Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation

Shenglin Zhao[1,2], Tong Zhao[1,2], Haiqin Yang[1,2], Michael R. Lyu[1,2], Irwin King[1,2]
[1]Shenzhen Research Institute
The Chinese University of Hong Kong, Shenzhen, China
[2]Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{*slzhao, tzhao, hqyang, lyu, king*}*@cse.cuhk.edu.hk*

Abstract

Successive point-of-interest (POI) recommendation in location-based social networks (LBSNs) becomes a significant task since it helps users to navigate a number of candidate POIs and provides the best POI recommendations based on users' most recent check-in knowledge. However, all existing methods for successive POI recommendation only focus on modeling the correlation between POIs based on users' check-in sequences, but ignore an important fact that successive POI recommendation is a time-subtle recommendation task. In fact, even with the same previous check-in information, users would prefer different successive POIs at different time. To capture the impact of time on successive POI recommendation, in this paper, we propose a ***s***patial-***te***mpora***l*** ***la***tent ***r***anking (STELLAR) method to explicitly model the interactions among user, POI, and time. In particular, the proposed STELLAR model is built upon a ranking-based pairwise tensor factorization framework with a fine-grained modeling of user-POI, POI-time, and POI-POI interactions for successive POI recommendation. Moreover, we propose a new interval-aware weight utility function to differentiate successive check-ins' correlations, which breaks the time interval constraint in prior work. Evaluations on two real-world datasets demonstrate that the STELLAR model outperforms state-of-the-art successive POI recommendation model about 20% in Precision@5 and Recall@5.

Introduction

Location-based social networks (LBSNs) become increasingly popular and provide users a new way to share their location and experience about point-of-interests (POIs) via check-in behaviors. To navigate users the most suitable POIs, POI recommendation methods are developed and play an important role in LBSN services. POI recommendation aims to learn users' preferences based on their check-in records and then recommend users preferred POIs. Several methods are proposed for POI recommendation. Ye et al. use memory-based methods to recommend POIs (Ye et al. 2011). Some other researchers (Cheng et al. 2012; Gao et al. 2013; Lian et al. 2014) turn to model-based methods to improve the scalability.

Successive POI recommendation, as a natural extension of general POI recommendation, is recently proposed and has attracted great research interest. Different from general POI recommendation that focuses only on estimating users' preferences on POIs, successive POI recommendation provides satisfied recommendations promptly based on users' most recent checked-in location, which requires not only the preference modeling from users but also the accurate correlation analysis between POIs. (Cheng et al. 2013) utilizes a personalized Markov chain and region localization to solve the problem. (Feng et al. 2015) proposes a personalized metric embedding method to model the check-in sequences. However, all previous methods ignore to investigate the impact of time on successive POI recommendation.

Successive POI recommendation is a time-subtle recommendation task since at different time, users would prefer different successive POIs. It is easy to imagine that a user may go to a restaurant after leaving from office at noon, while he/she may be more likely to go to a gym when he/she leaves office at night. However, previous successive POI recommendation methods only highlight the modeling of correlations between POIs within users' check-in sequences, but neglect to model such a time-sensitive property.

In this paper, we try to understand the underlying mechanism of how time influences successive POI recommendation performance. To motivate this work, we first conduct an empirical analysis on two real world LBSN datasets to verify that time is an important factor to affect users' successive POI check-in behaviors. Based on the analysis, we propose a ***s***patial-***te***mpora***l*** ***la***tent ***r***anking (STELLAR) model to recommend users most possible successive POIs based on their most recent check-in and the querying time stamp. The proposed STELLAR model is built upon a ranking-based pairwise tensor factorization framework with a fine-grained modeling of user-POI, POI-time, and POI-POI interactions for successive POI recommendation. To overcome the weaknesses of prior latent ranking models(e.g., Tucker Decomposition, Canonical Decomposition, and Pairwise Interaction Tensor Factorization) that suffer from coupled interaction on POI feature, we represent each POI by three different latent feature vectors and model the three kinds of interactions separately. Moreover, the proposed STELLAR method contains two specific characteristics making it more suitable for successive POI recommendation: 1) we design a three-slice time indexing scheme to capture the temporal features of check-in behavior–periodicity and preference variance;

2) we introduce an interval-aware weight utility function to differentiate the correlations of successive check-ins, which breaks the time interval constraint in prior work.

The contributions of this paper are as follows:

- We propose a time-aware successive POI recommendation method–the STELLAR model, by considering the time information. In this model, we employ a new POI latent feature representation means to resolve the problem of coupled interaction. Experimental results demonstrate our STELLAR model outperforms state-of-the-art successive POI recommendation method.
- We design a three-slice time indexing scheme to represent the time stamps, which captures the user check-ins specific characteristics: periodicity and preference variance. Experimental results show that our model better captures the temporal effect than state-of-the-art temporal model for POI recommendation.
- We introduce a new interval-aware weight utility function to differentiate successive check-ins' correlations, which improves the successive POI recommendation accuracy.

Related Work

In this section, we first review the literature of POI recommendation. Then we present the connection of our proposed model and prior work.

POI recommendation. POI recommendation is an important task in LBSNs. Ye et al. firstly discuss how to use memory-based methods to recommend POIs (Ye, Yin, and Lee 2010; Ye et al. 2011). In order to improve the memory-based models, advanced techniques are then leveraged to capture more information, including social and geographical influence (Wang, Terrovitis, and Mamoulis 2013; Zhang and Chow 2013; 2015), temporal effect (Yuan et al. 2013), and sequential check-ins' influence (Zhang, Chow, and Li 2014; Zhang and Chow 2015). On the other hand, model-based methods are proposed for the seek of scalability, most of which base on the latent ranking techniques. (Cheng et al. 2012) proposes a multi-center Gaussian model to capture user geographical influence and combines it with matrix factorization (MF) model (Koren, Bell, and Volinsky 2009) to recommend POIs. (Gao et al. 2013) proposes an MF-based model which captures the temporal effect to improve performance. (Yang et al. 2013), (Hu and Ester 2014), and (Gao et al. 2015) leverage user comments to improve the POI recommendation system. (Lian et al. 2014) and (Liu et al. 2014) improve POI recommendation by incorporating geographical information in a weighted regularized matrix factorization model. Instead of estimating the user preference score on POIs, (Cheng et al. 2013) and (Li et al. 2015) establish ranking models to learn the recommender system. Other techniques for POI recommendation include generative graphical models, metric learning techniques, and graph-based method. Readers may refer the papers and references therein (Feng et al. 2015; Ye, Zhu, and Cheng 2013; Yuan, Cong, and Sun 2014; Liu et al. 2013; Kurashima et al. 2013; Yin et al. 2013).

Connection to prior work. We focus on successive POI recommendation in this paper, which recommends POIs on basis of a user's most recent check-in. (Cheng et al. 2013) utilizes the latent ranking model to solve the problem, while (Feng et al. 2015) employs the metric embedding. Our work is most related to (Cheng et al. 2013). However, prior work does not consider the time effect on successive POI recommendation, which motivates us to propose the STELLAR model. Moreover, we propose a three-slice time indexing scheme to represent the time stamps and introduce an interval-aware weight utility function to differentiate the correlations of successive check-ins.

Data Description and Successive Check-in Analysis

Before we introduce the proposed method, in this section, we first introduce two real world LBSN datasets used in this paper and then conduct some empirical analysis on them to explore the spatial and temporal properties of users' successive check-in behaviors.

Data Description

We use two check-in datasets crawled from real world LBSNs: one is Foursquare data provided in (Gao, Tang, and Liu 2012) and the other is Gowalla data (Zhao, King, and Lyu 2013). Both contain users' check-in history from January 1, 2011 to July 31, 2011. We filter the POIs checked-in by less than 5 users and then choose users who check-in more than 10 times as our samples. After the preprocessing, the datasets contain the statistical properties as shown in Table 1.

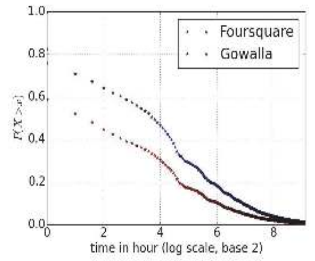
Table 1: Statistics of Datasets

	Foursquare	**Gowalla**
#users	10,034	3,240
#POIs	16,561	33,578
#check-ins	865,647	556,453
Density	0.0015	0.0028

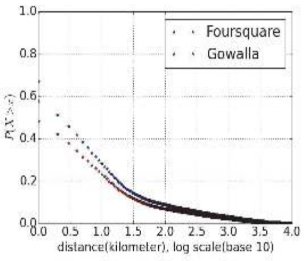
Successive Check-in Analysis

Now we conduct some empirical analysis to demonstrate the spatial and temporal properties of users' successive check-in behaviors.

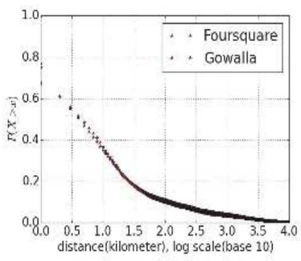
Spatial and temporal analysis. Successive check-ins demonstrate significant spatial and temporal property, shown in Figure 1. Figure 1(a) and 1(b) show the complementary cumulative distribution function (CCDF) of intervals and distances in successive check-ins. We verify the observation in (Cheng et al. 2013) that many successive check-ins are highly correlated especially in spatial relation: over 40% and 60% successive check-in behaviors happen in less than 4 hours since last check-in in Foursquare and Gowalla respectively; about 90% successive check-ins happen in less than 32 kilometers (half an hour driving distance) in Foursquare and Gowalla. Further, we check the CCDF of distances in successive check-ins that happen beyond 4 hours, shown in Figure 1(c). We observe although being



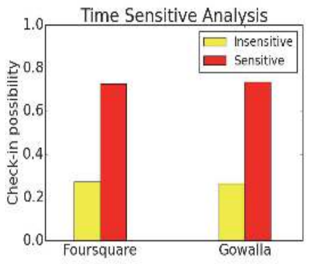
(a) CCDF of intervals in successive check-ins



(b) CCDF of distances in successive check-ins



(c) CCDF of distances in successive check-ins beyond 4 hours



(d) Time sensitive analysis of successive POI check-ins

Figure 1: Successive check-ins' spatial-temporal property

weaker the spatial correlations still exist: about 80% successive checked-in POIs happen in less than 32 kilometers. It is not hard to explain the phenomenon: a user always acts around his/her home or office, so the successive check-in, even independent with the last check-in, still possibly happens in the same activity area. Hence, successive checked-in POIs are generally spatially correlated, while successive check-ins in shorter interval contain stronger correlation.

Time sensitive analysis. Besides the spatial and temporal contiguity, we observe that users' successive check-ins are time-sensitive behaviors. We count in all users and calculate (1) the average probability of a previous check-in leading to the same successive POI at different time (time-insensitive) and (2) the average probability of a check-in followed by different successive POIs at different time stamps (time-sensitive). Figure 1(d) shows the analytical results. We can obviously find that with different time, given the same previous POI check-in, users' successive POI check-ins would be different. This observation triggers us to incorporate time impact into successive POI recommendation.

Spatial-Temporal Latent Ranking Model

In this section, we will detail the ***s**patial-**te**mpora**l** **la**tent* ***r**anking* (STELLAR) model for successive POI recommendation. We first demonstrate how to index time stamps in our model. Then we introduce the formulation our STELLAR model. Finally, we demonstrate how to make the model inference and learn the system.

Time Indexing Scheme

To capture the check-in behavior's specific temporal characteristics, we design a novel time indexing scheme to smoothly encode a standard time stamp to a particular time id. The check-in behavior's temporal characteristics contain two aspects: (1) Periodicity (Cho, Myers, and Leskovec 2011; Yuan et al. 2013). For example, users always visit restaurants at noon and bars at night; users check-in POIs around the office in weekdays but visit malls for shopping in weekends. (2) Preference variance (Gao et al. 2013). Users' check-in preferences change with time. In addition, the preference variance exists in three scales: hours of a day, different days of a week, and different months of a year, which is observed in (Gao et al. 2013) but not modeled. Our proposed scheme captures the two properties as follows. First, a time stamp is divided into three slices in terms of month, weekday type, and hour slot. Next, we split a week into weekday and weekend and a day into the following four sessions: the morning session from 6:00 a.m. to 10:59 a.m., the afternoon and night session from 0:00 a.m. to 2:59 a.m. and 3:00 p.m. to 11:59 p.m., two transitive sessions that range from 3:00 a.m. to 5:59 a.m. and 11:00 a.m. to 2:59 p.m.. Further, we use 4 bits to represent the month information, 1 bit to denote weekday or weekend, and 2 bits to show the hour session. Finally we convert the binary code into a unique decimal digit as the time id, where the id is in the range of 0 to 95. Figure 2 demonstrates the procedure of encoding an exemplary time stamp, "2011-04-05 18:10:23".

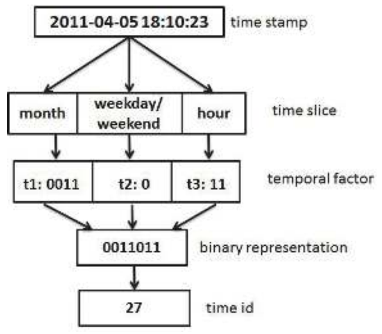


Figure 2: Time encoding demonstration

Model Formulation

The STELLAR system aims to provide time-aware successive POI recommendations. The task needs to learn a score function for a given user u to a candidate POI l^c at the time stamp t given his/her last check-in as a query POI l^q, which is defined as follows:

$$f(u, l^q, t, l^c), \tag{1}$$

where $f : \mathcal{U} \times \mathcal{L} \times \mathcal{T} \times \mathcal{L} \rightarrow \mathbb{R}$ maps a four-tuple tensor to real values. $\mathcal{U}$, $\mathcal{L}$, and $\mathcal{T}$ denote the set of users, the set of POIs, and the set of time ids, respectively. The score value represents the "successive check-in possibility" of a user to a candidate POI at the time stamp given the query POI.

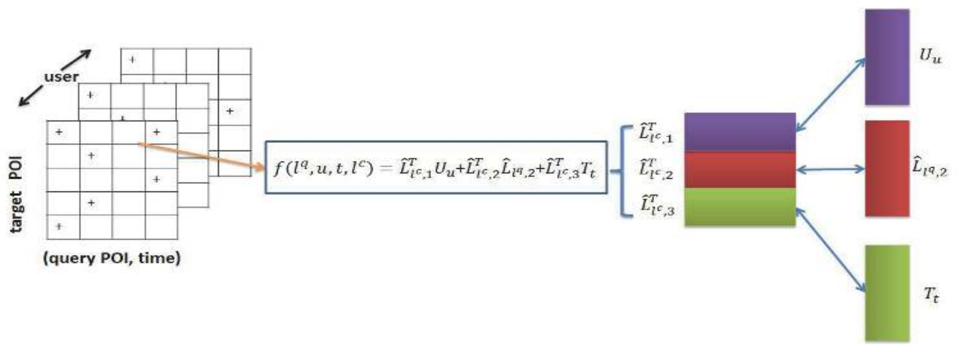


Figure 3: STELLAR model formulation demonstration

We establish a latent ranking framework to learn the score function, which employs pairwise tensor interactions to represent the following three key factors affecting users' check-in behavior: (1) the preference of a user u to a candidate POI l^c, (2) the temporal effect of time t on a candidate POI l^c, and (3) correlation of the last checked-in POI l^q and a candidate POI l^c. Correspondingly, the score value of $f(u, l^q, t, l^c)$ is determined by user-POI interaction, time-POI interaction, and POI-POI interaction together. In this case, a single vector representation for each POI is not semantically enough to capture the three different kinds of interactions. Therefore, we define a $3 \times d$ matrix to represent POI latent feature, where d is the latent space dimension. For each POI, three latent vectors are used to describe the POI-user, POI-time, and POI-POI interactions respectively. As shown in Figure 3, we formulate the function $f(u, l^q, t, l^c)$ as

$$f(u, l^q, t, l^c) = \hat{L}^T_{l^c,1} U_u + \hat{L}^T_{l^c,2} \hat{L}_{l^q,2} + \hat{L}^T_{l^c,3} T_t, \qquad (2)$$

where $U_u, T_t \in R^d$ are latent vectors of user u and time t, $\hat{L}_{l^c,1}, \hat{L}_{l^c,2}, \hat{L}_{l^c,3} \in R^d$ are candidate POI l^c's three d-dimension vectors which correspondingly interact with users, other POIs, and time labels, and $\hat{L}_{l^q,2}$ is query POI l^q's latent vector interacting to the candidate POI. Similar to (Gao et al. 2013; 2015), we set all latent vectors are **non-negative** to ensure better performance and real-world explanations on LBSNs for latent features. Further we denote $U \in R^{d \times |\mathcal{U}|}$ as the user latent matrix and $T \in R^{d \times |\mathcal{T}|}$ as the time latent matrix. In addition, we use $\hat{L}$, a $3 \times d \times |\mathcal{L}|$ tensor, to denote the POI latent factor.

From the observations in Figure 1, we find that successive POIs in a shorter interval contain a stronger correlation. To depict this observation, we introduce a weight utility function to differentiate the strong and weak correlations. The weight score value is in the range of [0,1], and the function is non-increasing with the duration of two successive check-ins. When two successive check-ins happen within a threshold interval, we assume they are highly correlated. Otherwise the correlation decreases with the increase of the time interval. In formal, we define the weight utility function as follows:

$$w = \begin{cases} 0.5 + \frac{2}{\Delta T} & \Delta T \geq s \\ 1 & otherwise \end{cases}, \qquad (3)$$

where ΔT is the interval of successive check-ins, in unit of hour; and s is the threshold of differentiating the correlations. In our experiments, s is set as 4 to get best performance. The check-in time of query POI l^q and current time t determine the interval ΔT. So we are able to refine the score function as

$$f(u, l^q, t, l^c, w) = \hat{L}^T_{l^c,1} U_u + w \cdot \hat{L}^T_{l^c,2} \hat{L}_{l^q,2} + \hat{L}^T_{l^c,3} T_t, \quad (4)$$

where w is the weight value to measure the POI-POI interaction.

The STELLAR model is proposed to handle the following two challenging issues: (1) **Disastrous sparsity**. Prior methods learn a model from a tensor with only three tuples. In our formulation, we focus on tuples of four elements, (user, POI, time, POI), which increase the sparsity of the tensor significantly. (2) **Coupled interaction**. The tuples in previously proposed tensor related methods are independent. For example, in (Rendle et al. 2009a; Rendle and Schmidt-Thieme 2010), the tuple includes user, item, and tag, which are independent. This is easier for updating the models. However, in our constructed tensor, the tuple includes two POIs coupled in the updating. This makes the previous tensor decomposition methods unsatisfactory. Our method represents the POI feature via a matrix and then models the three kinds of interactions separately. Further, we simplify the tensor completion problem as a combination of three low rank matrix factorization problems, which mitigates the sparsity trouble.

Model Inference and Learning

We make the model inference via learning the ranking order of successive check-in possibilities. Because we care more about the ranking order of the candidate POIs rather than the real values of check-in possibilities when recommending successive POIs for users. We follow the optimization criteria used in (Rendle et al. 2009b) and propose a pairwise ranking-based objective function for the proposed STELLAR model.

We demonstrate the inference procedure following (Rendle et al. 2009b). First we suppose that the scores of $f(u, l^q, t, l^c)$ at checked-in POIs are higher than the unchecked-in counterparts. Then we define the order $l^c_p >_{u,l^q,t} l^c_n$, which means at time t, given query POI l^q, user

u visits POI l_p^c but not l_n^c. Further we suffice to extract the set of all pairwise preference constraints

$$D_S := \{(u, l^q, t, l_p^c, l_n^c) | l_p^c >_{u,l^q,t} l_n^c\}. \quad (5)$$

Suppose the tuples in D_S are independent of each other, then to learn the parameters in the score function is to minimize the negative log likelihood of all the pair orders. Further, we add a Frobenius norm term to regularize the parameters to avoid the risk of overfitting. Then the objective function is

$$\mathcal{O} := \arg\min_{\Theta} \sum_{(u,l^q,t,l_p^c,l_n^c) \in D_S} - ln(\sigma(f(u, l^q, t, l_p^c) - f(u, l^q, t, l_n^c))) + \lambda ||\Theta||_F^2, \quad (6)$$

where σ is the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$, λ is the regularization parameter, and Θ denotes the parameter set, including U, T, and $\hat{L}$.

We leverage the stochastic gradient decent (SGD) algorithm to learn the objective function for efficacy. Denote $\delta = 1 - \sigma(y_{u,l^q,t,l_p^c,l_n^c})$, then we get the derivative of each parameter $\theta \in \Theta$ for a tuple $(u, l^q, t, l_p^c, l_n^c)$ as

$$\frac{\partial \mathcal{O}}{\partial \theta} = \begin{cases} -\delta \cdot (\hat{L}_{l_p^c,1} - \hat{L}_{l_n^c,1}) + \lambda \cdot U_u & \theta = U_u \\ -\delta \cdot (\hat{L}_{l_p^c,3} - \hat{L}_{l_n^c,3}) + \lambda \cdot T_t & \theta = T_t \\ -\delta \cdot w \cdot (\hat{L}_{l_p^c,2} - \hat{L}_{l_n^c,2}) + \lambda \cdot \hat{L}_{l^q,2} & \theta = \hat{L}_{l^q,2} \\ -\delta \cdot U_u + \lambda \cdot \hat{L}_{l_p^c,1} & \theta = \hat{L}_{l_p^c,1} \\ -\delta \cdot w \cdot \hat{L}_{l^q,2} + \lambda \cdot \hat{L}_{l_p^c,2} & \theta = \hat{L}_{l_p^c,2} \\ -\delta \cdot T_t + \lambda \cdot \hat{L}_{l_p^c,3} & \theta = \hat{L}_{l_p^c,3} \\ \delta \cdot U_u + \lambda \cdot \hat{L}_{l_n^c,1} & \theta = \hat{L}_{l_n^c,1} \\ \delta \cdot w \cdot \hat{L}_{l^q,2} + \lambda \cdot \hat{L}_{l_n^c,2} & \theta = \hat{L}_{l_n^c,2} \\ \delta \cdot T_t + \lambda \cdot \hat{L}_{l_n^c,3} & \theta = \hat{L}_{l_n^c,3}. \end{cases} \quad (7)$$

To ensure the non-negativity, we project the learned parameter to non-negative value. We define the projected operator $P(\cdot) : R^d \to R^d$ as $P[x_i] = \max(0, x_i), i = 1, \ldots, d$. For each sampled tuple $(u, l^q, t, l_p^c, l_n^c) \in D_S$, we update each parameter $\theta \in \Theta$ through the derivative,

$$\theta \leftarrow P(\theta - \gamma \frac{\partial \mathcal{O}}{\partial \theta}); \quad (8)$$

where γ is the learning rate. To train the model, we draw the tuple from D_S with bootstrap sampling rule, following (Rendle et al. 2009b). Algorithm 1 gives the detailed procedure to learn the STELLAR model. The convergent condition is satisfied when the negative log likelihood value for a fixed sampled tuples does not decrease.

Complexity. Calculating the preference score of a tuple (u, l^q, t, l^c) costs $O(d)$, where d is the latent vector dimension. The updating procedure for each parameter is also in $O(d)$. Hence training an example (u, l^q, t, l^c) is in $O(k \cdot d)$, where k is the number of sampled unchecked POIs. Therefore, the runtime of training the model is in $O(N \cdot k \cdot d)$, where N is the number of training examples.

Experiment

We conduct experiments to answer the following questions: 1) how our method performs comparing with state-of-the-art models? 2) whether our time indexing scheme works well? 3) how the parameters affect the model performance?

Algorithm 1 STELLAR model learning algorithm

```
Input: Training tuples $\{(u_i, l_i^q, t_i, l_i^c)\}_{i=1,...,N}$
Output: $U, T, \hat{L}$
1: Initialize $U, T, \hat{L}$
2: repeat
3:    Draw $(u, l^q, t, l_p^c)$ uniformly from training tuples
4:    For $s = 1, \cdots, k$, $k$ is #sampled unchecked POIs
5:        Draw $(u, l^q, t, l_p^c, l_n^c)$ uniformly
6:        Update parameters according to Eq. (8)
7: until convergence
8: return $U, T, \hat{L}$
```

Experimental Setting

In this paper, we evaluate our model on two datasets with statistics shown in Table 1. The system recommends a user a list of POIs, given his/her last checked-in POI and time stamp as query. It is equivalent to solve the collaborative retrieval task (Weston et al. 2012), treating (query POI, time id, weight) as query for each user. Following setting in (Weston et al. 2012), we extract tuples of (user, query POI, time id, weight, POI) from all successive check-ins. Here we get time id from the check-in time stamp via encoding procedure. And the weight value is calculated according to the interval between two successive check-ins through the utility function in Eq. (3). In order to make our model effective for future check-ins, we split the tuples into two parts, 80% and 20% according to time sequential order. So we take the first group of tuples for training and the second group for test. Finally, we measure different models through Precision@5 and Recall@5, which are general metrics for POI recommendation problem used in prior work (Cheng et al. 2013; Gao et al. 2013; Ye et al. 2011).

Comparison Methods

Our Methods. We propose three methods: **TLAR**, **SLAR**, and **STELLAR**. TLAR and SLAR methods are special cases of STELLAR, which correspondingly only ignore the POI-POI interaction and time-POI interaction.

Baselines. We compare our proposed model with state-of-the-art latent ranking models and POI recommendation methods. Prior work (Lian et al. 2014; Liu et al. 2014) indicates that treating the check-ins as implicit feedback is better to recommend POIs. Hence we introduce two comparative latent ranking methods that model the check-ins as implicit feedback: **WRMF** (Hu, Koren, and Volinsky 2008; Pan et al. 2008) and **BPRMF** (Rendle et al. 2009b). In addition, we introduce two state-of-the-art POI recommendation methods: **LRT** (Gao et al. 2013) and **FPMC-LR** (Cheng et al. 2013). LRT is state-of-the-art model that incorporates temporal information in a latent ranking model to improve POI recommendation. FPMC-LR is the state-of-the-art successive POI recommendation model.

Table 2: Performance Comparison

		BPRMF	WRMF	LRT	FPMC−LR	TLAR	SLAR	STELLAR
Gowalla	P@5	0.025	0.031	0.033	0.048	0.053	0.050	**0.059**
	R@5	0.020	0.025	0.030	0.167	0.204	0.197	**0.226**
Foursquare	P@5	0.031	0.033	0.061	0.109	0.119	0.114	**0.129**
	R@5	0.027	0.028	0.053	0.347	0.373	0.368	**0.425**

Experimental Results

In the following, we demonstrate the performance comparison. We set latent dimension as 40, and train different models to get their best performances at appropriate parameters.

Baselines vs. Our Methods. Table 2 shows the experimental results on Foursquare and Gowalla data. We see that: 1) Our proposed model outperforms state-of-the-art latent ranking methods and POI recommendation models. Compared with state-of-the-art successive POI recommendation method, STELLAR model gains about 22.9% and 35.3% improvement for Gowalla, and 18.3% and 22.5% improvement for Foursquare on Precision@5 and Recall@5 . We observe that all models perform much better on Foursquare dataset than Gowalla dataset, even though it is sparser. The reason lies in Foursquare data contain much less POIs. 2) Our proposed models and FPMC-LR perform much better than other models, especially at recall measure. The reason lies in that these models leverage more conditions for each query. Our models recommend a user POIs given a user's recent check-in, the specific time stamp, or both; and FPMC-LR recommends POIs given a user's recent checked-in POIs. On the contrary, other three models give general recommendations.

LRT vs. TLAR. The experimental results show that TLAR outperforms LRT model. Our model depicts the temporal effect with a latent feature, which gets rid of sparsity problem suffering in LRT model. Furthermore, since TLAR is a special case of STELLAR, it means that STELLAR model captures the temporal effect well from the time stamps.

FPMC-LR vs. SLAR. The experimental results show that SLAR outperforms FPMC-LR model. It means SLAR model really improves the recommendation performance by differentiating the correlations of successive check-ins.

Discussion of Time Indexing Scheme

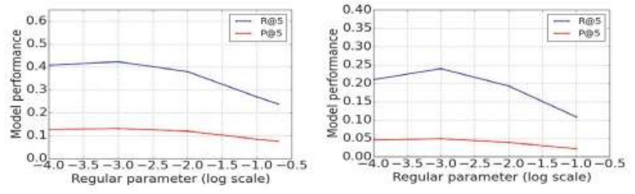
Our three-slice time indexing scheme effectively captures the temporal effect in three scales. In order to demonstrate its efficacy, we ignore one slice to index the time and then compare their results with our model, shown in Table 3. 'M', 'W', and 'D' represent month, week, and day slice respectively. Our model demonstrates the best performance.

Table 3: Comparison of Different Time Schemes

		M+W	M+D	W+D	M+W+D
Gowalla	P@5	0.051	0.053	0.054	**0.059**
	R@5	0.207	0.208	0.219	**0.226**
Foursquare	P@5	0.118	0.120	0.121	**0.129**
	R@5	0.371	0.389	0.398	**0.425**

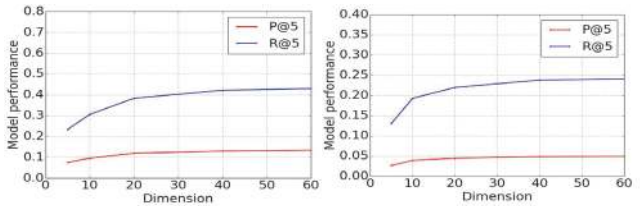
Parameter Effect

The regularization and latent dimension are important parameters to learn a latent ranking model. Figure 4 and 5 demonstrate the effect of the parameters on model performance. For simplicity, we set the same value for all latent vectors' regularizations in the model. The model has best performance when $\lambda = 0.001$. The performance of Stellar steadily rises with the increase of latent vector dimension. For the trade-off of performance and computation cost, we suggest to set dimension $d = 40$.



(a) Fousquare (b) Gowalla

Figure 4: The effect of regularization



(a) Foursquare (b) Gowalla

Figure 5: The effect of latent dimension

Conclusion and Further Work

In this paper, we study the problem of successive POI recommendation. Compared with previous work, we show that successive POI recommendation is a time-subtle recommendation task. To capture the time impact, we first design a time indexing scheme to smoothly encode time stamps to particular time ids and then incorporate the time ids into our proposed STELLAR model. Further, we establish the STELLAR model upon a ranking-based pairwise interaction tensor factorization framework with a fine-grained modeling of the interactions among time, user, and POI. Experimental results on two datasets, Foursquare and Gowalla, show that the STELLAR model outperforms state-of-the-art models. Our further work may incorporate more information in this system, e.g., users' comments and social relations.

Acknowledgments

The work described in this paper was fully supported by the National Grand Fundamental Research 973 Program of China (No. 2014CB340401 and No. 2014CB340405), the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 14203314 and CUHK 14205214), and Microsoft Research Asia Regional Seed Fund in Big Data Research (Grant No. FY13-RES-SPONSOR-036).

References

Cheng, C.; Yang, H.; King, I.; and Lyu, M. R. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*.

Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2605–2611. AAAI Press.

Cho, E.; Myers, S. A.; and Leskovec, J. 2011. Friendship and mobility: User movement in location-based social networks. In *SIGKDD*, 1082–1090. ACM.

Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q. 2015. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*.

Gao, H.; Tang, J.; Hu, X.; and Liu, H. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*, 93–100. ACM.

Gao, H.; Tang, J.; Hu, X.; and Liu, H. 2015. Content-aware point of interest recommendation on location-based social networks. In *AAAI*.

Gao, H.; Tang, J.; and Liu, H. 2012. gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *CIKM*, 1582–1586. ACM.

Hu, B., and Ester, M. 2014. Social topic modeling for point-of-interest recommendation in location-based social networks. In *ICDM*, 845–850.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*, 263–272.

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

Kurashima, T.; Iwata, T.; Hoshide, T.; Takaya, N.; and Fujimura, K. 2013. Geo topic model: Joint modeling of user's activity area and interests for location recommendation. In *WSDM*, 375–384. New York, NY, USA: ACM.

Li, X.; Cong, G.; Li, X.-L.; Pham, T.-A. N.; and Krishnaswamy, S. 2015. Rank-GeoFM: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, 433–442.

Lian, D.; Zhao, C.; Xie, X.; Sun, G.; Chen, E.; and Rui, Y. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *SIGKDD*, 831–840.

Liu, B.; Fu, Y.; Yao, Z.; and Xiong, H. 2013. Learning geographical preferences for point-of-interest recommendation. In *SIGKDD*, 1043–1051. ACM.

Liu, Y.; Wei, W.; Sun, A.; and Miao, C. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, 739–748.

Pan, R.; Zhou, Y.; Cao, B.; Liu, N. N.; Lukose, R.; Scholz, M.; and Yang, Q. 2008. One-class collaborative filtering. In *ICDM*, 502–511. IEEE.

Rendle, S., and Schmidt-Thieme, L. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, 81–90. ACM.

Rendle, S.; Balby Marinho, L.; Nanopoulos, A.; and Schmidt-Thieme, L. 2009a. Learning optimal ranking with tensor factorization for tag recommendation. In *SIGKDD*, 727–736. ACM.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009b. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, 452–461. AUAI Press.

Wang, H.; Terrovitis, M.; and Mamoulis, N. 2013. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL*, 364–373. ACM.

Weston, J.; Wang, C.; Weiss, R.; and Berenzweig, A. 2012. Latent collaborative retrieval. *ICML*.

Yang, D.; Zhang, D.; Yu, Z.; and Wang, Z. 2013. A sentiment-enhanced personalized location recommendation system. In *HT*. ACM.

Ye, M.; Yin, P.; Lee, W.-C.; and Lee, D.-L. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, 325–334. ACM.

Ye, M.; Yin, P.; and Lee, W.-C. 2010. Location recommendation for location-based social networks. In *SIGSPATIAL*, 458–461. ACM.

Ye, J.; Zhu, Z.; and Cheng, H. 2013. What's your next move: User activity prediction in location-based social networks. In *SDM*.

Yin, H.; Sun, Y.; Cui, B.; Hu, Z.; and Chen, L. 2013. Lcars: A location-content-aware recommender system. In *SIGKDD*, 221–229.

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

Yuan, Q.; Cong, G.; and Sun, A. 2014. Graph-based point-of-interest recommendation with geographical and temporal influences. In *CIKM*, 659–668. ACM.

Zhang, J.-D., and Chow, C.-Y. 2013. iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach. In *SIGSPATIAL*, 334–343.

Zhang, J.-D., and Chow, C.-Y. 2015. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *SIGIR*, 443–452.

Zhang, J.-D.; Chow, C.-Y.; and Li, Y. 2014. LORE: Exploiting sequential influence for location recommendations. In *SIGSPATIAL*.

Zhao, S.; King, I.; and Lyu, M. R. 2013. Capturing geographical influence in POI recommendations. In *ICONIP*, 530–537. Springer.