2018 19th IEEE International Conference on Mobile Data Management

POI Recommendation of Location-Based Social Networks Using Tensor Factorization

Liao Guoqiong, Jiang shan, Zhiheng Zhou, Wan Changxuan, and Liu Xiping

School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China

(liaoguoqiong@163.com, 879123406@qq.com, 785172991@qq.com, wanchangxuan@263.net,lewislxp@gmail.com)

***Abstract*—With the rapid development of wireless communication technologies, location-based social networks (LBSNs) like foursquare and Gowalla have become very popular. Point of interest (POI) recommendation is a kind of important recommendation in LBSNs for enhancing user experiences. Unlike online social networks, LBSNs have a great deal of check-in data and comment information, which can provide valuable information for POI recommendation. In this paper, a novel recommendation strategy using tensor factorization is proposed for improving accurate rate of POI recommendation. Firstly, the latent dirichlet allocation(LDA) topic model is used to extract topic information and generate topic probability distribution of each POI based on comment information from users. Secondly, the check-in data of each user is divided into multiple data slices corresponding to each hour of a day. By connecting with the topic distributions of the visited POIs of each user, a user-topic-time tensor is conducted to present the potential preferences of all users. Finally, a higher order singular value decomposition (HOSVD) algorithm is employed to decompose the third-order tensor, to get dense preference information for POI recommendation. The experiments on a real dataset show that the proposed approach have better performance than the baseline methods.**

***Keywords*—POI recommendation; Location-based social networks; LDA model; Tensor factorization**

I. Introduction

In recent years, with the rapid development of wireless communication technology such as global positioning system (GPS) and Wi-Fi technology, location-based social networks (LBSNs) like foursquare and Gowalla have become very popular. In LBSNs, users can build connections with their friends, upload photos, and share their locations via check-ins of points of interest (POIs), which are specific locations that someone may find useful or interesting things, such as restaurants, tourists spots, cinema, stores, etc.

Unlike online social networks, besides the social relations between users, the LBSNs have many geographic relations between locations, check-in relations and comment relations between users and locations. These rich relations in the LBSNs can be effectively utilized for POI recommendation, to enhance user experiences and improve user dependence on the LBSNs. However, differing from traditional item recommendation such as books and videos, the POI recommendation mainly faces on following challenges:

• **No-explicit user preferences.** Although there are a large number of check-in data in the LBSNs, the data cannot give explicit user preferences directly, since they only indicate the facts that some users visited some POIs [1].

• **Non-consistency of interests.** Generally speaking, the preferences of one user are not consistent in different time within a day. For example, someone likes to drink coffee in the morning, does fitness in the afternoon, while goes to KTV at night. Since the preferences of a user may be different with the time, the POI recommendation strategy should be time-aware [2].

• **Data sparsity.** As we know, there are a lot of POIs in the LBSNs. However, the check-in information of one user is always very limited, so the check-in data is sparse data [3].

Therefore, the traditional item recommendation strategies cannot be applied to POI recommendation effectively. In recent years, researchers pay more and more attentions on POI recommendation, and many strategies considering one or more factors were proposed, such as the temporal factor, geographical factor, social factor, emotion factor, the popularity of POI, and so on. Although many preference mining methods and time-aware POI recommendation strategies have been proposed, the precision of these strategies is still not high.

To approach the challenges mentioned above, we intend to propose a fine-granularity POI recommendation strategy in the paper. The main contributions are concluded as follows:

• Based on users comments, we first extract fine-grained topics and generate topic probability distributions of all POIs using the latent dirichlet allocation (LDA) topic model, which is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. Then, the check-in information of each user is divided into multiple fine-grained data slices in hours in a day, and connected with the topic distributions of the POIs, to generate potential time-aware topic preferences.

• We construct a 3-order tensor of user-topic-time, and use the Higher Order Singular Value Decomposition (HOSVD) algorithm to decompose the tensor, to get a dense tensor for POI recommendation.

2375-0324/18/$31.00 ©2018 IEEE
DOI 10.1109/MDM.2018.00028

• We construct experiments on a real dataset to verify that the proposed approach outperforms the state-of-the-art POI recommendation methods.

The remainder of the paper is organized as follows. In Section II, we propose a POI recommendation framework. Section III discusses the generation of time-aware user-topic distributions. In Section IV, the construction and decomposition of the 3-order tensor of User-Topic-Time are discussed in details. Section V is performance evaluation. The related works are introduced briefly in Section VI. We conclude the works in the end.

II. OVERLL RECOMMENDATION FRAMEWORK

In the paper, we intend to recommend the top-N POIs for the users at a given time point.

Problem Definition. Supposed $U = \{u_1, u_2, \cdots, u_i\}$ is the set of users, $L = \{l_1, l_2, \cdots, l_j\}$ is the set of POIs. For a given user $u_i \in U$ and a given time point m, we recommend a selected POI list $L_i^m = \{l_{i1}^m, l_{i2}^m, \cdots, l_{iN}^m\}$ to u_i .

The proposed POI recommendation framework is shown as Fig. 1, which consists of four steps:

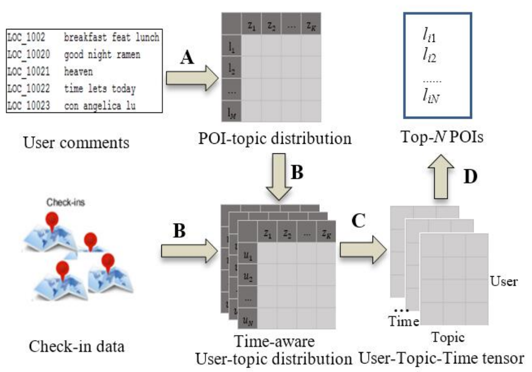


Fig. 1. The overall POI recommendation framework

Step A. Generating POI-Topic distribution. Based on the comments given by users, the topics and POI-topic distributions of all POIs are generated by the LDA (Latent Dirichlet Allocation) model.

Step B. Generating user-topic distribution. The user-topic distributions are generated by connecting the check-in data and the POI-topic distributions.

Step C. Constructing and decomposing tensor. A User-Topic -Time tensor, namely UZT, is constructed and decomposed, to obtain a dense User-Topic-Time tensor.

Step D. Recommending POIs. The user-topic distributions in each time slice are conversed into user-POI distributions for POI recommendation.

III. TIME-AWARE PREFERENCE MINING

In this section, the LDA model is used to extract the topics and topic distribution of each POI. Then, through connecting the check-in data and the POI-topic distributions, we generate the time-aware topic preferences of users.

A. Generating POI-Topic Distribution

The LDA model is a kind of language model, which uses the natural language model to identify the topic information hidden in the large text collections or corpus. In the model, each document is represented as a probability distribution of multiple topics, and each topic is represented as a probability distribution of multiple words. Thus, the model contains two hidden variables: the topic-word distribution Φ and the document-topic distribution Θ. Here, we use the LDA model to extract the topics of all POIs and generate the topic probability distributions.

At first, we aggregate all comments of each POI into a comment document. Then, we generate the topic distribution of each POI by means of the LDA model [4], which is shown in Fig.2. The parameters in the figure are explained as follows:

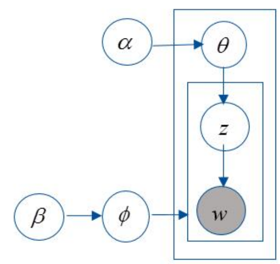


Fig. 2. LDA topic generation model

• α and β are the priori parameters on the corpus level. α represents the Dirichlet prior parameters of multinomial distribution over the topics in each document. β represents the Dirichlet prior parameters of multinomial distribution over the words in each topic.

• θ and φ are implicit variables. θ represents the multinomial distribution between the POIs and the topics, and φ_i represents the multinomial distribution between the topics and the words.

• ω is the explicit observed word vector and Z is the implicit topic vector. $z_{n,m}$ represents the n-th word in the m-th document, and $z_{n,m}$ represents the topic of the n-th word in the m-th document.

The generation process of POI-topic distributions based on the LDA model is summarized as Algorithm 1. Two matrices can be generated by the LDA model:

1) The topic-word probability matrix Φ_{K*V}: K represents the number of topics, V represents the number of words which don't repeat in the document D, and the vector φ_i in Φ_{K*V} represents the probability distribution over the words of the i-th topic.
2) The POI-topic probability matrix Θ_{M*K}: M represents the number of POIs, and the vector θ_i represents the probability distribution over the topic of the i-th POI.

Algorithm 1 POI-topic distribution generation

Input: K, D, Word corpus; α, β
Output: Φ, Θ, z
// topic plate
for all topic $k \in [1, K]$ **do**
  Sample mixture components $\phi_k \sim Dir(\beta)$
end for
// document plate
for all documents $m \in [1, M]$ **do**
  Sample mixture proportion $\theta_m \sim Dir(\alpha)$
  // word plate
  for all words $n \in [1, N_m]$in document m **do**
    Sample topic index $Z_{m,n} \sim Mult(\theta_m)$
    Sample word index $W_{m,n} \sim Mult(\phi_{Z_{m,n}})$
  end for
end for

So the unknown variables θ and φ can be obtained by Equation (1).

$$p(W, Z, \Theta, \Phi \mid \alpha, \beta) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} p(w_{m,n} \mid \phi_{z_{m,n}}) \cdot p(Z_{m,n} \mid \theta_m \mid \alpha) \cdot p(\Phi \mid \beta), \tag{1}$$

where the Gibbs sampling algorithm is employed to learn the parameters.

We take two POIs, i.e. l_{2612} and l_{2681}, in our dataset as examples. Their topic distributions are obtained by the LDA model as shown in Table I.

TABLE I: The example of POI-topic distributions

POI	Topic distributions
l_{2612}	$(z_1, 0.09318514444939833), (z_{10}, 0.1897591149590416),$ $(z_{13}, 0.5833369144379820), (z_{39}, 0.1274688261535182)$
l_{2681}	$(z_{20}, 0.70321428571430942), (z_{30}, 0.0732142857150930),$ $(z_{39}, 0.2169085644634494)$

As can be seen, l_{2612} have four topics, namely z_1, z_{10}, z_{13} and z_{39}, while l_{2681} has three topics, namely z_{20}, z_{30} and z_{39}. Every topic of a POI is associated with a probability, to represent the weight of the topic in the POI. For example, in l_{2612}, z_{13} has higher weight than other three topics.

For the LDA model, the time complexity of multiple iterations is $O(K * C * r)$, where K is the number of topics, C is the total number of words , r is the number of iterations.

B. Generating User-topic Distribution

The mining process of user-topic distribution is shown as Fig.3. That is, after generating POI-topic distribution (Fig.3(a)), we first divide user's historical check-in data into 24 time slices corresponding to the 24 hours of a day (Fig.3(b)), then connect them to generate the time-aware user-topic distribution(Fig.3(c)).

The procedure of the data connection includes four steps:

Step 1. Calculate the numbers that each user checked in each POI belonging to each topic in each time slice.

Step 2. Calculate the check-in ratio of per POI belonging to the specific topic in each time slice.

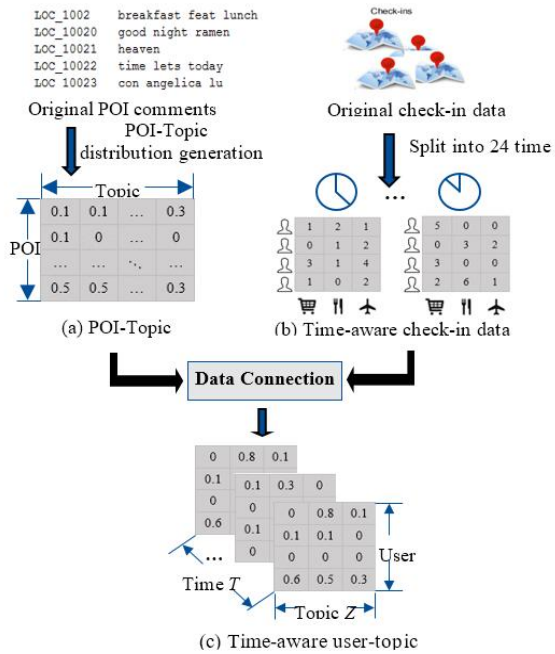


Fig. 3. The generation process of user-topic distribution

Let $p(z_j^{k,m}(u_i))$ be the check-in ratio that user u_i checked in the j-th POI with the k-th topic in the m-th time slice, which is calculated by Equation (2).

$$p(z_j^{k,m}(u_i)) = \frac{c_j^{k,m}(u_i)}{\sum_j^J c_j^{k,m}(u_i)}, \tag{2}$$

where $c_j^{k,m}(u_i)$represents the number that u_i checked in the j-th POI belonging to the k-th topic in the m-th time slice, and J represents the total number of POIs belonging to the topic in the time slice.

Taking u_2 as an example, the data connection results are shown in Table II. In m_{18}, u_2 checked in POIs l_{2612} and l_{2681} belonging to topic z_{39} once and twice respectively. So the check-in ratios of the two POIs in m_{18} are 33.3% and 66.7% respectively.

TABLE II: The example of data connection

user	Time slice	topic	POI	Check-in number	Check-in ratio
u_2	m_{18}	z_{39}	l_{2612}	1	33.3%
			l_{2681}	2	66.7%

Step 3. Calculate the initial user-topic distribution. The initial topic distribution of a given user at a specific time slice is the sum of the products of the users check-in ratio in each POI and the topic distribution in the POI.

Let $Z^{k,m}(u_i)$ be the initial topic distribution of u_i in the k-th topic in the m-th time slice, which is calculated by Equation (3).

$$Z^{k,m}(u_i) = \sum_j^J Z_j^{k,m}(u_i) \cdot p(Z_j^{k,m}(u_i)), \tag{3}$$

where $Z_j^{k,m}(u_i)$ is the distribution of the j-th POI with the k-th topic that u_i checked in the m-th time slice.

According to Equation (3), the initial user-topic distribution of u_2 on z_{39} in t_{18} is calculated as Equation (4).

$$\begin{aligned} Z^{39,18}(u_2) &= 0.12746883 \times 33.3\% + 0.21690856 \times 66.6\% \\ &= 0.18709534 \end{aligned} \tag{4}$$

The initial user-topic distribution, obtained by Equation (3), only reflects users check-in preferences for one topic at the specific time slice. However, it doesn't consider the check-in information on the other topics in the same time slice. In order to guarantee that we can compare the distribution of all the topics at the same time slice, the initial user-topic distribution should be standardized.

Let $\delta^{k,m}(u_i)$ be the standardized factor, representing the ratio of the total number that u_i checked in the k-th topic in the m-th time slice and the total check-in number of all topics in the same time slice. It is calculated by Equation (5).

$$\delta^{k,m}(u_i) = \frac{C^{k,m}(u_i)}{\sum_k^K C^{k,m}(u_i)}. \tag{5}$$

Step 4. Calculate the standardized user-topic distribution.

Let $\hat{Z}^{k,m}(u_i)$ be the standardized user-topic distribution, which is calculated by Equation (6).

$$\hat{Z}^{k,m}(u_i) = Z^{k,m}(u_i) \cdot \delta^{k,m}(u_i). \tag{6}$$

For user u_2, the standardized user-topic distributions are list in Table III. In t_{18}, u_2 checked in three topics z_{39}, z_{51} and z_{60}. The sum of the total check-in number is 6, so the standardized factors of each topic are $1/2$, $1/6$ and $1/3$, respectively. Thus, each standardized user-topic distribution of one topic in the time slice can be obtained by calculating the product of the standardized factor and the initial user-topic distribution of the topic.

TABLE III: The example of standardized user-topic distributions

user	Time slice	Topic	Check-in number	initial user-topic distribution	standardized factor	standardized user-topic distributions
u_2	t_{18}	z_{39}	3	0.18709534	1/2	0.093547659
		z_{51}	1	0.31524682	1/6	0.052541137
		z_{60}	2	0.21683426	1/3	0.072278087

IV. TENSOR DECOMPOSITION

Since the check-in data is sparse, the time-aware user-topic distributions are also sparse. This section intends to adopt the High-Order Singular Value Decomposition (HOSVD) approach to decompose the tensor constructed by the user-topic distribution.

A. UZT Tensor Decomposition

The user-topic distributions in different time slices can form a 3-order tensor. Thus, we construct a 3-order tensor of User-Topic-Time, named as UZT: $Y \in R^{N \times K \times O}$, where N is the number of users, K is the number of topics, O is the number of time slices (Fig. 4).

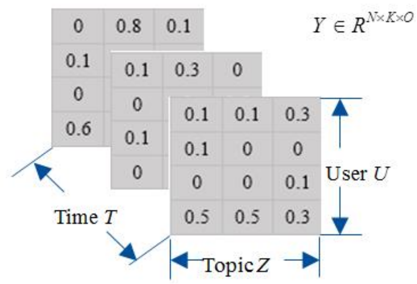


Fig. 4. The original UZT tensor

Each element $Y_{ikm} \in Y$ is the standardized topic distribution of u_i on the k-th topic in the m-th time slice, i.e.,

$$Y_{ikm} = \hat{Z}^{k,m}(u_i). \tag{7}$$

Since Y is a sparse tensor, it cannot reflect users preference on all topics. Thus, the HOSDV approach is used to decompose Y into a dense 3-order tensor. The steps are summarized as follows.

Step 1. Decompose the UZT into three factor matrices: $U \in R^{N \times d_U}$, $Z \in R^{k \times d_Z}$ and $T \in R^{O \times d_T}$,

Step 2. Construct a core tensor $G \in R^{d_U \times d_Z \times d_T}$ for controlling the interaction of the three matrices,

Step 3. Calculate the product of the core tensor and the factor matrices, and construct a new 3-order tensor $\hat{Y} \in R^{N \times K \times O}$ as follows.

$$\hat{Y} = G \times_U U \times_Z Z \times_T T, \tag{8}$$

where '$\times_U$' indicates that the tensor is multiplied by the matrix according to U-mode expansion, and the subscript U indicates the direction of the tensor multiplied by the matrix. '$\times_Z$' and '$\times_T$' have the same explanation.

B. Model Training

To get the optimal solution of the tensor, the objective function $L(\hat{y}, Y)$ is defined as Equation (9).

$$L(\hat{y}, Y) := \min_{U,Z,T} \frac{1}{2}\|\hat{Y} - Y\|_F^2 = \min \sum_{i,k,m} l(\hat{Y}_{ikm}, Y_{ikm}), \tag{9}$$

where $l(\hat{Y}_{ikm}, Y_{ikm})$ is the loss function, which can be calculated by Equation (10).

$$L(\hat{y}_{ikm}, Y_{ikm}) := \frac{1}{2}\|\hat{Y}_{ikm} - Y_{ikm}\|_F^2. \tag{10}$$

In order to avoid over-fitting, the regularization associated with U, Z, T and G is introduced into Equation (9), i.e., adding the normal term of the F-norm. Thus, the objective function is transformed into Equation (11).

$$\begin{aligned} L(\hat{y}, Y) :=& \min_{U,Z,T} \frac{1}{2}\|\hat{Y} - Y\|_F^2 \\ &+ \frac{\lambda}{2}(\|Y\|_F^2 + \|Z\|_F^2 + \|T\|_F^2) + \frac{\lambda_G}{2}\|G\|_F^2, \end{aligned} \tag{11}$$

where λ and λ_G are regularization parameters.

The online algorithm is utilized to iterate the factor matrices and the core tensor G at the same time, and the stochastic gradient descent (SGD) is applied to minimize the objective function. The iterative processes of the three factor matrices and the core tensor are listed in Equation (12) - (16):

$$\partial_{U_{i*}} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) G \times_Z Z_{k*} \times_T T_{m*}, \quad (12)$$

$$\partial_{Z_{k*}} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) G \times_U U_{i*} \times_T T_{m*}, \quad (13)$$

$$\partial_{T_{m*}} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) G \times_U U_{u*} \times_Z Z_{k*}, \quad (14)$$

$$\partial_{G} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) U_{i*} \otimes Z_{k*} \otimes T_{m*}, \quad (15)$$

$$\hat{Y}_{ikm} = G \times_U U_{i*} \times_Z Z_{k*} \times_T T_{m*}, \quad (16)$$

where U_{i*} represents the i-th line of U, M_{k*} represents the k-th line of M, and T_{m*} represents the m-th line of T, $\otimes$ represents the Khatri-Rao product of the matrix, i.e., the outer product of matrix. $\hat{Y}_{ikm}$ is a reconstructed dense 3-order tensor.

The tensor factorization algorithm of UTZ is summarized as Algorithm 2.

Algorithm 2 UZT decomposition

Input: Y
Output: U, Z ,T ,G
1: Initialize $U \in R^{N \times d_U}$,$Z \in R^{K \times d_Z}$,$T \in R^{O \times d_T}$ and $G \in R^{d_U \times d_Z \times d_T}$ with small random values
2: Set step size η
3: **while** (n, k, t) in observations **do**
4: $\quad \partial_{U_{i*}} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) G \times_Z Z_{k*} \times_T T_{m*}$
5: $\quad \partial_{Z_{k*}} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) G \times_U U_{i*} \times_T T_{m*}$
6: $\quad \partial_{T_{m*}} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) G \times_U U_{u*} \times_Z Z_{k*}$
7: $\quad \partial_{G} l(\hat{Y}_{ikm}, Y_{ikm}) = \partial_{\hat{Y}_{ikm}} l(\hat{Y}_{ikm}, Y_{ikm}) U_{i*} \otimes Z_{k*} \otimes T_{m*}$
8: $\quad \hat{Y}_{ikm} = G \times_U U_{i*} \times_Z Z_{k*} \times_T T_{m*}$
9: **end while**

The time complexity of this algorithm is $O(d_U * d_Z * d_T * r)$, where d_U, d_Z and d_T are the dimensionalities of the factors U, Z and T respectively, r is the number of iteration.

C. POI Recommendation

When doing recommendation, the users topic preferences should be converted into users POI preferences. According to the POI-topic distributions and the decomposed User-Topic-Time tensor, we can calculate the user-POI distributions.

Let $P = \{p_1, p_2, \cdots, p_n\}$ be a K dimensional vector to represent the users interest score in a specific period of time for all the topics, which $p_k (k = 1, 2, \cdots, K)$ indicates users final interest preference value for the k-th topic. Let Q be an $M \times K$ dimensional matrix to express the POI-topic distribution, in which q_{mk} represents the proportion of the k-th topic in the m-th POI.

Multiplying P by the transposed matrix of Q, we can get an M dimensional vector F, namely user preferences on all POIs at the m-th time slice, i.e.,

$$F = P^k(u_i) \times Q^T. \quad (17)$$

The element $f_j \in F$ indicates the u_i's preference on the j-th POI l_j at the m-th time, i.e.,

$$f_j = p_k \times q_{mk}. \quad (18)$$

Thus, we can select the N POIs with higher preference values to recommend to u_i.

V. PERFORMACE EVALUATION

In this section, we evaluate the performance of the proposed strategy, denoted as UZT.

A. Experiment Setup

Dataset. We conduct our experiment on the real dataset from the WW(World-Wide). The dataset is a check-in document, where each line has 5 attributes: user-ID, POI-ID, POI latitude and longitude, registration time and comments. The dataset contains a total of 74938 records, which has 3883 users and 49357 POIs. The check-in time is from November 1, 2012 to February 13, 2013.

Evaluation Metrics. We use three metrics, namely precision, recall rate and mean average precision, denoted as $Pre@N(t)$, $Rec@N(t)$ and $MAP(t)$, respectively, where N is the number of top-ranked POI recommendations, t is the time slice. They are calculated as Equations(19) $\sim$ (21), respectively.

$$Pre@N(t) = \frac{\sum_{u \in U} \mid Top\_N(u) \cap L(u) \mid}{\sum_{u \in U} \mid Top\_N(u) \mid}, \quad (19)$$

$$Rec@N(t) = \frac{\sum_{u \in U} \mid Top\_N(u) \cap L(u) \mid}{\sum_{u \in U} \mid L(u) \mid}, \quad (20)$$

$$MAP(t) = \frac{\sum_{n=1}^{N} loc_n(Top\_N(u) \cap L(u))/n}{\mid U \mid}, \quad (21)$$

where $Top\_N(u)$ represents the Top-N recommendation list obtained by the algorithm; $L(u)$ represents the POI list that user has visited; $loc_n(Top\_N(u) \cap L(u))$ represents the location of the n-th POI in the Top-N list at the correct recommended list. Note that, when a POI is in the Top-N list but not in the correct recommendation list, the values of $loc_n(Top\_N(u) \cap L(u))$ is 0.

Baseline methods. The methods of TLA[4], CMF[9], and ULT[14] are selected to be the baselines.

Liu et al. in [4] applied the LDA method to extract the topic distributions of POIs and interest distributions of users, and mapped them to obtain user interest preferences.

Gao et al. in [9] emphasized two temporal properties of user daily check-in preferences: non-uniformness and consecutiveness. It divided the check-in information into 24 matrices by hours and got the cosine similarity of users preference on consecutive temporal state.

Yao et al. in [14] constituted a 3-order tensor including user, location and time based on historical check-in information to recommend POIs. The method could cope with sparse data better through tensor decomposition, but they only captured user preferences by the check-in data.

B. Experiment Results and Analysis

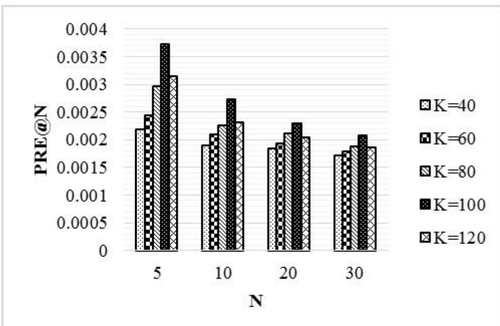
1) The influence of the number of topics

In LDA model, the topic structure is directly affected by the topic number K. In our experiment, we take different numbers of topics (40, 60, 80, 100, 120) to test the performance of the UZT method. The results are shown in Fig.5. It can be seen, when K is 100, the precision rate, recall rate and MAP are higher than other values of K. Thus, K is set to 100 in our experiences.

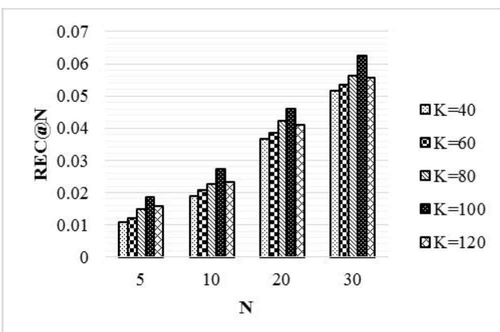
2)The influence of the rank of the tensor

In tensor decomposition, the rank of the core tensor determines the number of potential features. It is proven it is a NP hard problem [25]. So the experiment sets different values to obtain the optimal value of the rank.

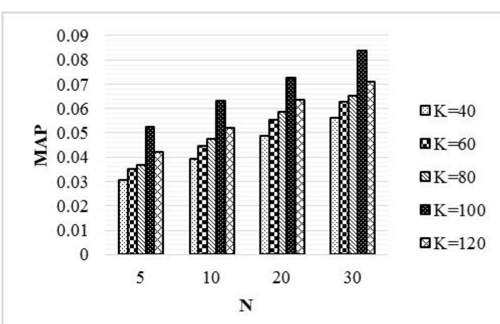
The dimension of the original tensor is 3883*24*100. We set the rank to 20%, 40%, 60%, 80%, 100% of the dimension, respectively. The results are shown in Fig.6. It can be seen, the performance of the dimension with 60% is better than the others. Thus, the rank is set to 60% of the dimension of the original tensor in our experiences.



(a) Precision



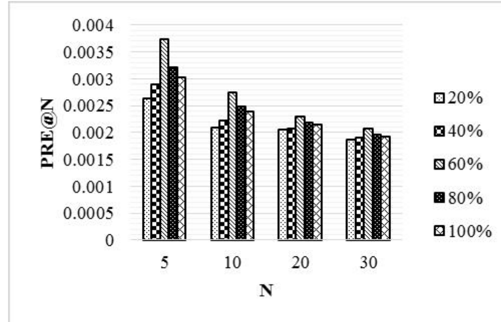
(b) Recall rate



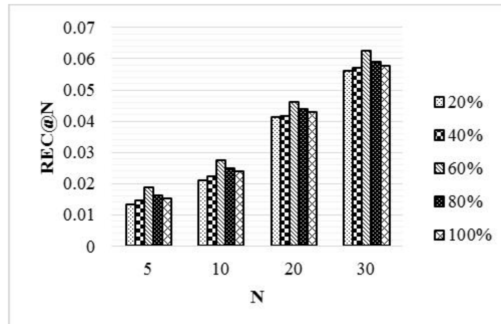
(c) MAP

Fig. 5. The influence of the number of topics on UZT performance

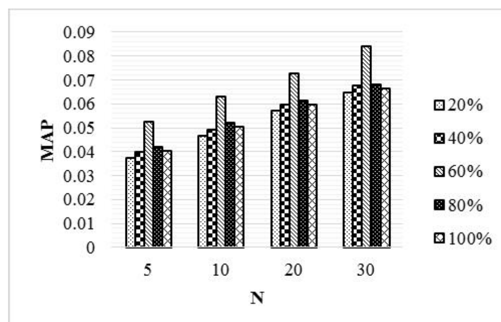
3) Performance comparison of different recommendation methods



(a) Precision



(b) Recall rate



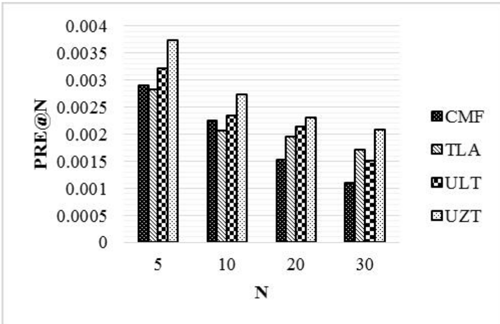
(c) MAP

Fig. 6. The influence of the rank of the tensor on UZT performance

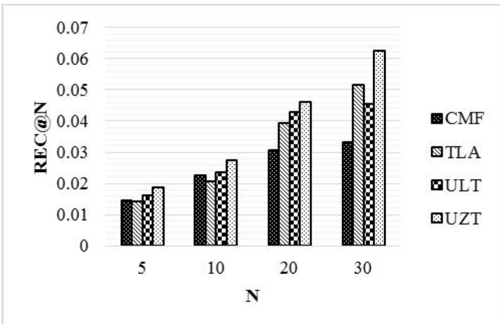
Based on the above experimental parameters, we compare the performance of the UZT method with the selected three baselines. The experimental results are shown in Fig.7.

It can be found, the performance results of UZT are better than the others. That is because UZT generates the fine-grained time-aware user preferences exploiting the check-in data and user comments simultaneously, and uses tensor decomposition method to solve the problem of data sparsity. However, the ULT algorithm only uses the check-in data for recommendation. Although the TLA method considers the user-topic distribution, but it doesn't take the temporal characteristic of user preferences into account. CMF can only deal with two-dimensional data, so the accuracy of use preference is lower than UZT.

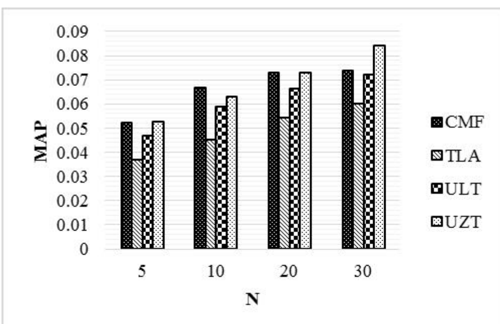
4) The influence of time granularity



(a) Precision



(b) Recall rate



(c) MAP

Fig. 7. Performance comparison of different recommendation methods

In the paper, the check-in data is divided into 24 data slices by the time granularity of 'the hours of a day'. We use different time granularities to compare the performance as follows:

• Because of the regularity of daily life, the check-in data is divided into 5 data slices by the time granularity of "section". The section consists of [6:00−10:00), [10:00−14:00), [14:00−18:00), [18:00−22:00), [22:00−2:00), [2:00−6:00).

• The check-in data is divided into 7 slices by the granularity of 'day of a week'. That is, every week is divided into 7 time periods, and one day represents one period.

• The check-in data is divided into 30 slices by the granularity of 'day of a month'. That is, every month is divided into 30 time periods, and one day represents one period.

Fig.8 show that the finer the time granularity is, the more accurate the recommendation results can be obtained.

VI. RELATED WORKS

In recent years, researchers pay more and more attentions on POI recommendation. In this section, we give a brief review on the related works.

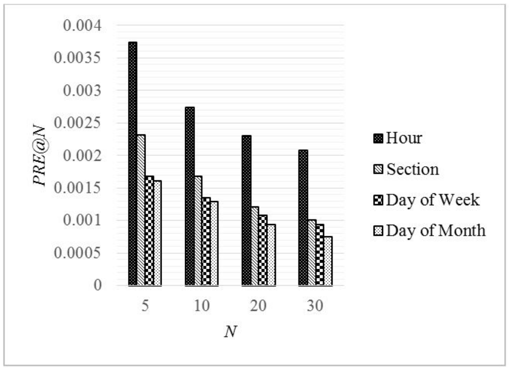
User preference mining. The users' interest preferences in LBSNs are mainly obtained from the POI check-in history or user comments. Liu et al. [4] applied the LDA method to extract the topic distributions of POIs and interest distributions of users, and mapped them to obtain user interest preferences. Gao et al. [5] fused the number of check-in and emotional intention into a matrix factorization model for POI recommend. Ference et al. [6] developed a collaborative filtering method exploiting geographical influence which is based on the user comments. Gao et al. [7] proposed a Hierarchical Pitman-Yor (HPY) language model to deal with the historical check-in information for inferring user preferences. These methods didnt take the characteristics of non-consistency of interests into account, that is, they assumed the interests of users are always the same at any time.

Time-aware recommendation. The temporal factor are getting more and more attentions in POI recommendation. Yuan et al. [8] first proposed a time-aware recommendation strategy, which added the temporal factor into user-based collaborative filtering strategy. The advantage is it can capture user interests at different time slices. Gao et al. [9] emphasized two temporal properties of users daily check-in preferences: non-uniformness and consecutiveness. It divided the check-in information into 24 matrices by hours and got the cosine similarity of users preference on consecutive temporal state. Yuan et al. [10] proposed a graph-based POI recommended method. It divided the original data by hours, and regarded POIs, sessions and users as three kinds of nodes and linked them together.

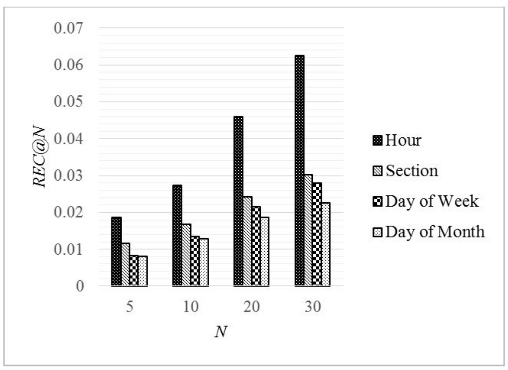
Considering data sparsity. In LBSNs, data sparsity becomes more serious than traditional item recommendation. Ye et al. [11] utilized traditional memory-based collaborative filtering method to recommend POIs, which depends on common check-in data to compare the similarities between users and POIs. Nevertheless, the check-in data shared by different users or POIs is little in fact.

Liu et al. [12] used the non-negative Bayesian matrix factorization method to combine the location information and text content for recommendation. It can handle the check-in data with zero and non-zero values. But it failed to cope with the missing value.

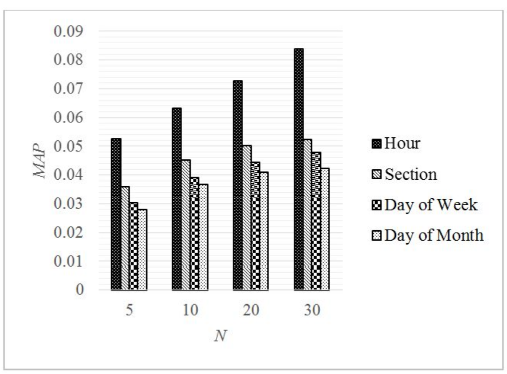
Cheng et al. [13] constructed a Multi-center Gaussian model to deal with the effect of geographical locations, and use the matrix factorization model for recommendation. But the method cannot handle the high-dimensional data. Yao et al. [14] constituted a 3-order tensor including user, location and time based on historical check-in information to recommend POIs. The method could cope with sparse data



(a) Precision



(b) Recall rate



(c) MAP

Fig. 8. Performance comparison on different time granularity

better through tensor decomposition, and they only captured users preference by the check-in data.

In addition, some POI recommendation methods considering different factors were proposed, such as geographical locations, social relations, user emotion, the popularity of POI, and so on [15]–[22].

VII. CONCLUSIONS

In this paper, a fine-granularity POI recommendation strategy for the location-based social networks is proposed. The strategy divides time within a day into "hour" granularity (i.e., 24 time slices), and POI into "topic" granularity, to get more accurate user preferences for POI recommendation.

The main work of this paper includes two parts: mine users potential time-aware topic preferences and decompose the tensor of User-Topic-Time for solving the problem of data sparsity. The experiments on the real dataset show that the proposed approach outperforms the state-of-the-art POI recommendation methods.

In the future,we will study the continuous time-aware strategy to improve the performance of POI recommendation further.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61262009), the Jiangxi Natural Science Foundation (No. 20151122040083), and the Jiangxi Foundation of Superiority Science and Technology Innovation Team Building Program (No. 20113BCB24008).

REFERENCES

[1] Bobadilla J, Ortega F, Hernando A, et al. *Recommender systems survey* [J]. Knowledge-Based Systems, 2013, 46(1):109-132.

[2] Gao H, Liu H. *Data Analysis on Location-Based Social Networks* [M]// Mobile Social Networking. Springer New York, 2014:165-194.

[3] Bao J, Zheng Y, Wilkie D, et al. *Recommendations in location-based social networks: a survey* [J]. GeoInformatica, 2015, 19(3):525-565.

[4] Gao H, Tang J, Hu X, et al. *Content-Aware Point of Interest Recommendation on Location-Based Social Networks*[C] //Proceedings of the 29th International AAAI Conference. AAAI, 2015: 1721-1727.

[5] Liu B, Xiong H. *Point-of-Interest Recommendation in Location Based Social Networks with Topic and Location Awareness*[C] //Proceedings of SIAM International Conference on Data Mining. SIAM, 2013: 396-404.

[6] Ference G, Ye M, Lee W C. *Location recommendation for out-of-town users in location-based social networks*[C]//Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, 2013: 721-726.

[7] Gao H, Tang J, Liu H. *Exploring Social-Historical Ties on Location-Based Social Networks*[C] //Proceedings the 6th International AAAI Conference on Weblogs and Social Media. AAAI, 2012:114-121.

[8] Yuan Q, Cong G, Ma Z, et al. *Time-aware Point-of-interest recommendation*[C] //Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2013: 363-372.

[9] Gao H, Tang J, Hu X, et al. *Exploring temporal effects for location recommendation on location-based social networks*[C] //Proceedings of the 7th International ACM Conference on Recommender Systems. ACM, 2013: 93-100.

[10] Yuan Q, Cong G, Sun A. *Graph-based Point-of-interest recommendation with geographical and temporal influences*[C] //Proceedings of the 23rd International ACM Conference on Conference on Information and Knowledge Management. ACM, 2014: 659-668.

[11] Ye M, Yin P, Lee W C, et al. *Exploiting geographical influence for collaborative Point-of-interest recommendation*[C] //Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2011: 325-334.

[12] Liu B, Fu Y, Yao Z, et al. *Learning geographical preferences for Point-of-interest recommendation*[C] //Proceedings of the 19th International ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2013: 1043-1051.

[13] Cheng C, Yang H, King I, et al. *Fused matrix factorization with geographical and social influence in location-based social networks*[C] //Proceedings of the 26th International AAAI Conference on Artificial Intelligence. AAAI, 2012: 17-23.

[14] Yao L, Sheng Q Z, Qin Y, et al. *Context-aware Point-of-interest recommendation using tensor factorization with social regularization*[C] //Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015: 1007-1010.

[15] Bao J, Zheng Y, Mokbel M F. *Location-based and preference-aware recommendation using sparse geo-social networking data*[C] //Proceedings of the 20th International Conference on Advances in Geographic Information Systems. ACM, 2012: 199-208.

[16] Noulas A, Scellato S, Lathia N, et al. *Mining user mobility features for next place prediction in location-based services*[C] //Proceedings of the 12th International Conference on Data Mining. IEEE, 2012: 1038-1043.

[17] Noulas A, Scellato S, Mascolo C, et al. *An Empirical Study of Geographic User Activity Patterns in Foursquare*[C] // Proceedings of 5th International AAAI Conference on Weblogs and Social Media (ICWSM). AAAI, 2011: 570-573.

[18] Li X, Cong G, Li X L, et al. *Rank-GeoFM: A ranking based geographical factorization method for Point of interest recommendation*[C] //Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015: 433-442.

[19] Gao H, Tang J, Hu X, et al. *Modeling temporal effects of human mobile behavior on location-based social networks*[C] //Proceedings of the 22nd International ACM Conference on Information & Knowledge Management. ACM, 2013: 1673-1678.

[20] Ifada N, Nayak R. *Tensor-based item recommendation using probabilistic ranking in social tagging systems*[C] //Proceedings of the 23rd International Conference on World Wide Web. ACM, 2014: 805-810.

[21] Yuan Q, Cong G, Ma Z, et al. *Who, where, when and what: discover spatio-temporal topics for twitter users*[C] //Proceedings of the 19th International ACM SIGKDD Conference on Knowledge Discovery and Data Mining. ACM, 2013: 605-613.

[22] Liu B, Xiong H, Papadimitriou S, et al. *A general geographical probabilistic factor model for Point of interest recommendation* [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5): 1167-1179.

[23] Cao Jiuxin, Dong Yi and Yang Pengwei, *POI recommendation based on meta-path in LBSN* [J]. Chinese Journal of Computers, 2016, 39(4): 676-684 (in Chinese)

[24] Cao Juan, Zhang Yongdong and Li Jintao, *A method of adaptively selecting best LDA model based on density*, Chinese Journal of Computers, 2008, 31(10):1780-1787

[25] Anandkumar A, Ge R and Hsu D, *Tensor decompositions for learning latent variable models*, Journal of Machine Learning Research, 2014, 15(1): 2773-2832