# Exploring Temporal Effects for Location Recommendation on Location-Based Social Networks

Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu
Computer Science and Engineering, Arizona State University, Tempe, AZ, USA
{huiji.gao, jiliang.tang, xia.hu, huan.liu}@asu.edu

## ABSTRACT

Location-based social networks (LBSNs) have attracted an inordinate number of users and greatly enriched the urban experience in recent years. The availability of spatial, temporal and social information in online LBSNs offers an unprecedented opportunity to study various aspects of human behavior, and enable a variety of location-based services such as location recommendation. Previous work studied spatial and social influences on location recommendation in LBSNs. Due to the strong correlations between a user's check-in time and the corresponding check-in location, recommender systems designed for location recommendation inevitably need to consider temporal effects. In this paper, we introduce a novel location recommendation framework, based on the temporal properties of user movement observed from a real-world LBSN dataset. The experimental results exhibit the significance of temporal patterns in explaining user behavior, and demonstrate their power to improve location recommendation performance.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

## Keywords

Location Recommendation, Location-Based Social Networks, Temporal Effects

## 1. INTRODUCTION

The rapid growth of location-based social networks (LBSNs) has attracted billions of users, promoting our urban experience to a new stage [30]. Typical location-based social networking sites (e.g., Foursquare[1] and Facebook Places[2]) allow a user to "check in" at a location of interest with her smartphone, which informs her friends, along with creating the opportunity to make new friends or receive better recommendations. To benefit LBSN users and advance location-based marketing, location recommendation on LBSNs has become an essential task [24, 25], aiming to recommend *new* POIs (Points of Interest) to a user according to his personal preferences and facilitate his exploration of new areas of the city.

Location-based social networks present unprecedented large-scale check-in data to describe a user's mobile behavior in spatial, temporal, and social aspects. In previous work researchers explored users' personal static check-in preferences through geographical check-ins for location recommendation [28, 1, 29, 26]. Inspired by social influence theories that social friends tend to have similar check-in behavior, researches started to investigate the explicit social friendships on LBSNs [5, 9, 8] and leverage their power for improving location recommendation services [23, 3, 25]. Among existing work, the temporal patterns of a user's check-in actions have not been explored in depth.

As suggested in [22, 4, 16], human geographical movement exhibits significant temporal patterns on LBSNs and is highly relevant to the location property, while the daily pattern (hours of the day) is one of the most fundamental patterns that reflects a user's mobile behavior. For example, a user may regularly arrive to the office around 9:00 am, go to a restaurant for lunch at 12:00 pm, and watch movies at night around 10:00 pm. Therefore, investigating the features embedded in daily patterns enables us to better understand human mobile behavior, providing a potential opportunity to design more advanced location recommender systems on LBSNs.

Previous work reports that a user's preferences change continuously over time [21, 10], indicating two temporal properties of a user's daily check-in preferences: (1) **non-uniformness**: a user exhibits distinct check-in preferences at different hours of the day; and (2) **consecutiveness**: a user tends to have more similar check-in preferences in consecutive hours than in non-consecutive hours. In Section 5 we validate these properties experimentally on a real-world dataset. Figure 1 plots an illustrative example of a user's aggregated check-in activities on his top 5 most visited locations over 24 hours on a real-world LBSN. Each cell represents the total number of check-in activities happened at a specific location during the corresponding hour, colored from black (least active) to white (most active). The user's check-in behavior presents a different check-in location distribution at each hour, which changes continually over time.

The non-uniformness and consecutiveness properties suggest strong correlations between a user's check-in time and the corresponding check-in preferences. However, these properties have not been exploited for location recommendation on LBSNs. In this paper, we aim to leverage them for location recommendation. To the best of our knowledge, this is the first work of modeling temporal ef-

[1]http://foursquare.com/
[2]http://www.facebook.com/about/location/


*RecSys'13,* October 12–16, 2013, Hong Kong, China.

http://dx.doi.org/10.1145/2507157.2507182.

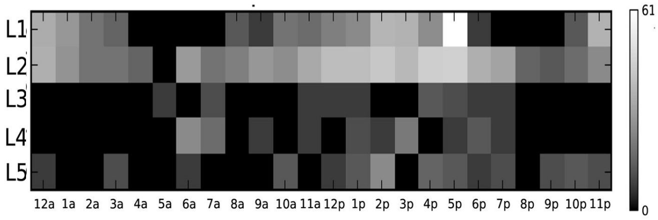



**Figure 1: Daily Check-in Activities on LBSN**

fects on location-based social networks for location recommendation. The contributions of this paper are summarized below:

- We propose a location recommendation framework with temporal effects based on observed temporal properties.
- We introduce four temporal aggregation strategies to integrate a user's check-in preferences of different temporal states.
- We evaluate the temporal effects for location recommendation; the results exhibit its good recommendation performance, and demonstrate the advantage of considering time-dependent check-in preferences over static check-in preferences.

The remainder of this paper is organized as follows. We first give a brief review of some related work in Section 2, then introduce a basic location recommendation model and discuss the modeling of temporal properties in Section 3. Next we introduce our location recommendation framework **LRT** with temporal effects in terms of temporal regularization and temporal aggregation in Section 4, followed by the discussion of experimental design and results on a real-world LBSN dataset in Section 5. We conclude this work with future work in Section 6.

# 2. RELATED WORK

The properties of location-based social networks has been widely studied w.r.t. the geographical and social aspects. Cheng et al. [4] investigated the "Lèvy Flight" property of human check-in patterns, and discovered that social status is affected by geographic constraints. In [19, 18], the authors investigated the spatial properties of the social networks on main popular LBSNs. They observed strong heterogeneity across users with different characteristic geographic scales of interactions across social ties. Efforts have also been made to utilize the geo and social properties for improving location-based social services. Gao et al. [9] studied geo-social correlations on LBSNs to solve the "cold start" location prediction problem. Zhang et al. [27] proposed a unified influence metric to evaluate the geo-social influence among users in LBSNs.

Ye et al. [24] introduced location recommendation into location-based social networks. Specifically, the authors investigated the geographical influence [25] and social influence [23] for location recommendation, and discovered that user preference plays a more important role in contributing to the recommendation than social and geographical influence. Berjani et al. [1] proposed a location recommendation system utilizing matrix factorization methods. Cheng [3] investigated the geographical influence through a multi-center Gaussian model, together with matrix factorization and social influence for location recommendation. Zhou et al. [29] studied the location recommendation problem on location-based social networks with various collaborative filtering approaches. Ying et al. [26] proposed a set of features related to social factor, individual preference, and location popularity, and utilized a regression-tree model to recommend POIs. Most recently, Yang et al. [7] introduced sentiment information into location recommendation system, and reported its good performance over state-of-the-art approaches.

Among the current work in LBSNs, temporal information has not been explored for location recommendation. Temporal information has been studied for other location-based social services such as location prediction and location classification. Liu et al. [14] analyzed the daily temporal distribution of check-ins and leveraged them to infer the types of locations. Cho et al. [5] proposed a Periodic & Social Mobility Model for location prediction with two temporal states ("home" and "work") affected by social effects and non-social effects. Chang et al. [2] proposed a logistic regression model with observed temporal patterns as one type of feature, and found that hourly patterns have a small but significant effect, while weekly patterns are not predictive.

# 3. MODELING TEMPORAL EFFECTS FOR LOCATION RECOMMENDATION

The large-scale check-in data on LBSNs is usually very sparse due to the user-driven check-in property [19, 17, 8]. To solve large-scale recommendation problems, matrix factorization is state-of-the-art technology that has been proven to be successful in the Netflix Competition [11, 12], and is being used for item recommendation and trust prediction on product review sites like Epinions and Ciao for research purposes [15, 20, 21]. Therefore, in this paper, we leverage the temporal properties on LBSNs with low-rank matrix factorization for location recommendation.

## 3.1 Location Recommendation without Temporal Effects

We first introduce a basic location recommendation model based on low-rank matrix factorization without considering temporal effects. Let $\mathbf{u} = \{u_1, u_2, ..., u_m\}$ be the set of users, and $\mathbf{l} = \{l_1, l_2, ..., l_n\}$ be the set of locations, where m and n denote the number of users and locations, respectively. $\mathbf{C} \in \mathbb{R}^{m\times n}$ is a user-location matrix with each element $\mathbf{C}_{ij}$ representing the number of check-ins made by user $u_i$ at location $l_j$. Let $\mathbf{U} \in \mathbb{R}^{m\times d}$ be the user check-in preferences and $\mathbf{L} \in \mathbb{R}^{n\times d}$ be the location characteristics, with $d \ll min(m, n)$ being the number of latent preference factors. The basic location recommendation model approximates $u_i$'s check-in preference on an unvisited $l_j$ via solving the following optimization problem:

$$\min_{\mathbf{U}_i\geq 0, \mathbf{L}_j\geq 0} \sum_i^m \sum_j^n \mathbf{Y}_{ij}(\mathbf{C}_{ij} - \mathbf{U}_i\mathbf{L}_j^\top)^2, \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{m\times n}$ is a check-in indicator matrix, $\mathbf{Y}_{ij} = 1$ indicating that $u_i$ has checked in at $l_j$, $\mathbf{Y}_{ij} = 0$ otherwise.

After obtaining $\mathbf{U}_i$ and $\mathbf{L}_j$, the missing value in $\mathbf{C}$, represented as $\widetilde{\mathbf{C}}_{ij}$, indicating the preference of a user $u_i$ at an unvisited location $l_j$, is then approximated by $\mathbf{U}_i\mathbf{L}_j^\top$. To avoid over-fitting, two smoothness regularizations are added on $\mathbf{U}_i$ and $\mathbf{L}_j$ respectively. Eq.(1) can then be represented in matrix format as

$$\min_{\mathbf{U}\geq 0, \mathbf{L}\geq 0} \|\mathbf{Y} \odot (\mathbf{C} - \mathbf{U}\mathbf{L}^\top)\|_F^2 + \alpha\|\mathbf{U}\|_F^2 + \beta\|\mathbf{L}\|_F^2, \tag{2}$$

where $\alpha$ and $\beta$ are non-negative parameters to control the capability of $\mathbf{U}$ and $\mathbf{L}$. $\odot$ is the Hadamard product operator, where $(A \odot B)_{i,j} = A_{i,j} \times B_{i,j}$. $\|\cdot\|_F$ is the Frobenius norm of a matrix.

## 3.2 Modeling Temporal Non-Uniformness for Location Recommendation

According to the temporal property of non-uniformness as described above, users exhibit distinct check-in preferences at different hours of the day. This inspires us to consider a user's check-in behavior as a set of time-dependent check-in preferences, with each preference corresponding to an hour of the day. To model this property, we first introduce temporal state $t \in [1, T]$ to represent the hour of the day, where $T = 24$ is the total number of temporal states. For example, $t = 1$ for check-in time at "2012-10-24 00:30:00pm", indicating the check-in happens during hour 0 to 1.

We then define $\mathbf{U}_t \in \mathbb{R}^{m\times d}$ as the time-dependent user check-in preferences under temporal state $t$. As observed in [22], location characteristics are inherent properties that do not change much as time goes by. Therefore, we define location characteristics to be time-independent, denoted as $\mathbf{L} \in \mathbb{R}^{n\times d}$. By approximating the check-in activities at each temporal state $t$ and minimizing their aggregation, we obtain time-dependent user check-in preferences via the following optimization problem:

$$\min_{\mathbf{U}_t\geq 0, \mathbf{L}\geq 0} \sum_{t=1}^{T} \|\mathbf{Y}_t \odot (\mathbf{C}_t - \mathbf{U}_t\mathbf{L}^\top)\|_F^2 + \alpha \sum_{t=1}^{T} \|\mathbf{U}_t\|_F^2 + \beta\|\mathbf{L}\|_F^2, \quad (3)$$

where $\mathbf{C}_t \in \mathbb{R}^{m\times n}$ contains the check-in activities at temporal state $t$, and $\mathbf{Y}_t$ is the corresponding indicator matrix.

## 3.3 Modeling Temporal Consecutiveness with Temporal Regularization

Inspired by the temporal consecutiveness property, which implies that users on LBSNs tend to have closer check-in preferences on consecutive temporal state, we propose a temporal regularization to minimize the following terms:

$$\min \sum_{t=1}^{T} \sum_{i=1}^{m} \psi_i(t, t-1)\|\mathbf{U}_t(i,:) - \mathbf{U}_{t-1}(i,:)\|_2^2, \quad (4)$$

where $\psi_i(t, t-1) \in [0, 1]$ is defined as a temporal coefficient that measures the temporal closeness of $u_i$'s check-in preferences between temporal state $t$ and $t-1$. The larger $\psi_i(t, t-1)$ is, the closer $u_i$'s check-in preferences between $t$ and $t-1$. We use cosine similarity to measure $\psi_i(t, t-1)$, defined as

$$\psi_i(t, t-1) = \frac{\mathbf{C}_t(i,:) \cdot \mathbf{C}_{t-1}(i,:)}{\sqrt{\sum_j \mathbf{C}_t^2(i,j)}\sqrt{\sum_j \mathbf{C}_{t-1}^2(i,j)}}. \quad (5)$$

Note that we consider the temporal state $t-1$ as $T$ when $t = 1$, e.g., $U_{t-1} = U_T$ when $t = 1$. After some derivations, we can get the matrix form of temporal regularization,

$$\begin{aligned}
&\sum_{t=1}^{T} \sum_{i=1}^{m} \psi_i(t, t-1)\|\mathbf{U}_t(i,:) - \mathbf{U}_{t-1}(i,:)\|_2^2 \\
&= \sum_{t=1}^{T} \sum_{i=1}^{m} \psi_i(t, t-1)(\mathbf{U}_t - \mathbf{U}_{t-1})^\top(i,:)(\mathbf{U}_t - \mathbf{U}_{t-1})(i,:) \\
&= \sum_{t=1}^{T} \sum_{i=1}^{m} (\mathbf{U}_t - \mathbf{U}_{t-1})^\top(i,:)\psi_i(t, t-1)(\mathbf{U}_t - \mathbf{U}_{t-1})(i,:) \\
&= \sum_{t=1}^{T} Tr((\mathbf{U}_t - \mathbf{U}_{t-1})^\top\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1})), \quad (6)
\end{aligned}$$

where $\Sigma_t$ is the diagonal temporal coefficient matrix among $m$ users, defined as

$$\mathbf{\Sigma}_t = \begin{bmatrix} \psi_1(t, t-1) & 0 & \cdots & 0 \\ 0 & \psi_2(t, t-1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_m(t, t-1) \end{bmatrix}. \quad (7)$$

# 4. LRT: LOCATION RECOMMENDATION FRAMEWORK WITH TEMPORAL EFFECTS

In this section, we formally introduce our location recommendation framework **LRT**. Figure 2 illustrates the working flow of our location recommendation framework. "$x$" denotes the observed check-in frequency by the user on the corresponding location, and "?" represents the user's check-in preferences on an unvisited location that the framework is going to infer. The whole framework consists of three steps: temporal division, temporal factorization, and temporal aggregation. Firstly, the original user-location matrix $\mathbf{C}$ is divided into $T$ sub-matrices according to the $T$ temporal states, with each sub-matrix only containing check-in actions that happened at the corresponding temporal state. Secondly, each $\mathbf{C}_t$ is factorized into the user check-in preference $\mathbf{U}_t$ and the location characteristics $\mathbf{L}$ based on the model presented in Section 3, while $\mathbf{L}$ is shared by all of $\mathbf{U}_t$. Finally, the corresponding low-rank approximation $\widetilde{\mathbf{C}}_t$ is constructed and aggregated into $\widetilde{\mathbf{C}}$, representing the user check-in preferences of each location.

Since the temporal division is straightforward to implement, in the following, we will describe in details the second and third steps, i.e., learning user temporal check-in preferences at each temporal state and aggregating temporal check-in preferences for location recommendation.

## 4.1 Learning Temporal Check-in Preferences

Based on the discussion of modeling temporal non-uniformness and consecutiveness properties in the above sections, the user temporal check-in preferences can be obtained by solving the following optimization problem:

$$\begin{aligned}
\min_{\mathbf{U}_t\geq 0, \mathbf{L}\geq 0} &\sum_{t=1}^{T} \|\mathbf{Y}_t \odot (\mathbf{C}_t - \mathbf{U}_t\mathbf{L}^\top)\|_F^2 + \alpha \sum_{t=1}^{T} \|\mathbf{U}_t\|_F^2 + \beta\|\mathbf{L}\|_F^2 \\
&+ \lambda \sum_{t=1}^{T} Tr((\mathbf{U}_t - \mathbf{U}_{t-1})^\top\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1})), \quad (8)
\end{aligned}$$

where $\lambda$ is a non-negative parameter to control the temporal regularization. The corresponding objective function $\mathcal{J}$ is

$$\begin{aligned}
\mathcal{J} = &\sum_{t=1}^{T} Tr((\mathbf{Y}_t^\top \odot \mathbf{C}_t^\top)(\mathbf{Y}_t \odot \mathbf{C}_t) - (\mathbf{Y}_t^\top \odot \mathbf{C}_t^\top)(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top) \\
&- (\mathbf{Y}_t \odot \mathbf{C}_t)(\mathbf{Y}_t^\top \odot \mathbf{L}\mathbf{U}_t^\top) + (\mathbf{Y}_t^\top \odot \mathbf{L}\mathbf{U}_t^\top)(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)) \\
&+ \lambda \sum_{t=1}^{T} Tr((\mathbf{U}_t - \mathbf{U}_{t-1})^\top\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1})) \\
&+ \alpha \sum_{t=1}^{T} Tr(\mathbf{U}_t^\top\mathbf{U}_t) + \beta Tr(\mathbf{L}^\top\mathbf{L}) \\
&- \sum_{t=1}^{T} Tr(\Gamma_{\mathbf{U}_t}\mathbf{U}_t^\top) - Tr(\Gamma_{\mathbf{L}}\mathbf{L}^\top), \quad (9)
\end{aligned}$$

where $\Gamma_{\mathbf{U}_t}$ and $\Gamma_{\mathbf{L}}$ are Lagrangian multipliers for non-negativity of $\mathbf{U}_t$ and $\mathbf{L}$, respectively. By taking the derivation of $\mathcal{J}$ with respect to $\mathbf{U}_t$ and $\mathbf{L}$, we obtain

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{U}_t} &= -2(\mathbf{Y}_t \odot \mathbf{C}_t)\mathbf{L} + 2(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + 2\lambda\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1}) \\ &\quad + 2\alpha\mathbf{U}_t - \Gamma_{\mathbf{U}_t}, \\ \frac{\partial \mathcal{J}}{\partial \mathbf{L}} &= -2\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t + 2\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t \\ &\quad + 2\beta\mathbf{L} - \Gamma_{\mathbf{L}}. \end{aligned} \tag{10}$$

Let $\frac{\partial \mathcal{J}}{\partial \mathbf{U}_t} = 0$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{L}} = 0$, we obtain

$$\begin{aligned}\Gamma_{\mathbf{U}_t} &= -2(\mathbf{Y}_t \odot \mathbf{C}_t)\mathbf{L} + 2(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + 2\lambda\Sigma_t(\mathbf{U}_t - \mathbf{U}_{t-1}) \\ &\quad + 2\alpha_t\mathbf{U}_t, \\ \Gamma_{\mathbf{V}} &= -2\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t + 2\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t + 2\beta\mathbf{V}. \end{aligned} \tag{11}$$

According to the Karush-Kuhn-Tucker condition,

$$\begin{aligned}&\mathbf{U}_t(i,k)\Gamma_{\mathbf{U}_t}(i,k) = 0, \forall i \in [1,m], k \in [1,d], t \in [1,T] \\ &\mathbf{L}(i,k)\Gamma_{\mathbf{L}}(i,k) = 0, \forall i \in [1,n], k \in [1,d]. \end{aligned} \tag{12}$$

We obtain the following updating formula of $\mathbf{U}_t$ and $\mathbf{L}$ with a similar derivation process in [6]

$$\begin{aligned}\mathbf{U}_t(i,k) &\leftarrow \mathbf{U}_t(i,k)\sqrt{\frac{[(\mathbf{Y}_t \odot \mathbf{C}_t)\mathbf{L} + \lambda\Sigma_t\mathbf{U}_{t-1}](i,k)}{[(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + \lambda\Sigma_t\mathbf{U}_t + \alpha\mathbf{U}_t](i,k)}} \\ \mathbf{L}(i,k) &\leftarrow \mathbf{L}(i,k)\sqrt{\frac{[\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t](i,k)}{[\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t + \beta\mathbf{L}](i,k)}}. \end{aligned} \tag{13}$$

## 4.2 Temporal Aggregation for Location Recommendation

By solving the above optimization problem, the user check-in preferences $\widetilde{\mathbf{C}}_t(i,j)$ at each temporal state can be computed through $\mathbf{U}_t(i,:)\mathbf{L}(j,:)^\top$. To recommend locations to a user w.r.t. each $\widetilde{\mathbf{C}}_t(i,j)$, we define an aggregation function $f(\cdot)$ to compute the final user check-in preferences $\widetilde{\mathbf{C}}(i,j)$.

$$\widetilde{\mathbf{C}}(i,j) = f(\widetilde{\mathbf{C}}_1(i,j), \widetilde{\mathbf{C}}_2(i,j), ..., \widetilde{\mathbf{C}}_T(i,j)). \tag{14}$$

In this paper, we propose four aggregation strategies for $f(\cdot)$, defined below:

- **Sum**: we consider a user's check-in preferences at a location as the sum of his check-in preferences from each temporal state, i.e., $\widetilde{\mathbf{C}}(i,j) = \sum_{t=1}^{T}\widetilde{\mathbf{C}}_t(i,j)$.
- **Mean**: we consider a user's check-in preferences at a location as the average non-zero preferences from each temporal state, i.e., $\widetilde{\mathbf{C}}(i,j) = \frac{\sum_{t=1}^{T}\widetilde{\mathbf{C}}_t(i,j)}{|\{\widetilde{\mathbf{C}}_t(i,j)|\widetilde{\mathbf{C}}_t(i,j)\neq 0\}|}$.
- **Maximum**: we consider a user's check-in preferences at a location as his maximum temporal check-in preferences, i.e., $\widetilde{\mathbf{C}}(i,j) = max(\widetilde{\mathbf{C}}_1(i,j), ..., \widetilde{\mathbf{C}}_T(i,j))$.
- **Voting**: Each $\widetilde{\mathbf{C}}_t(i,j)$ acts as a recommender, and nominates top n locations to a user. The final recommended locations are those locations that have been nominated by most $\widetilde{\mathbf{C}}_t(i,j)$.

The location recommendation will then be performed based on the final user check-in preference $\widetilde{\mathbf{C}}(i,j)$.

## 4.3 Algorithm Analysis and Time Complexity

Algorithm 1 presents the detailed procedures of the proposed framework. Compared to the temporal division of $C$ and temporal aggregation of $\widetilde{\mathbf{C}}$, the updating rules for $\mathbf{U}_t$ and $L$ in each iteration corresponds to the major cost of Algorithm 1. Therefore, we next analyze the time complexity of updating $\mathbf{U}_t$ and $L$. For the updating rule of $\mathbf{U}_t$, $(\mathbf{Y}_t \odot \mathbf{C}_t)\mathbf{L}$ takes $O(md^2)$ operations due to the sparsity of $\mathbf{Y}_t$ and $\mathbf{C}_t$. Since $\Sigma_t$ is a diagonal matrix, the time complexity of $\lambda\Sigma_t\mathbf{U}_{t-1}$ is $O(md)$. $(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L}$ takes $O(mnd)$ operations, while the time complexity of $\lambda\Sigma_t\mathbf{U}_t$ and $\alpha\mathbf{U}_t$ is $O(md)$. Therefore, it takes $O(mndT)$ operations to update all of $\mathbf{U}_t$. Similarly, the time complexity of $\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t$ for updating $\mathbf{L}$ is $O(md^2T)$. $\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t$ takes $O(mndT)$ operations and $\beta\mathbf{L}$ takes $O(nd)$ operations, resulting in the time complexity of updating $\mathbf{L}$ as $O(mndT)$. Since T is usually a constant of small value, in sum, the time complexity of Algorithm 1 is $O(mnd)$.

**Algorithm 1** Location Recommendation with Temporal Effects

**Input:** user-location check-in matrix $\mathbf{C}$, $\alpha$, $\beta$, possible temporal states $\{1, 2, ..., T\}$
**Output:** approximated user-location preference matrix $\widetilde{\mathbf{C}}$

Divide $\mathbf{C}$ into $\{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_T\}$ according to $T$
Generate $\{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_T\}$ based on $\{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_T\}$
Construct $\{\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, ..., \mathbf{\Sigma}_T\}$ based on $\{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_T\}$
Initialize $\{\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_T\}$ and $\mathbf{L}$ randomly
**while** Not Convergent **do**
  **for** $t = 1$ to $T$ **do**
    **for** $i = 1$ to $m$ **do**
      **for** $k = 1$ to $d$ **do**
        $\mathbf{U}_t(i,k) \leftarrow \mathbf{U}_t(i,k)\sqrt{\frac{[(\mathbf{Y}_t \odot \mathbf{C}_t)\mathbf{L} + \lambda\Sigma_t\mathbf{U}_{t-1}](i,k)}{[(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)\mathbf{L} + \lambda\Sigma_t\mathbf{U}_t + \alpha\mathbf{U}_t](i,k)}}$
      **end for**
    **end for**
  **end for**
  **for** $i = 1$ to $n$ **do**
    **for** $k = 1$ to $d$ **do**
      $\mathbf{L}(i,k) \leftarrow \mathbf{L}(i,k)\sqrt{\frac{[\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{C}_t)^\top\mathbf{U}_t](i,k)}{[\sum_{t=1}^{T}(\mathbf{Y}_t \odot \mathbf{U}_t\mathbf{L}^\top)^\top\mathbf{U}_t + \beta\mathbf{L}](i,k)}}$
    **end for**
  **end for**
**end while**
**for** $t = 1$ to $T$ **do**
  Set $\widetilde{\mathbf{C}}_t = \mathbf{U}_t\mathbf{L}^\top$
**end for**
**for** $i = 1$ to $m$ **do**
  **for** $j = 1$ to $n$ **do**
    Set $\widetilde{\mathbf{C}}(i,j) = f(\widetilde{\mathbf{C}}_1(i,j), \widetilde{\mathbf{C}}_2(i,j), ..., \widetilde{\mathbf{C}}_T(i,j))$
  **end for**
**end for**
**return** $\widetilde{\mathbf{C}}$

# 5. EXPERIMENTS

In this section, we evaluate the performance of our framework **LRT** for location recommendation. In particular, we evaluate the following: (1) how the proposed framework fares in comparison with state-of-the-art models that capture static check-in preferences; (2) how the proposed framework recommends locations with various temporal aggregation strategies; and (3) whether other temporal patterns could be leveraged for location recommendation with the proposed framework. Before we delve into experiment details, we first discuss an LBSN dataset and evaluation metrics.

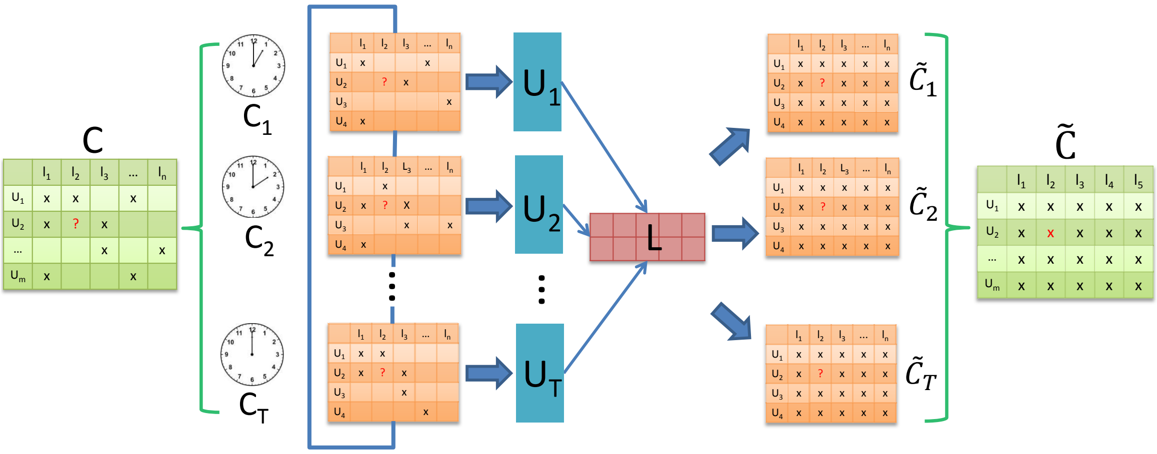



**Figure 2: Location Recommendation Framework with Temporal Effects**

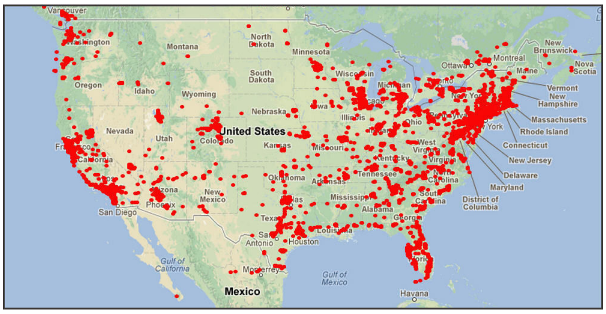



**Figure 3: The Check-in Distribution over the U.S.**

## 5.1 Dataset and Experiment Setup

We crawled the experimental dataset from Foursquare and obtained check-ins for three months (Jan 2011 - Mar 2011) to evaluate our proposed framework. The dataset is publicly available from the first author's homepage[3]. Foursquare allows a user to check in at a physical location via his cellphone, and then let his online friends know where he is by publishing such check-in action online. We select check-in locations which have been visited by at least two distinct users, and users who have checked in at least 10 distinct locations. The statistics of the final dataset are shown in Table 1. The majority of check-ins happened in the U.S.; Figure 3 shows the corresponding check-in distributions in the U.S.

We organize the dataset as a user-location matrix. The check-in density of the matrix is $8.84 \times 10^{-4}$. Logistic function $\frac{1}{1+(e^x)^{-1}}$ is commonly used in recommender system [15] to map each matrix element into [0,1]. We notice that in contrast with online item recommendation, where x (the rating of an item) is usually ranging from 1 to 5, in location recommendation, the value of x (check-in frequency of a location) is commonly large, while the function "$(e^x)^{-1}$" would result in very small and indistinguishable values, with x being larger than 7. Therefore, we adjust the mapping function as $\frac{1}{1+x^{-1}}$, with $x$ corresponding to $\widetilde{\mathbf{C}}(i,j)$ in our data, which works better than the logistic function in our experiment.

[3] http://www.public.asu.edu/~hgao16/dataset.html

**Table 1: Statistical Information of the Dataset**

| | |
|---|---|
| duration | Jan 1, 2011-Mar 31, 2011 |
| No. of users | 5,269 |
| No. of check-ins | 288,079 |
| No. of unique locations | 26,381 |
| Average check-ins per user | 55 |
| Check-in density | $8.84 \times 10^{-4}$ |

For each individual user in the dataset, we randomly mark off 20% and 40% of all locations that he has checked-in for testing. The rest of the user-location pairs are used as training data to infer $\mathbf{U}_t$ and $\mathbf{L}$ for location recommendation. The random selection is conducted 5 times individually, and we report the average results.

To evaluate the recommendation performance, we are interested in: (1) how many previously marked off locations are recommended to the users among the total number of recommended locations, and (2) how many previously marked off locations are recommended to the users among the total number of marked off locations. Thus, we use *precision@N* and *recall@N* as our evaluation metrics, defined as follows:

$$precision@N = \frac{\sum_{u_i \in U} |TopN(u_i) \bigcap L(u_i)|}{\sum_{u_i \in U} |TopN(u_i)|} \tag{15}$$

$$recall@N = \frac{\sum_{u_i \in U} |TopN(u_i) \bigcap L(u_i)|}{\sum_{u_i \in U} |L(u_i)|}, \tag{16}$$

where $TopN(u_i)$ is the set of locations recommended to user $u_i$ that $u_i$ has not visited in the training set. $L(u_i)$ is the set of locations that has been visited by $u_i$ in the testing set. In our experiment, $N$ is set to 5 and 10, respectively.

All the parameters in this paper are set through cross-validation. For the proposed method, the experimental results use d=10 dimensions to represent the latent features, the regularization coefficients $\alpha$ and $\beta$ are set to 2, and $\lambda$ is set to 1. As suggested in [23], the effectiveness of recommender systems with sparse datasets (i.e., low-density user-item matrix) is usually not high. For example, the reported top 5 precision is 5% over a dataset with $8.02 \times 10^{-3}$ density and 3.5% over a dataset with $4.24 \times 10^{-5}$ density [23, 25]. Therefore, the low precision obtained in our experiment is reasonable. In this paper, we focus on **comparing the relative performance of algorithms instead of comparing their absolute performance**.

## 5.2 Evaluating Data Properties of Temporal Non-uniformness and Consecutiveness

In this section, we discuss the properties of temporal non-uniformness and temporal consecutiveness in our dataset. The temporal non-uniformness property, which states that a user exhibits distinct check-in preferences at different hours of the day, is straightforward to evaluate with a two-sided hypothesis testing on the check-in behavior of two temporal states for each user. Our experiment shows that this property does hold in our dataset. Due to the space limit, we will ignore its evaluation details and focus on evaluating temporal consecutiveness. We firstly define the check-in similarity of a user between two temporal states $t_i$ and $t_j$:

$$sim_u(t_i, t_j) = \frac{\mathbf{C}_{t_i}(u,:) \cdot \mathbf{C}_{t_j}(u,:)}{|\mathbf{C}_{t_i}(u,:)|_2 \times |\mathbf{C}_{t_j}(u,:)|_2}, \tag{17}$$

where $\mathbf{C}_t(u,:)$ is the check-in vector of user $u$ at temporal state $t$. $|\bullet|_2$ is the 2-norm of a vector.

To evaluate temporal consecutiveness, we calculate two similarities for each user $u$: consecutive similarity $\mathbf{S}_c(u)$ and non-consecutive similarity $\mathbf{S}_n(u)$. $\mathbf{S}_c(u)$ is the average similarity of all $sim_u(t_i, t_j)$, where $t_i$ and $t_j$ are consecutive temporal states. Note that T temporal states have T consecutive temporal similarities in total, i.e., $sim_u(t_1, t_2)$, $sim_u(t_2, t_3)$,..., $sim_u(T-1, T)$, and $sim_u(T, 1)$. Similarly, $\mathbf{S}_n(u)$ is the average similarity of all $sim_u(t_i, t_j)$, where $t_i$ and $t_j$ are non-consecutive temporal states. For fair comparison, we randomly sample T non-consecutive temporal similarities $sim_u(t_i, t_j)$ to ensure that both $\mathbf{S}_c(u)$ and $\mathbf{S}_n(u)$ have the same sample size and then take the average to calculate $\mathbf{S}_n(u)$.

We conduct a two-sample t-test on the vectors $\mathbf{S}_c$ and $\mathbf{S}_n$. The null hypothesis is $H_0$: $\mathbf{S}_c \leq \mathbf{S}_n$, i.e., check-ins between consecutive temporal states are less or equally similar than that between non-consecutive temporal states. The alternative hypothesis is $H_1$: $\mathbf{S}_c > \mathbf{S}_n$. In our experiment, the null hypothesis is rejected at significant level $\alpha = 0.001$ with p-value of $5.6e$-191, i.e., a user's check-ins in two consecutive temporal states have a higher similarity than those in non-consecutive temporal states.

## 5.3 Comparison of Various Recommendation Models

In this section, we compare our proposed location recommendation framework **LRT** with various recommendation models. Three baseline methods are introduced w.r.t. time-dependent and static check-in preferences, as defined below:

- **User-Based Collaborative Filtering (CF)**
  User-based collaborative filtering is a state-of-the-art approach for recommender systems. We adopt the user-based recommender [29] for location recommendation. It computes a user's interest in a location based on other users' interests in that location. Temporal information is not considered in this approach.

- **Non-negative Matrix Factorization (NMF)**
  Non-negative Matrix Factorization (NMF) [13] computes non-negative user check-in preferences under the whole user-location matrix, which is our basic location recommendation model, as defined in Eq. (2), without temporal effects.

- **Random LRT (R-LRT)**
  We randomly divide the original user-location matrix **C** into 24 pieces $\mathbf{C}_t$ without considering the temporal state, and then apply the same recommendation process in Figure 2.

Figure 4 reports the comparison results of **LRT** with the proposed baseline methods. The aggregation strategy is selected as voting due to its superior performance (more details on the comparison of aggregation strategies will be discussed in the next subsection). The results precipitate several observations, which we summarize below:

- CF performs the worst among all the approaches. The data sparseness could be one reason to explain this performance. Due to the low density of the user-location matrix, the collaborative filtering approach fails to accurately recommend locations and performs worse than matrix factorization approaches, which leverage the low-rank approximation of user check-in preferences.

- Both NMF and R-LRT perform better than CF, demonstrating their ability in dealing with sparse data for location recommendation. Furthermore, the better performance of **LRT** than NMF suggests that time-dependent check-in preference capture user mobile behavior better than static check-in preferences.

- **LRT** performs better than R-LRT, suggesting that the division strategy is important. Our model, with the consideration of temporal effects, is able to improve location recommendation performance, while without an appropriate temporal division the matrix divide-aggregation strategy could result in a bad performance.

**LRT** performs the best among all the baseline methods. It considers time-dependent check-in preferences and outperforms approaches that capture static check-in preferences. The standard deviation of the performance from each method is less than $2 \times 10^{-4}$, confirming the reliability of our comparison results. As we mentioned before, **the recommendation effectiveness is usually low due to the sparseness of data with low density**. Therefore, the absolute performance on precision and recall of **LRT** seems to be small but is still reasonable and significant compared to other baseline methods.

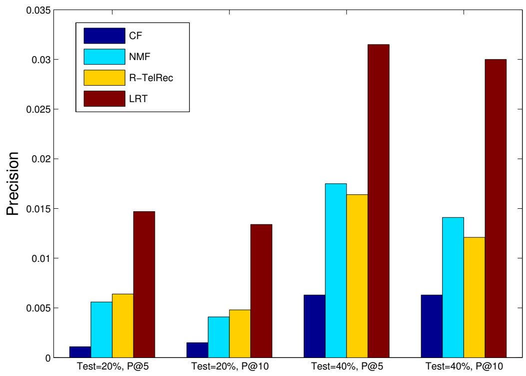
To further evaluate the significance of our framework, we launch a random recommendation [24]. For each user, we randomly select 5/10 locations from the total 26,381 locations (excluding locations that have been previously visited by the user), and recommend them to the user. The recommendation performance with this strategy is shown in Table 2. Compared to the random recommendation, our proposed framework is, on average, 73.27 times better than the random performance, demonstrating the power of temporal effects for improving location recommendation performance.

**Table 2: Performance of Random Recommendation**

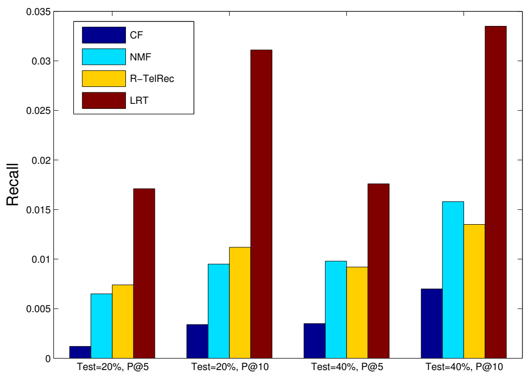
| Testing | Metrics | @5 | @10 |
|---|---|---|---|
| 20% | Precision | 0.0152% | 0.0190% |
| | Recall | 0.0177% | 0.0442% |
| 40% | Precision | 0.0266% | 0.0361% |
| | Recall | 0.0149% | 0.0403% |

## 5.4 Location Recommendation with Various Aggregation Strategies

In this subsection, we discuss the performance of various aggregation strategies. We compare the recommendation performance




(a) Recommendation Performance (Precision)




(b) Recommendation Performance (Recall)

**Figure 4: Performance of Location Recommendation Models**

of four aggregation strategies and list the results in Table 3 and Table 4. We summarize the essential observations below:

- The **mean** performs the worst among all the aggregation strategies. This is because taking the average of all the temporal preferences degrades the preference variance and makes the personal preferences indistinguishable. It validates the fact that a user's check-in preferences are highly dependent on the temporal state, approaches regardless of this may fail in recommending the right locations.

- The **maximum** has similar performance to the **sum**, suggesting that if a user's check-in preferences are strongly indicated by one temporal state, there is a high probability it indicates the true preferences of the user. This is also consistent with the observation reported by [10] that a user's check-in behavior presents Gaussian distribution over hours of the day, in which a user mostly checks-in at a location during a specific period of time and rarely visits during other time periods.

- The **voting** performs the best among all the aggregation strategies. Compared to the **sum**, it filters controversial location candidates at each temporal state, and reduces the uncertainty brought by the noisy location candidates, demonstrating its robustness in dealing with noisy data.

**Table 3: Comparison of Aggregation Strategies (Precision)**

| Testing | Metrics | Sum | Mean | Max | Voting |
|---|---|---|---|---|---|
| 20% | P@5 | 1.37% | 0.03% | 1.35% | 1.47% |
| | P@10 | 1.31% | 0.03% | 1.30% | 1.34% |
| 40% | P@5 | 3.08% | 0.46% | 3.10% | 3.20% |
| | P@10 | 2.95% | 0.44% | 2.95% | 3.00% |

**Table 4: Comparison of Aggregation Strategies (Recall)**

| Testing | Metrics | Sum | Mean | Max | Voting |
|---|---|---|---|---|---|
| 20% | R@5 | 1.60% | 0.03% | 1.57% | 1.71% |
| | R@10 | 3.05% | 0.08% | 3.03% | 3.11% |
| 40% | R@5 | 1.73% | 0.03% | 1.74% | 1.79% |
| | R@10 | 3.25% | 0.05% | 3.30% | 3.35% |

## 5.5 Exploring Various Temporal Patterns

**LRT** is designed to recommend locations to a user by taking advantage of temporal patterns. So far, we have evaluated its recommendation performance with daily patterns, while its recommendation ability is not limited to one specific temporal pattern. By taking different definitions of temporal state, many other temporal patterns can be used for location recommendation with **LRT**, as long as they contain the non-uniformness and consecutiveness properties. For example, we could define the temporal state as t=[1,T], with T=7 for weekly (day of the week) patterns, T=2 for weekday/weekend patterns, and T=12 for monthly (month of the year) patterns, etc. The only change is to divide the original user-location matrix $\mathbf{C}$ into a set of $\mathbf{C}_t$ according to the corresponding temporal state. Table 5 shows the recommendation results of **LRT** with weekly patterns and weekday/weekend patterns. Due to the space limit, we only present the results on testing size = 40%. The results indicate that weekly patterns and weekday/weekend patterns can also capture users' temporal check-in preferences and improve the location recommendation performance.

**Table 5: Comparison of Temporal Patterns**

| Temporal Patterns | Metrics | @5 | @10 |
|---|---|---|---|
| Day of the Week | Precision | 2.32% | 2.18% |
| | Recall | 1.30% | 2.45% |
| Weekday/Weekend | Precision | 2.23% | 2.04% |
| | Recall | 1.21% | 2.28% |

# 6. CONCLUSION AND FUTURE WORK

In this paper, we investigated the temporal properties of user check-in behavior on location-based social networks, and leveraged them to generate a location recommendation framework with temporal effects. The experimental results exhibit the power of temporal effects for capturing a user's mobile behavior, and demonstrate their potential ability in improving location recommendation performance. Considering the various types of temporal patterns, investigating other patterns (e.g., monthly/ yearly patterns) could provide the model with potential power to predict the future. On the other hand, how to integrate these patterns for location recommendation could also be an interesting direction for future work. Furthermore, it would be interesting to study the complementary effects of temporal patterns with social and geographical information on LBSNs, and leverage multiple resources to generate a spatial-temporal-social framework for location recommendation.

# 7. ACKNOWLEDGMENTS

This work is supported, in part, by ONR (N000141010091).

## 8. REFERENCES


[1] B. Berjani and T. Strufe. A recommendation system for spots in location-based online social networks. In *Proceedings of the 4th Workshop on Social Network Systems*, pages 1–6. ACM, 2011.

[2] J. Chang and E. Sun. Location 3: How users share and respond to location-based data on social networking sites. *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[3] C. Cheng, H. Yang, I. King, and M. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. *AAAI, Toronto, Canada*, 2012.

[4] Z. Cheng, J. Caverlee, K. Lee, and D. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

[5] E. Cho, S. Myers, and J. Leskovec. 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*, pages 1082–1090. ACM, 2011.

[6] C. Ding, T. Li, and M. Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 183–192. IEEE, 2008.

[7] Z. Y. Dingqi Yang, Daqing Zhang and Z. Wang. A sentiment-enhanced personalized location recommendation system. *ACM Hypertext*, 2013.

[8] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*, 2012.

[9] H. Gao, J. Tang, and H. Liu. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. *21st ACM International Conference on Information and Knowledge Management*, 2012.

[10] H. Gao, J. Tang, and H. Liu. Mobile location prediction in spatio-temporal context. *Nokia Mobile Data Challenge Workshop*, 2012.

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

[12] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[13] D. Lee, H. Seung, et al. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[14] H. Liu, B. Luo, and D. Lee. Location type classification using tweet content. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1, pages 232–237. IEEE, 2012.

[15] H. Ma, H. Yang, M. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[16] E. Malmi, T. Do, and D. Gatica-Perez. Checking in or checked in: Comparing large-scale manual and automatic location disclosure patterns. *The 11th International Conference on Mobile and Ubiquitous Multimedia (MUM 2012)*, 2012.

[17] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. *Proceeding of the 5th International AAAI Conference on Weblogs and Social Media*, 2011.

[18] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora. Distance matters: Geo-social metrics for online social networks. In *Proceedings of the 3rd conference on online social networks*, pages 8–8. USENIX Association, 2010.

[19] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. *Proceeding of the 5th International AAAI Conference on Weblogs and Social Media*, 11, 2011.

[20] J. Tang, H. Gao, and H. Liu. mTrust: Discerning multi-faceted trust in a connected world. In *the 5th ACM International Conference on Web Search and Data Mining*, 2012.

[21] J. Tang, H. Gao, H. Liu, and A. D. Sarma. eTrust: Understanding trust evolution in an online world. In *KDD*, 2012.

[22] M. Ye, K. Janowicz, C. Mülligann, and W. Lee. What you are is when you are: the temporal dimension of feature types in location-based social networks. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 102–111. ACM, 2011.

[23] M. Ye, X. Liu, and W. Lee. Exploring social influence for recommendation - a probabilistic generative approach. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334, 2012.

[24] M. Ye, P. Yin, and W. Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 458–461, 2010.

[25] M. Ye, P. Yin, W. Lee, and D. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 325–334, 2011.

[26] J. Ying, E. Lu, W. Kuo, and V. Tseng. Urban point-of-interest recommendation by mining user check-in behaviors. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, pages 63–70. ACM, 2012.

[27] C. Zhang, L. Shou, K. Chen, G. Chen, and Y. Bei. Evaluating geo-social influence in location-based social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1442–1451. ACM, 2012.

[28] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, 5(1):5, 2011.

[29] D. Zhou, B. Wang, S. Rahimi, and X. Wang. A study of recommending locations on location-based social network by collaborative filtering. *Advances in Artificial Intelligence*, pages 255–266, 2012.

[30] K. Zickuhr. Three-quarters of smartphone owners use location-based services. *Pew Internet & American Life Project*, 2012.