}$
$\circ$	The outer product operation
$\odot$	The Khatri-Rao operation
$*$	The Hadamard product
$\lVert \cdot \rVert_F$	The Frobenius norm

3.1 Empirical Analysis

LBSN users share their current locations by checking-in on websites such as Foursquare, Facebook, Gowalla, Brightkite etc. The check-in data can be collected to examine the exact location a user has visited [8]. To gain a better understanding of users' check-in activities, we study the check-in data from Brightkite[1] (see Section 5.1.1 for details) in terms of social, spatial and temporal dimensions in the check-in data.

We first examine the effect of spatial influence to users' check-ins. Specifically, for each user, we calculate the distance between every pair of her visited locations. Then, we aggregate the results of all users and plot the number of check-ins as a function of distance in Figure 1 (a). From the figure, we observe that the probability that a person checks-in at two mutually distant POIs follows the power-law distribution. This observation is consistent with the observation made in [43]. The results show that users are more likely to visit locations close to their visited locations, and thus the locations visited by a user form several spatial clusters. To better illustrate the spatial aggregation effect, we randomly select a user from the data set and plot the distribution of all the user's check-in locations in Figure 1 (b). From the figure, we observe that most of the locations visited by the user can be grouped into several dominant geographical regions.

Next, we analyze the temporal patterns of users' check-ins. The check-in data also show the periodical features depending on the types of POIs (see Figure 1 (c) - (f)). We only use the day information of check-in data and decompose a day into 24 hours. For example, restaurants' peak time may be in the lunch hours, yet nightclubs or cinemas are mostly during night times and weekends. In particular, Figure 1 (c) shows the check-in frequencies of a specific user at a particular location over 24 hours a day and 7 days a week, where we use colors to denote different check-in frequencies. Figure 1 (d) particularly depicts the users' check-in distributions of a specific location over 24 hours. It shows that the location in the figure has more check-ins between 10am and 8pm (e.g., shopping mall). In addition, Figure 1 (e) shows the check-in overlapping probability (i.e.,

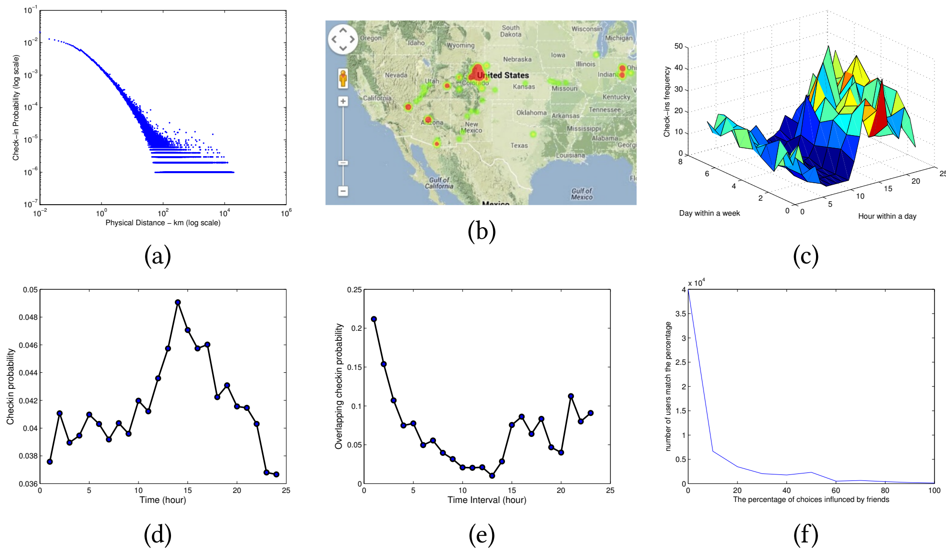


Fig. 1. Illustration of check-in patterns along spatial and temporal dimensions (a) check-in probability distribution w.r.t. physical distance; (b) the clustered locations visited by a specific user; (c) check-in distribution of a specific user over 24 hours a day and 7 days of a week; (d) check-in distribution probability of top 1,000 active users over 24 hours at a particular location; (e) check-in overlapping probability over time distance (in one-hour time window); and (f) check-in patterns in terms of friendship influence

the probability that a user check-in at a same location within different time windows) based on one-hour time window. We observe that the probability at [0,1] is the highest (e.g., at home); and it declines as the time difference increases (e.g., a user may check in at various locations like bus stations and offices in the morning), and increases again after lunch time (e.g., back to office after lunch, back home after work). On the other hand, the probability that a user stays at a given region is affected by specific days of a week, e.g., users are more likely to stay at the work regions on weekdays than weekends. Moreover, given a region, users may have different temporal patterns on different days (weekdays or weekends). For example, a user may visit her shopping regions in the afternoon of weekends, or evenings of weekdays. Finally, we investigate the social influence on users' check-in patterns. Users' social relationships have shown an impact on checking-in activities. Around 10% of the users' check-in records have overlapping locations with their friends (as shown in Figure 1 (f)).

3.2 Matrix Factorization based Recommendation

The two dimensional matrix factorization based collaborative filtering recommendation is the most widely used and successful recommendation model [21]. Let $\mathbf{u} = \{u_1, ..., u_m\}$ be the set of users, and $\mathbf{v} = \{v_1, ..., v_n\}$ be the set of locations, the basic idea is to interpret two dimensional user-location matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$ into two low rank matrices: users' preference matrix $\mathbf{U} \in \mathbb{R}^{m \times d}$, and location matrix $\mathbf{V} \in \mathbb{R}^{n \times d}$ with $d \ll \min(m, n)$ dimensional shared latent space. The probability of user u_i checking-in location v_j will be approximated by solving the following optimization problem:

$$\mathfrak{L}(\mathbf{U}, \mathbf{V}) = \min_{\mathbf{U}, \mathbf{V}} \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 \tag{1}$$

To avoid overfitting, two smoothing regularizations are imposed on **U** and **V** respectively. Thus, Equation 1 can be rewritten as:

$$\mathfrak{L}(\mathbf{U}, \mathbf{V}) = \min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \lambda_{\mathbf{U}} ||\mathbf{U}||_F^2 + \lambda_{\mathbf{V}} ||\mathbf{V}||_F^2 \tag{2}$$

where $\lambda_{\mathbf{U}}$ and $\lambda_{\mathbf{V}}$ are regularization parameters. The optimization process aims at minimizing the sum-of-squared-errors objective function with quadratic regularization terms and gradient based approaches can be applied to find a local minimum.

3.3 Tensor Factorization

In the standard matrix factorization, each piece of check-in information in the user-location matrix is determined by the inner product of user feature and location feature. Tensor Factorization (TF) models their time-evolving behavior by using the tensor notation. We can denote a piece of check-in information as $\mathcal{R}_{ijk}$ where i, j index users and locations as before, and k indexes the time slice when the check-in information is given. Then similar to the static case, we can organize the check-in information into a three-dimensional tensor $\mathcal{R} \in \mathbb{R}^{m \times n \times q}$, whose three dimensions correspond to user, location, and time slice with the sizes of m, n, and q, respectively.

Extending the idea of MF, we assume that each entry $\mathcal{R}_{ijk}$ can be expressed as the inner-product of three d-dimensional vectors:

$$\mathcal{R}_{ijk} \approx \mathbf{U}_i \mathbf{V}_j \mathbf{T}_k \tag{3}$$

The goal is to estimate matrices $\mathbf{U}_i$, $\mathbf{V}_j$, $\mathbf{T}_k$ subject to constraints. These include scaling to unit length vectors, nonnegativity, orthogonality, sparseness and/or smoothness of all or some of the columns. This leads to the following optimization problem:

$$\mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T}) = \frac{1}{2} \min_{\mathbf{U},\mathbf{V},\mathbf{T}} ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 \tag{4}$$

where $\hat{\mathcal{R}}$ denotes predicted approximation of $\mathcal{R}$. The optimization problem can be solved by various existing methods such as the stochastic gradient descendant methods.

In the above, we introduce the background of the matrix factorization based recommendation frameworks, which have several intriguing properties: (1) it is very flexible and can easily accommodate the additional information, e.g., spatial proximity or users' social similarity; (2) its optimization can be solved in linear time with simple gradient based methods.

4 THE PROPOSED METHOD

In this section, we systematically describe our tensor-based POI recommendation framework with social and spatial regularizations. Specifically, Section 4.1 introduces the check-in data model using a three-order tensor, Section 4.2 describes how to incorporate users' social similarity and locations' spatial proximity to regularize latent factors, and Section 4.3 gives the details of proposed model optimization process.

4.1 Tensor based Check-in Representation

A three-way tensor can be considered as a higher order generalization of two user-location dimensional matrices into a three-dimensional space. Therefore, the check-in data $\mathcal{R}$ is described by three dimensionalities according to user, location, and time other than the two-dimensionalities w.r.t. users and locations. In particular, the check-in frequency can be represented as points in the three-dimensional space, with the coordinates of each point corresponding to the index of the

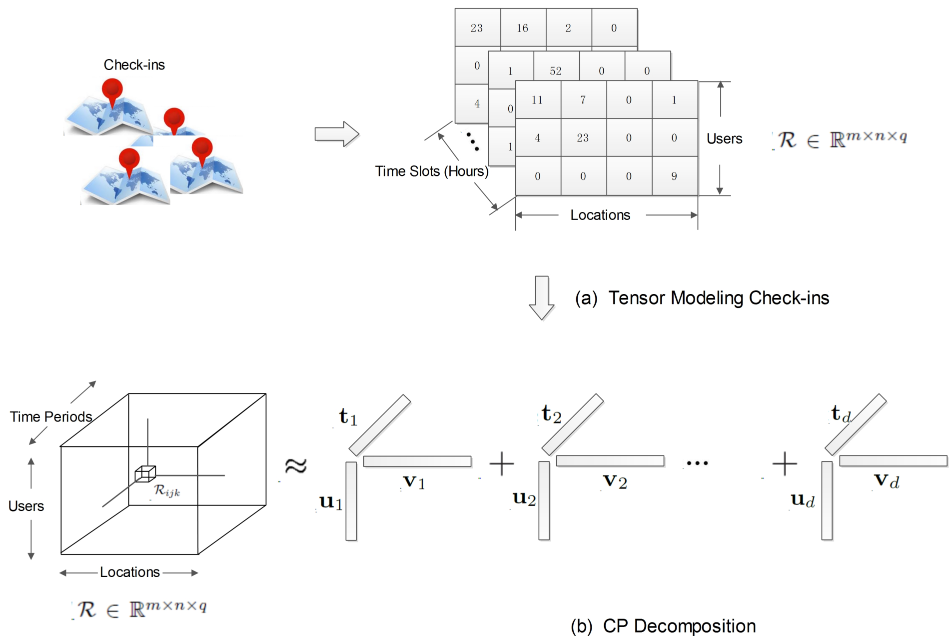


Fig. 2. Tensor-based check-in representation: (a) tensor modeling and (b) CANDECOMP/PARAFAC (CP) decomposition

triplet in terms of user, location and check-in time. More specifically, we model these relations as a tensor.

The tensor factorization framework of POI recommendation problem in this paper can be defined as follows. Given the historical check-in records of m users $\{u_i\}_{i=1}^m$ on n locations $\{v_j\}_{j=1}^n$ and q time frames $\{t_k\}_{k=1}^q$, recommending the target users a set of locations that they might be interested. We model the check-in matrix $\mathcal{R} \in \mathbb{R}^{m \times n \times q}$ via a third order tensor, where each $\mathcal{R}_{ijk}$ quantifies users' preference in terms of frequency, i.e., the times that user u_i visits location v_j within time period t_k. The tensor construction from check-in records $\mathcal{R} \in \mathbb{R}^{m \times n \times q}$ can be defined as $\mathcal{R}_{ijk} \subset \mathcal{R} \in \mathbb{R}^{m \times n \times q}$, indicating the number of check-ins made by user $\{u_i\}_{i=1}^m$ at location $\{v_j\}_{j=1}^n$ within time period $\{t_k\}_{k=1}^q$, as shown in Figure 2 (a).

The preference of user $\{u_i\}_{i=1}^m$ on location $\{v_j\}_{j=1}^n$ within time frame $\{t_k\}_{k=1}^q$ can be approximated via optimizing the following objective function by generalizing Equation 2 into the tensor factorization framework in Equation 4 where $\hat{\mathcal{R}}$ denotes the predicted approximation of $\mathcal{R}$, which can be estimated from check-in records using tensor decomposition techniques, such as the High Order Singular Value decomposition (HOSVD) and CANDECOMP/PARAFAC (CP) decomposition [10] as follows:

$$\hat{\mathcal{R}} \approx \sum_{d=1}^{D} \mathbf{u}_d \circ \mathbf{v}_d \circ \mathbf{t}_d \tag{5}$$

Therefore, Equation 1 can be rewritten as:

$$\mathfrak{L}(\mathbf{U},\mathbf{V},\mathbf{T}) = \frac{1}{2}\min_{\mathbf{U},\mathbf{V},\mathbf{T}} ||\mathcal{R} - \sum_{d}^{D} \mathbf{u}_d \circ \mathbf{v}_d \circ \mathbf{t}_d||_F^2 \tag{6}$$

where $\mathbf{u}_d \in \mathbb{R}^m$, $\mathbf{v}_d \in \mathbb{R}^n$ and $\mathbf{t}_d \in \mathbb{R}^q$, as shown in Figure 2 (b).

We aim at finding the decomposition $\hat{\mathcal{R}}$ that best approximates the original tensor $\mathcal{R}$ to achieve the best recommendation results. Similarly, to avoid overfitting, the regularization terms associated with $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{T}$ are introduced in Equation 6, which is reformulated as:

$$\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{T}) = ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \lambda(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2 + ||\mathbf{T}||_F^2) \tag{7}$$

where $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_D] \in \mathbb{R}^{m\times D}$, $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_D] \in \mathbb{R}^{n\times D}$, and $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_D] \in \mathbb{R}^{q\times D}$ are all factor matrices.

4.2 Integrating Contextual Regularizations

Even though we impose three regularizations, the low-rank assumption is insufficient by itself and we still need other assumptions to introduce more prior knowledge of entities of check-in activities. In particular, we exploit several inherited intriguing features of matrix factorization for such purposes.

In many cases, we have not only relational information among objects, but also internal information within the contextual entities themselves. Under this condition, the model in Equation 7 can be enhanced with relational information across different contextual entities like modeling as a tensor, but also information on the objects themselves, e.g., users' mutual friendships, and location proximity. Fusing the internal relations as regularization terms can effectively boost the recommendation accuracy, especially for the sparse cases [23]. We explain the two kinds of internal information used in our approach as follows:

- Friends tend to have similar behaviors because they might share common interests, thus leading to correlated check-in behaviors. For example, two friends may hang out to see a movie together sometimes, or a user may go to a restaurant highly recommended by his/her friends. All those possible reasons suggest that friends might provide good recommendations for a given user due to their potential correlated check-in behavior. In other words, we can turn to a user's friends for recommendation, which we call *recommendation based social influence from friends*. Users' explicit social friendships tend to improve the recommendation accuracy, which has been widely used in the two-dimensional matrix factorization based recommendation frameworks [32]. The social regularization term can constrain the matrix factorization objective function and indirectly propagate users' preferences.
- On the other hand, as mentioned earlier, the check-in activities of users record their physical interactions (i.e., visits) at POIs. To explore more on geographic information, we now build a model which depends only on location distance. The intuition behind the model is that: i) a user tends to visit locations near his/her home or office, and ii) a user may also favor several locations within a neighborhood. Thus, we assume that the majority locations a user checks in are within some certain distance.

According to the analysis above and in order to fully exploit the social and spatial information, in this paper, we further regularize the internal relations within contextual entities w.r.t. users and locations. To integrate these two relations, the basic idea is to make the latent representations of two users or two locations to be as close as possible if they have links and are similar enough. Thus, the objective function in Equation 7 can be reformulated by importing social influence regularization and spatial influence regularization:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{T}} \mathfrak{L} = ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \lambda_1 \underbrace{\sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{A}_{ij}||\mathbf{U}_{i*} - \mathbf{U}_{j*}||^2}_{\text{Social Regularization}} + \lambda_2 \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{B}_{ij}||\mathbf{V}_{i*} - \mathbf{V}_{j*}||^2}_{\text{Spatial Regularization}} + \lambda_3(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{T}||^2) \tag{8}$$

where $\mathbf{A}_{ij} \in \mathbb{R}^{m \times m}$ indicates the social strength between users u_i and u_j, and $\mathbf{B}_{ij}$ denotes the spatial proximity between locations v_i and v_j. In the following, we will introduce how to calculate user social strength and location spatial proximity. The first term $||\mathcal{R} - \hat{\mathcal{R}}||_F^2$ decomposes the check-in frequency w.r.t. each entity including user, location, and time tensor $\mathcal{R}$ as an outer-product of three dimensional representations. The second term poses a regularization term on user's mutual friendships and location similarity, forcing the low-dimensional representations of two users and two locations as close as possible if they are similar. The last term $(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{T}||^2)$ is used to avoid overfitting.

Social Similarity. We calculate the social strength between two users i and j based on both of their common social connections and similarity of check-in activities [18, 42].

$$\mathbf{A}_{ij} \propto s_{ij}^f = \eta_s \frac{|F_i \cap F_j|}{|F_i \cup F_j|} + (1 - \eta_s)\frac{|L_i \cap L_j|}{|L_i \cup L_j|} s.t. j \in \mathcal{N}_i \tag{9}$$

where $\mathcal{N}_i$ denotes user i's neighborhood, $F.$ denotes user's friendship set and $L.$ denotes the locations checked in by each user. λ is used to balance the importance of friend impact and impact of shared checked in locations.

Spatial Proximity. The spatial proximity between two locations is inversely proportional to the distance between two locations. Given a large number of locations, we propose a local neighborhood selection method to construct the location similarity matrix.

The locations v_i and v_j are considered as neighbors if i) v_i is among the k nearest neighbors of v_j or ii) v_j is among the k nearest neighbors of v_i. We use the inverse orthodromic distance to determine the weight between two locations. Orthodromic distance is the shortest distance between two points on the surface of a sphere since the earth is a sphere other than a plane. Thus, the distance between two locations $dis(v_i, v_j)$ is calculated as:

$$dis(v_i, v_j) = \mathrm{r} \times \Delta\hat{\sigma} \tag{10}$$

where $\Delta\hat{\sigma} = \arccos(\sin\phi_i \sin\phi_j + \cos\phi_i \cos\phi_j \cos(\Delta\lambda))$, ϕ_i and ϕ_j are the longitude and latitude of locations v_i and v_j, $\Delta(\lambda)$ is the absolute value between locations v_i and v_j, and r is the radius of the earth.

Therefore, the weight $\mathbf{B}_{ij} \in \mathbb{R}^{n \times n} \propto \frac{1}{dis(v_i, v_j)}$ between two locations v_i and v_j is nonzero only when $v_j \in \mathcal{N}_{v_i}$, where $\mathcal{N}_{u_i}$ denotes the local neighborhood of location v_i.

Up to this point, the proposed framework aims at optimizing the following objective function:

$$\begin{aligned}
\min_{\mathbf{U},\mathbf{V},\mathbf{T}} \mathfrak{L} &= ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \lambda_1 \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{A}_{ij}||\mathbf{U}_{i*} - \mathbf{U}_{j*}||^2 \\
&+ \lambda_2 \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{B}_{ij}||\mathbf{V}_{i*} - \mathbf{V}_{j*}||^2 + \lambda_3(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{T}||^2) \\
&= ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{U}^\top \mathbf{L_U}\mathbf{U}) + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{V}^\top \mathbf{L_V}\mathbf{V}) + \lambda_3(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2 + ||\mathbf{T}||_F^2) \\
&= ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \mathrm{tr}[\mathbf{U}^\top(\lambda_3\mathbf{I} + \lambda_1\mathbf{L_U})\mathbf{U}] + \mathrm{tr}[\mathbf{V}^\top(\lambda_3\mathbf{I} + \lambda_2\mathbf{L_V})\mathbf{V}] + \frac{\lambda_3}{2}\mathrm{tr}(\mathbf{T}\mathbf{T}^\top)
\end{aligned} \tag{11}$$

where α, β and λ_1, λ_2, λ_3 are the model parameters, $\mathbf{L_U}$ is the Laplacian matrix induced from users' social networks matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ and location proximity $\mathbf{B} \in \mathbb{R}^{n\times n}$, respectively. $\mathbf{L_U} = \mathbf{D_A} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal matrix whose i-th diagonal element is the sum of all the elements in the i-th row of $\mathbf{A}$, i.e., $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$.

4.3 Optimization

Our objective function is non-convex, and we can apply an alternative algorithm to find optimal solutions with computing gradients with respect to each factor within **U**, **V** and **T**, while fixing the others. The algorithm will keep updating the variables until convergence or reaching the maximum iterations. We follow a three-step alternating optimization strategy to solve this problem. Specifically, we alternately take derivatives for each factor within **U**, **V** and **T**, while fixing the other two until convergence to find the optimal solution.

Updating U. We optimize **U** with fixed **V** and **T**. Equation 11 can be rewritten with the mode-1 matricization, which is a special case of mode-n matricization to reorder the elements of a tensor into a matrix. The mode-1 matricization of tensor $\mathcal{R} \in \mathbb{R}^{m\times n\times q}$ is denoted by $\mathcal{R}_{(1)}$. More details can be found in [17]. At this point, we obtain:

$$\begin{aligned}
\mathfrak{L}(\mathbf{U},\mathbf{V},\mathbf{T}) = &\frac{1}{2}\Big(\mathcal{R}_{(1)} - \mathbf{U}(\mathbf{V}\odot\mathbf{T})^\top\Big)^\top\Big(\mathcal{R}_{(1)} - \mathbf{U}(\mathbf{V}\odot\mathbf{T})^\top\Big) \\
&+ \frac{\lambda_1}{2}tr(\mathbf{U}^\top\mathbf{L_U}\mathbf{U}) + \lambda_3||\mathbf{U}||_F^2
\end{aligned} \tag{12}$$

Taking the derivatives over the objective function of Equation 12, we can obtain:

$$\begin{aligned}
\nabla_{\mathbf{U}}\mathfrak{L}(\mathbf{U},\mathbf{V},\mathbf{T}) &= -\mathcal{R}_{(1)}(\mathbf{V}\odot\mathbf{T}) + \mathbf{U}[(\mathbf{V}^\top\mathbf{V}) * (\mathbf{T}^\top\mathbf{T})] + \lambda_1\mathbf{L_U}\mathbf{U} + \lambda_3\mathbf{U} \\
&= -\mathbf{U}(\mathbf{V}\odot\mathbf{T})^\top(\mathbf{V}\odot\mathbf{T}) + \mathbf{U}[(\mathbf{V}^\top\mathbf{V}) * (\mathbf{T}^\top\mathbf{T})] + \lambda_1\mathbf{L_U}\mathbf{U} + \lambda_3\mathbf{U}
\end{aligned} \tag{13}$$

Updating V. We minimize **V** with fixed **U** and **T**. The objective function in Equation 11 can be rewritten with the mode-1 matricization, we obtain:

$$\mathfrak{L}(\mathbf{U},\mathbf{V},\mathbf{T}) = \frac{1}{2}\Big(\mathcal{R}_{(2)} - \mathbf{V}(\mathbf{U}\odot\mathbf{T})^\top\Big)^\top\Big(\mathcal{R}_{(2)} - \mathbf{V}(\mathbf{U}\odot\mathbf{T})^\top\Big) + \frac{\lambda_2}{2}tr(\mathbf{V}^\top\mathbf{L_V}\mathbf{V}) + \lambda_3||\mathbf{V}||_F^2 \tag{14}$$

Taking the derivatives over the objective function of Equation 14, we can obtain:

$$\begin{aligned}
\nabla_{\mathbf{V}}\mathfrak{L}(\mathbf{U},\mathbf{V},\mathbf{T}) &= -\mathcal{R}_{(2)}(\mathbf{U}\odot\mathbf{T}) + \mathbf{V}[(\mathbf{V}^T\mathbf{V}) * (\mathbf{T}^\top\mathbf{T})] + \lambda_2\mathbf{L_V}\mathbf{V} + \lambda\mathbf{V} \\
&= -\mathbf{V}(\mathbf{V}\odot\mathbf{T})^\top(\mathbf{V}\odot\mathbf{T}) + \mathbf{V}[(\mathbf{V}^\top\mathbf{V}) * (\mathbf{T}^T\mathbf{T})] + \lambda_2\mathbf{L_V}\mathbf{V} + \lambda_3\mathbf{V}
\end{aligned} \tag{15}$$

ALGORITHM 1: Learning Algorithm of the Proposed Method

Input: User-Location-Time Tensor $\mathcal{R} \in \mathbb{R}^{m \times n \times q}$; User friendship matrix $\mathbf{A}$ and Location proximity matrix $\mathbf{B}$;
Output: Approximation tensor $\hat{\mathcal{R}}$, three factor matrices $\hat{\mathbf{U}} \in \mathbb{R}^{m \times D}$, $\hat{\mathbf{V}} \in \mathbb{R}^{n \times D}$ and $\hat{\mathbf{T}} \in \mathbb{R}^{q \times D}$
Initializing $\mathbf{U} \in \mathbb{R}^{m \times D}$, $\mathbf{V} \in \mathbb{R}^{n \times D}$ and $\mathbf{T} \in \mathbb{R}^{q \times D}$ with random values;
while *convergence or maximum steps* **do**
  Calculating $\nabla_{\mathbf{U}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$ (Equation 13);
  Calculating $\nabla_{\mathbf{V}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$ (Equation 15);
  Calculating $\nabla_{\mathbf{T}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$ (Equation 17);
  Updating $\hat{\mathbf{U}} \leftarrow \mathbf{U} - \delta \nabla_{\mathbf{U}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$;
  Updating $\hat{\mathbf{V}} \leftarrow \mathbf{V} - \delta \nabla_{\mathbf{V}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$;
  Updating $\hat{\mathbf{T}} \leftarrow \mathbf{T} - \delta \nabla_{\mathbf{T}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$;
end
$\hat{\mathcal{R}} \leftarrow \left[\left[\hat{\mathbf{U}}, \hat{\mathbf{V}}, \hat{\mathbf{T}}\right]\right]$

Updating T. We optimize $\mathbf{T}$ with fixed $\mathbf{U}$ and $\mathbf{V}$. The objective function in Equation 11 can be rewritten with the mode-1 matricization, we obtain:

$$\mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T}) = \left(\mathcal{R}_{(3)} - \mathbf{T}(\mathbf{U} \odot \mathbf{V})^\top\right)^\top \left(\mathcal{R}_{(3)} - \mathbf{T}(\mathbf{U} \odot \mathbf{V})^\top\right) + \lambda_3 ||\mathbf{T}||_F^2 \tag{16}$$

Taking the derivatives over the objective function of Equation 16, we can obtain:

$$\begin{aligned} \nabla_{\mathbf{T}} \mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T}) &= -\mathcal{R}_{(3)}(\mathbf{U} \odot \mathbf{V}) + \mathbf{T}[(\mathbf{U}^\top \mathbf{U}) * (\mathbf{V}^\top \mathbf{V})] + \lambda \mathbf{V} \\ &= -\mathbf{T}(\mathbf{U} \odot \mathbf{V})^\top (\mathbf{U} \odot \mathbf{V}) + \mathbf{V}[(\mathbf{V}^\top \mathbf{V}) * (\mathbf{T}^\top \mathbf{T})] + \lambda_3 \mathbf{T} \end{aligned} \tag{17}$$

where $\mathcal{R}_{(1)} \approx \mathbf{U}(\mathbf{V} \odot \mathbf{T})^\top \in \mathbb{R}^{m \times nq}$, $\mathcal{R}_{(2)} \approx \mathbf{V}(\mathbf{U} \odot \mathbf{T})^\top \in \mathbb{R}^{n \times mq}$ and $\mathcal{R}_{(3)} \approx \mathbf{T}(\mathbf{U} \odot \mathbf{T})^\top \in \mathbb{R}^{n \times mq}$.

Matrix factorization based methods can be calculated in linear time. One key issue of current optimization is to find an appropriate learning rate δ, which on the one hand should be big enough to have convergence after a reasonable number of iterations, and on the other hand be small enough so that the gradient steps are made towards the minimum which is especially important in the latter iteration stages. In this paper, we carry out an annealing procedure to discount δ by a factor of 0.9 after each iteration, as suggested by [19]. The algorithm loops over all the observations and updates the parameters by moving in the direction defined by the negative gradient. The algorithm will keep updating the variables until convergence or reaching the number of maximum iterations. The overall solution is shown in Algorithm 1.

5 EXPERIMENTS

In this section, we report the performance evaluation of our proposed method for location recommendation. In particular, we evaluated the following: i) how the proposed framework performs in comparison with the state-of-the-art models that capture contextual information (Section 5.2.1); ii) how the proposed framework conducts time-aware recommendation with various contextual regularization combinations, especially the tensor-based method without or partial social, spatial regularizations (Section 5.2.2); iii) how the different time granularities can affect the temporal recommendation performance (Section 5.3); and iv) how the tensor density influences prediction accuracy in the sense that the factor matrices dimensionality determines how many the latent factors have direct influence on the prediction accuracy (Section 5.4). Before we delve into experiment details, we first discuss the dataset used in the experiments and the evaluation metrics.

5.1 Experimental Settings

5.1.1 Dataset Preprocessing. We selected the check-in occurred during June 2010 to September 2010 from the original Brightkite dataset, and whole June check-ins from the Gowalla dataset [8]. Each check-in contains user, time and location ID information. We removed users who have checked in fewer than 10 locations and then removed POIs which have checked in by fewer than 10 users. As expected, after splitting a day into 24 slots by hours, the data became much sparser. The densities of the BrightKite and Gowalla datasets after splitting are 1.2×10^{-3} and 8.54×10^{-5}, respectively. The statistics of the preprocessed datasets are shown in Table 2. For each user, we randomly marked off 20% of locations as testing data to evaluate the effectiveness of the recommendation methods. Our model relies on unique and discrete locations, but the location information is given as continuous longitude/latitude coordinates in the original datasets. It can be observed from our datasets, that the check-in frequency commonly changes between a large range (e.g., over 500 times of a location). We used the mapping function $1/(1 + r^{-1})$ (r is the check-in frequency) to convert the check-in frequency into $[0, 1]$ [13]. Figure 3 (a) and (b) show the corresponding check-in distributions and the social links over the processed Brightkite dataset. It is noted that the social relations in Brightkite are asymmetric.

Table 2. Statistics of the Processed Datasets

	BrightKite	**Gowalla**
Duration	June 1, 2010 - Sept 30, 2010	June 1, 2010 - 30 June, 2010
# of Users	3,081	19,672
# of Locations	4,828	14,403
# of Check-ins	419,507	583,593
# of Friendship Links	31,924	15,568
Check-in Density	2.82×10^{-2}	2.06×10^{-3}
Check-in Density/Hours	1.2×10^{-3}	8.54×10^{-5}
User Social Link Density	3.4×10^{-3}	3.99×10^{-5}

5.1.2 Performance Metrics. To quantitively evaluate how well the proposed method can recover the POIs in the testing data for a given user at a given time, we used *Precision@x* and *Recall@x* to evaluate our proposed method (x= 5, 10, 15, 20). The precision@x measures how many previously marked off POIs are recommended to the users among the total number of recommended locations, and recall@x measures how many previously marked off POIs are recommended to the users among the total number of marked off locations. Following the definition in [45], the precision and recall for each time slot t are computed as:

$$precision@x(t) = \frac{\sum_{u_i \in \mathrm{U}} |Res(u_i) \cap Check(u_i)|}{\sum_{u_i \in \mathrm{U}} |Res(u_i) \cap \overline{Check(u_i)} + Res(u_i) \cap Check(u_i)|} \tag{18}$$

$$recall@x(t) = \frac{\sum_{u_i \in \mathrm{U}} |Res(u_i) \cap Check(u_i)|}{\sum_{u_i \in \mathrm{U}} |Res(u_i) \cap Check(u_i) + \overline{Res(u_i)} \cap Check(u_i)|} \tag{19}$$

where $Check(u_i)$ denotes the set of corresponding checked in groundtruth locations in the testing dataset for a given user u_i at time t, and $Res(u_i)$ denotes the set of recommendation locations by the proposed method for user u_i at time t.

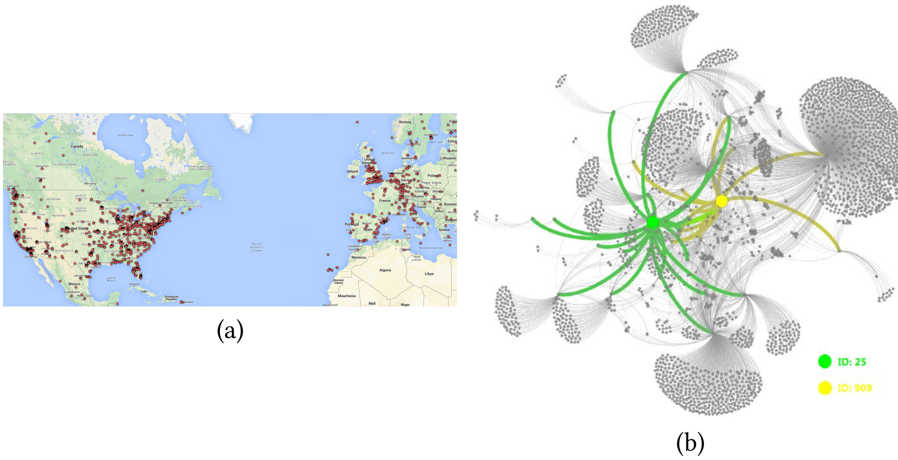


Fig. 3. (a) Check-in distribution over the processed dataset of BrightKite; (b) User friendships over the processed BrightKite dataset, each node represents a user. For example, the links of user id = 25 and id = 909 are highlighted in green and yellow respectively

The overall precision and recall will take the averaging value over all 24 time slots, i.e.,

$$precision@x = \frac{1}{24}\sum_{t=1}^{24} precision@x(t) \tag{20}$$

$$recall@x = \frac{1}{24}\sum_{t=1}^{24} recall@x(t) \tag{21}$$

5.1.3 Comparison Methods. We evaluated our proposed method, tensor factorization based location recommendation (**TenMF**), by comparing with 12 state-of-the-art methods, which were divided into two categories regarding whether the time factor is considered during prediction process for a given user. The first group is to evaluate the effectiveness of the proposed methods without a specific time frame. The second group is to evaluate the effectiveness of recommendation given a user in a specific time frame.

Group 1: Without utilizing temporal information

NMF. It only considers the 2-D user-location matrix. NMF applies non-negative matrix factorization on user-location matrix to predict the possibility of check-in. The user-location matrix can be decomposed into two lower dimension matrices in this method and contextual influence is not considered:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(r_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 \tag{22}$$

User-CF. It predicts a user's preferences by taking the preferences of other similar users into account and use *Jacaard similarity* for similarity computation.

Item-CF. It predicts a user's preferences on a target location by taking her preferences on similar locations into account [35] and *Jaccard similarity* for calculating the preference of similar users.

FA. It implements a friendship-aware (FA) recommendation. The probability of user i checks in location j can be calculated as:

$$\mathbf{P}_{ij}^{f} = \frac{\sum_{j \in F_i} s_{ij}^{f} \cdot c_{jk}}{\sum_{j \in F_i} s_{ij}} \tag{23}$$

where $c_{jk} = 1$ if user j checked in location k, otherwise 0. s_{ij}^{f} is computed as:

$$s_{ij}^{f} = \gamma \frac{|F_i \cap F_j|}{|F_i \cup F_j|} + (1-\gamma) \frac{|L_i \cap L_j|}{|L_i \cup L_j|} \tag{24}$$

where $F.$ denotes user's friendship set and $L.$ denotes the locations checked in by each user. λ is used to balance the importance of friend impact and the impact of shared checked in locations [43]. $\gamma = 0.4$ is the best setting based on our empirical study in this work.

GA. It implements the geographical-aware (GA) recommendation using Gaussian Mixture Model (GMM) to capture the geographical clustering influence, where the Gaussian center could be user's home, office, or entertainment places like a shopping mall or a restaurant [6]. The probability that user i visits a location k is modeled as below:

$$\begin{aligned} Pr(l_k|i) &= \sum_{m=1}^{M} q_{i,m} \mathcal{N}(l_k|\mu_{i,m}, \Sigma_{i,m}) \\ &= \sum_{m=1}^{M} \frac{q_{l,m}}{\sqrt{(2\pi)^{\mathcal{D}}|\Sigma_{i,m}|}} \exp(-\frac{1}{2}(x-\mu_{i,m})^T \Sigma_{i,m}^{-1}(k-\mu_{i,m})) \end{aligned} \tag{25}$$

where l_k denotes location k, which is represented by longitude and latitude coordinates, and m is the number of Gaussian clusters. $q_{i,m}$,$\mu_{i,m}$, and $\Sigma_{i,m}$ form the model parameter set $\mathbf{\Phi}_i$ at cluster l. q_m is the mixture weighted factor that describes the prior probability of the m^{th} mixture component. $\mu_{i,m}$ and $\Sigma_{i,m}$ are the mean and covariance of the m^{th} Gaussian distribution.

The traditional GMM learning process with Expectation Maximization (EM) limits to the manual determination of how many gaussian components (m) in the GMM. We adopt the Dirichlet Process Gaussian Mixture Model (DPGMM) [4] in observation probability distribution in this work. It uses the Dirichlet process as a prior over the distribution of the parameters and there is no need to explicitly declare the number of components. The approximate inference algorithm uses a truncated distribution with a fixed maximum number of components, but the number of components actually used almost always depends on the data.

Linear Model (LIM). We compare with a linear model (LIM) by integrating three partial contextual models together. The overall probability that user i would visit location k can be obtained using:

$$\mathbf{P}_{ik} = w_1 \mathbf{P}_{ik}^{t} + w_2 \mathbf{P}_{ik}^{f} + (1 - w_1 - w_2) \mathbf{P}_{ik}^{g} \tag{26}$$

where $w_1 = 0.1$ and $w_2 = 0.6$ are the best settings based on our empirical study.

Group 2: With temporal information

User-based Time (UBT). This model is the added time decay function based on original user-based CF model [9]. Given a user u_i and time frame t_k, the preference prediction is calculated as:

$$\hat{\mathbf{P}}_{ij}^{k} = \frac{\sum_{t_{k'}} sim_{u_i, t_{k'}} f(t_{k',j}, t_k)}{\sum_{k'} sim(u_i, t_{k'})} \tag{27}$$

where time function $f(t_{k',j}, t_k) = e^{|t_k - t_{k',v_j}|/H}$ is the time gap between time frame $t_{k'}$ checked in location v_j and the time of recommendation t_k. We set H=1 according to empirical study in [45].

Item-based Time (IBT). The same model with the time decay function based on item-based CF, similar to UBT.

Time-aware. To address the temporal influence in users' check-in behaviors, we decompose the time over two dimensions of day (Monday to Sunday) and hour (1 to 24). The probability of check-in is computed as:

$$\mathbf{P}^t_{ik} = \frac{\sum_{l=1}^{L} \mathbb{I}(i,l) \cdot sim(l,k) \cdot f(T)}{\sum_{l=1}^{L} sim(l,k) \cdot f(T)} \tag{28}$$

where l denotes a subset closely associated with user i according to her historical check-in records, $sim(l,k)$ can be computed using Jaccard similarity, and $f(T)$ is a temporal adjustment function for each user, which can be computed by using:

$$f(T) = \eta \cdot Pr(k|h) + (1-\eta) \cdot Pr(k|d) \tag{29}$$

where $Pr(k|h)$ is the probability of checking-in at location k, given the h-hour within a day (24 hours one day). $Pr(k|d)$ is the probability of checking-in at location k, given the d-th day within a week (7 days one week). All model parameters can be estimated by maximum likelihood estimation from the training dataset.

BasicTenMF. The special case of the proposed method without social and spatial regularizations in Equation 11.

$$\min_{\mathbf{U},\mathbf{V},\mathbf{T}} \mathfrak{L} = ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{T}||^2) \tag{30}$$

TenMF + Social. It is a special case of our proposed framework that only considers social regularization.

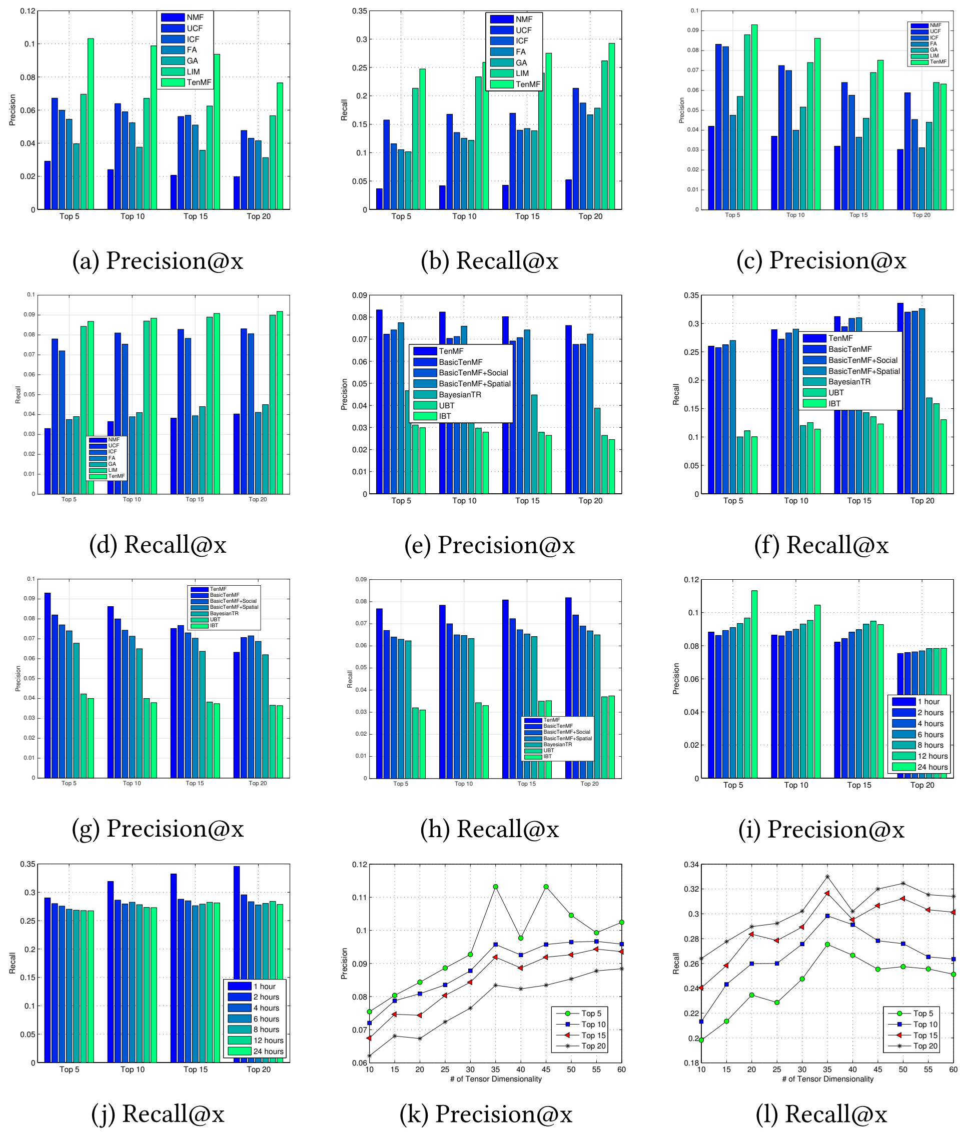
$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{T}} \mathfrak{L} = ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \alpha \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbf{A}_{ij}||\mathbf{U}_{i*} - \mathbf{U}_{j*}||^2 \\ + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{T}||^2) \end{aligned} \tag{31}$$

TenMF + Spatial. It is a special case of our proposed framework that only considers spatial regularization.

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{T}} \mathfrak{L} = ||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \beta \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{B}_{ij}||\mathbf{V}_{i*} - \mathbf{V}_{j*}||^2 \\ + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{T}||^2) \end{aligned} \tag{32}$$

5.2 Performance Comparison

We conducted two groups of experiments. The first group is used to evaluate the accuracy of the proposed recommendation methods with given users without considering specific time (i.e., two-dimensional recommendation). The second group is used to evaluate the recommendation accuracy by considering given users at given time frames (i.e., time-aware recommendation).



(a) Precision@x (b) Recall@x (c) Precision@x

(d) Recall@x (e) Precision@x (f) Recall@x

(g) Precision@x (h) Recall@x (i) Precision@x

(j) Recall@x (k) Precision@x (l) Recall@x

Fig. 4. Overall comparison *without* specific time frames, using the BrightKite dataset (a) and (b); and the Gowalla dataset (c) and (d) Overall comparison *with* given specific time frames over BrightKite dataset (e) and (f); Gowalla dataset (g) and (h); Impact of time granularity (i) and (j); Impact of tensor dimensionality (k) and (l)

5.2.1 Comparison without Temporal Information. It is noted that some of the methods can not be directly applied to context-aware prediction problem in a specific time frame [48]. So we employed a special formulation to enable the comparison. We considered the three-dimensional user-location-time tensor as a set of user-service matrix slices in terms of time intervals. Firstly, we compressed the tensor into a user-location matrix. Each element of this matrix is the average

Table 3. Comparison with other methods over Gowalla dataset

Methods	Precision@5	Precision@10	Recall@5	Recall@10
PMF [34]	0.041	0.027	0.014	0.013
BNMF [37]	0.057	0.022	0.014	0.012
POIFM [31]	0.042	0.048	0.023	0.014
Fu-POIFM [6]	0.094	0.081	0.044	0.029
Geo-BNMF [26]	0.073	0.062	0.032	0.023
Geo-PFM [28]	**0.113**	**0.098**	0.048	0.033
TenMF	**0.094**	**0.087**	**0.086**	**0.088**

of the specific user-location pairs during all the time intervals. For each slice of the tensor, the comparison methods were applied for the prediction. Secondly, we computed precision and recall of these baselines and made the comparison with our proposed method.

The overall comparison results using the two datasets are shown in Figure 4, from which we summarize two main observations. First, our tensor-based recommendation method significantly outperforms all the compared methods (including both non-context aware methods and context-aware methods) in terms of top 5 to top 20 validations. Our method obtains better prediction than the linear model, mostly because the tensor-based factorization can better reveal the hidden information. Second, all context-aware methods (e.g., the linear model fully combined with friendship, spatial and temporal influence) have better performance than the ones without or with only partial context-awareness, such as GA integrating spatial influence. It is interesting to note that user-based CF and item-based CF seem to be more effective than the basic matrix factorization method NMF, which has the worst accuracy. The reason may be that it only works on user-location matrix and does not integrate any contextual information. It also includes no regularizations for avoiding over-fitting.

We also compared our proposed TenMF model with several recent representative latent factorization based models, including Probabilistic Matrix Factorization (PMF) [34], Bayesian Non-negative Factorization (BNMF) [37], Poisson Factor Model (PFM) [31], Fused Poisson factor model (Fu-PoiFM) [6], Geo-BNMF [26], and Geo-PMF [28] using the Gowalla dataset. Table 3 shows the final results. From the table, we can find that our TenMF performs better than these state-of-the-art works except Geo-PFM [28]. Although Geo-PFM achieves better performance in terms of precision, our model produces better recall results.

To sum up, the results demonstrate the effectiveness of incorporating multi-dimensional contextual information in a unified tensor-based approach for improving the recommendation performance.

5.2.2 *Comparison with Temporal Information.* The second group is to evaluate the accuracy of our proposed recommendation method by utilizing specific time frames. We compared methods UBT, IBT, BayesianTA, TenMF+Social, BasicTenMF, TenMF+Spatial, TenMF. From the results shown in Figure 4 (e) - (h), we can draw several observations.

First, our proposed tensor-based method consistently outperforms all other comparison methods over the BrightKite dataset, even the data become more sparse after splitting by 24 hours. The results show that our proposed method is robust to deal with sparse check-in data and can make a better prediction. However, the performance is not quite consistent over the Gowalla dataset. For example, basic TenMF performs better than basic TenMF with social and TenMF with spatial information on recommending top 15 and 20 POIs though it underperforms when dealing with top 5 and top 10 recommendations. The reason lies in that the Gowalla dataset is more sparse than the BrightKite dataset in terms of check-in density and number of friendship links, while the social influence is not sufficient to calibrate the prediction results especially for a bigger size of recommendation list.

Second, BasicTenMF with spatial regularity is slightly better than BasicTenMF with social regularization, which indicates that spatial influence performs more dominant influence on recommendation results compared with social influence. Third, user-based time decay method and item-based time decay method play the worst prediction, and the reason lies in their weak capabilities of handling sparse datasets. The BayesianTR method has a stable performance.

Therefore, we can draw the general conclusion after the analysis of the results above, that our proposed method TenMF achieves the best results in most cases in terms of the evaluation metrics, within the settings including i) POI recommendation given the specific users and ii) POI recommendation given the specific users and time frames. The result demonstrates the advantage of modeling different context types in check-in data as compact tensors, as well as combining with internal relations within context types. The experimental result also shows the effectiveness of our proposed method.

5.3 Impact of Time Granularity

This experiment was designed to study the impact of different time granularity on POI recommendation accuracy. We varied the length of time frames as 1 hour, 2 hours, 4 hours, 8 hours, 12 hours and 24 hours. We performed the experiments using two datasets and achieved similar results. Due to space constraints, Figure 4 (i) and (j). shows the results from the BrightKite dataset. We can observe that the performance is generally getting better with longer length of time frames. The reason is that with a longer time frame, more discriminative information and more check-in data will be integrated and the data is getting denser, all of which can contribute to better accuracy. In this way, we can also observe that the effect of the time dimension is actually keeping decreasing with a wider time window. Thus, although the recall slightly decreases with larger time windows in some cases, the overall performance is getting improved. This finding is also consistent with the results observed in [45].

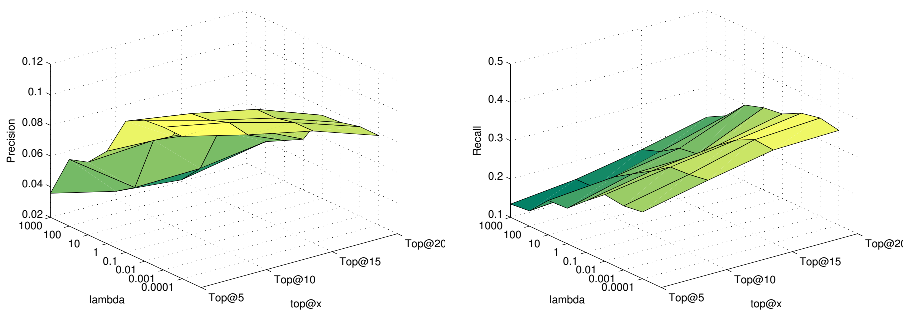
5.4 Tensor Dimensionality Sensitivity

As parameter dimensionality fundamentally determines the number of latent factors involved in the tensor factorization, in this section, we investigate the impact of this dimensionality by varying the value of dimensionality from 10 to 60 with a step size 5.

Figure 4 (k) and (l) shows the precision and recall at top 5, 10, 15 and 20 under different tensor dimensions using the BrightKite dataset. We observe that the precision and recall keep increasing with larger dimensions. However, they begin to slightly drop when dimensions reach around 35. The results reveal the fact that a larger dimensionality can effectively uncover information of check-ins and improve the recommendation performance. But when the dimensionality exceeds a certain threshold (35 in our case), the performance may degrade because of over-fitting. In addition, larger dimensionality also commits more computational cost.

5.5 Impact of Regularization Parameter λ

For tuning the regularization parameter λ_3, we varied its value from $\{10^{-4}, 10^{-3}, ..., 10^{3}\}$ with 10^x where $x = -4, -3, ..., 3$. Figure 5 shows the variations of precision and recall along with different λ. As demonstrated from the figure, precision increases until λ hits 10^{-3} and later decreases when λ has a bigger value. Recall exhibits the similar trend. The results demonstrate that the performance of the proposed model is sensitive to the regularization parameters. For example, in most cases, with bigger regularization, the precision@5 gets better and reaches the best when λ is around 0.1



(a) Precision@x (b) Recall@x

Fig. 5. Impact of regularization λ

to 1, and then gradually degrades when the value becomes bigger than 10. The recall follows the similar trend. We set $\lambda = 10^{-3}$ by default in this paper.

5.6 Discussion

In this section, we draw some discussions on the efficiency concerns of the proposed model. For the complexity of the current algorithm, at each step for deriving gradient, and at each iteration for calculating $\nabla_{\mathbf{U}}\mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$, the time complexity is $\mathcal{O}(mnq)$. Similarly, the time complexity of $\nabla_{\mathbf{V}}\mathfrak{L}(\mathbf{U}, \mathbf{V}, \mathbf{T})$ $w.r.t$ location $\mathbf{V}$ is $\mathcal{O}(mnq)$. Given that the number of iterations is K, the time complexity is $\mathcal{O}(K(mnq))$. We implement the optimization process using MATLAB Tensor Toolbox[2].

However, since this approach is not efficient enough, some recent efforts have proposed to use the *alternating direction method of multipliers* (ADMM) to efficiently solve the tensor problem [11]. It has been successful in solving large scale problems and optimization problems with multiple nonsmooth terms in the objective function, and has been widely used in many tensor-related applications. We develop an ADMM based optimization process to solve the objective function in Equation 11, which can be written by augmenting Lagrangian function as follows:

$$\begin{aligned} \mathcal{L} &= |||\mathcal{R} - \hat{\mathcal{R}}||_F^2 + \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{U}^\top \mathbf{L}_\mathbf{U}\mathbf{U}) + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{V}^\top \mathbf{L}_\mathbf{V}\mathbf{V}) + \lambda_3(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2 + ||\mathbf{T}||_F^2) \\ &\quad + < \mathbf{Y}_1, \mathbf{Z}_1 - \mathbf{U} > + < \mathbf{Y}_2, \mathbf{Z}_2 - \mathbf{V} > + \alpha(||\mathbf{Z}_1 - \mathbf{U}|| + ||\mathbf{Z}_2 - \mathbf{V}||) \\ &\quad s.t. \mathbf{Z}_1 = \mathbf{U}, \mathbf{Z}_2 = \mathbf{V} \end{aligned} \tag{33}$$

where $\mathbf{Y}$ is the matrix of Lagrange multipliers. We run some preliminary evaluation on the BrightKite dataset. The results, as shown in Table 4, indicate the consistent improvement achieved by the ADMM based optimization (TenMF(A)), comparing with TenMF based optimization.

6 CONCLUSION

In this paper, we have proposed a novel Point-of-Interest (POI) recommendation approach based on tensor factorization with users' social constraints and spatial influence as regularization terms. In particular, we model the check-in records of POI as a three-dimensional tensor and employ the tensor factorization method to enable effective POI recommendation in a higher dimensional space. We also propose to impose two relations within contextual entities as regularization terms of the tensor factorization to further improve the recommendation accuracy. Our proposed approach

[2]http://www.sandia.gov/ tgkolda/TensorToolbox/index-2.6.html

Table 4. Comparison with different optimization process

Metrics	Methods	@5	@10	@15	@20
Precision	**TenMF**	0.1032	0.0967	0.09325	0.07452
	TenMF(A)	0.1044	0.1003	0.09411	0.07724
Recall	**TenMF**	0.2489	0.2633	0.2779	0.2912
	TenMF(A)	0.2517	0.2639	0.2793	0.2996

achieves better performance than the state-of-the-art methods, which has been demonstrated from the results of our extensive experimental studies using two large real-life datasets.

Our future work will focus on two main directions. Firstly, the SGD (stochastic gradient descent) algorithm used in this paper suffers from high complexity and slow convergence. We plan to develop more efficient optimization algorithms to solve the objective function in Equation 11, e.g., using parallel factorization to accelerate the convergence process. We will also investigate a non-SGD optimization approach, e.g., reformulating the objective function in Equation 16 into $\mathrm{tr}(\mathbf{U}^\top * \mathbf{U}) - 2\mathrm{tr}(\mathbf{U}^\top * \mathbf{B})$, which would have a closed-form solution $\mathbf{U} = inv(\mathbf{A}) * \mathbf{B}$. Secondly, we will extend our TenMF model to integrate more temporal patterns, such as the dependency between adjacent time slots and the periodical relations of POI check-in behaviors. Finally, we plan to further validate our approach using more recent large-scale check-in datasets such as the Foursquare dataset [40] and compare our approach with some other existing POI recommendation methods as discussed in [29].

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

[2] Jie Bao, Yu Zheng, and Mohamed F Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 199–208.

[3] Betim Berjani and Thorsten Strufe. 2011. A Recommendation System for Spots in Location-based Online Social Networks. In *Proceedings of the 4th Workshop on Social Network Systems (SNS'11)*. ACM, 4.

[4] David M Blei, Michael I Jordan, et al. 2006. Variational inference for Dirichlet process mixtures. *Bayesian analysis* 1, 1 (2006), 121–143.

[5] Xuefeng Chen, Yifeng Zeng, Gao Cong, Shengchao Qin, Yanping Xiang, and Yuanshun Dai. 2015. On information coverage for location category based point-of-interest recommendation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[6] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence.*

[7] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2605–2611.

[8] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1082–1090.

[9] Yi Ding and Xue Li. 2005. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 485–492.

[10] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. 2011. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5, 2 (2011), 10.

[11] Silvia Gandy, Benjamin Recht, and Isao Yamada. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27, 2 (2011), 025010.

[12] Huiji Gao and Huan Liu. 2014. Data analysis on location-based social networks. In *Mobile Social Networking*. Springer, 165–194.

[13] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 93–100.

[14] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-Aware Point of Interest Recommendation on Location-Based Social Networks. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

[15] Longke Hu, Aixin Sun, and Yong Liu. 2014. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 345–354.

[16] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 79–86.

[17] Tamara Gibson Kolda. 2006. *Multilinear operators for higher-order decompositions.* United States. Department of Energy.

[18] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M Jose. 2009. On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 195–202.

[19] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[20] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[22] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* (2009), 30–37.

[23] Wu-Jun Li and Dit-Yan Yeung. 2009. Relation regularized matrix factorization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*.

[24] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 433–442.

[25] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 831–840.

[26] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1043–1051.

[27] Bin Liu and Hui Xiong. 2013. Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness. In *SDM*, Vol. 13. SIAM, 396–404.

[28] Bin Liu, Hui Xiong, Spiros Papadimitriou, Yanjie Fu, and Zijun Yao. 2015. A general geographical probabilistic factor model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2015), 1167–1179.

[29] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment* 10, 10 (2017), 1010–1021.

[30] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 739–748.

[31] Hao Ma, Chao Liu, Irwin King, and Michael R Lyu. 2011. Probabilistic factor models for web site recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 265–274.

[32] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 287–296.

[33] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. 2012. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery* 25, 2 (2012), 298–324.

[34] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *In Proc. of the 20th International Conference on Neural Information Processing Systems (NIPS 2007)*. Vancouver, Canada, 1257–1264.

[35] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. ACM, 285–295.

[36] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. 2011. Socio-Spatial Properties of Online Location-Based Social Networks. *ICWSM* 11 (2011), 329–336.

[37] Mikkel N Schmidt, Ole Winther, and Lars Kai Hansen. 2009. Bayesian non-negative matrix factorization. In *International Conference on Independent Component Analysis and Signal Separation*. Springer, 540–547.

[38] Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. 2015. Regularity and conformity: location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1275–1284.

[39] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G Schneider, and Jaime G Carbonell. 2010. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization.. In *SDM*, Vol. 10. SIAM, 211–222.

[40] Dingqi Yang, Daqing Zhang, Zhiyong Yu, Zhiwen Yu, and Djamal Zeghlache. 2014. SESAME: Mining User Digital Footprints for Fine-Grained Preference-Aware Social Media Search. *ACM Transactions on Internet Technology* 14, 4, Article 28 (2014), 28:1–28:24 pages.

[41] Lina Yao, Quan Z. Sheng, Yongrui Qin, Xianzhi Wang, Ali Shemshadi, and Qi He. 2015. Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[42] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 671–680.

[43] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 325–334.

[44] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. 2013. Lcars: A location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 221–229.

[45] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 363–372.

[46] Quan Yuan, Gao Cong, and Aixin Sun. 2014. Graph-based Point-of-interest Recommendation with Geographical and Temporal Influences. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 659–668.

[47] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: personalized geo-social location recommendation: a kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Informationfd Systems*. ACM, 334–343.

[48] Wancai Zhang, Hailong Sun, Xudong Liu, and Xiaohui Guo. 2014. Temporal qos-aware web service recommendation via non-negative tensor factorization. In *Proceedings of the 23rd international conference on World wide web*. International World Wide Web Conferences Steering Committee, 585–596.

[49] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*. ACM, 1029–1038.