Explicable Location Prediction Based on Preference Tensor Model

Duoduo Zhang, Ning Yang(✉), and Yuchi Ma

College of Computer Science, Sichuan University, Chengdu, China
cherry.scu@gmail.com, yangning@scu.edu.cn, scuRichard.Ma@gmail.com

Abstract. Location prediction has been attracting an increasing interest from the data mining community. In real world, however, to provide more targeted and more personal services, the applications like location-aware advertising and route recommendation are interested not only in the predicted location but its explanation as well. In this paper, we investigate the problem of Explicable Location Prediction (ELP) from LBSN data, which is not easy due to the challenges of the complexity of human mobility motivation and data sparsity. In this paper, we propose a Preference Tensor Model (PTM) to address the challenges. The core component of PTM is a preference tensor, each cell of which represents how much a user prefers to a specific place at a specific time point. The explicable location prediction can be made via a retrieval of the preference tensor, and meanwhile a motivation vector is generated as the explanation of the prediction. To model the complicated motivations of human movement, we propose two motivation tensors, a social tensor and a personal tensor, to represent the social cause and the personal cause of human movement. From the motivation tensors, the motivation vector consisting of a social ingredient and a personal ingredient can be produced. To deal with data sparsity, we propose a Social Tensor Decomposition Algorithm (STDA) and a Personal Tensor Decomposition Algorithm (PTDA), which are able to fill missing values of a sparse social tensor and a sparse personal tensor, respectively. Particularly, to achieve a higher accuracy, STDA fuses an additional social constraint with the decomposition. The experiments conducted on real-world datasets verify the proposed model and algorithms.

Keywords: Location prediction · Tensor Model · Location Based Social Network

1 Introduction

Recently, location prediction based on check-in data of Location-Based Social Networks (LBSNs) has attracted increasing attention from the community of

This work was supported by the National Science Foundation of China under Grant 61173099 and the Sci. & Tech. Support Programs of Sichuan Province under Grant 2014JY0220.

B. Cui et al. (Eds.): WAIM 2016, Part I, LNCS 9658, pp. 205–216, 2016.
DOI: 10.1007/978-3-319-39937-9_16

data mining, due to its benefit for location-based applications such as location-aware advertising and marketing [1,10] and epidemic control [5] etc. Although a few techniques are proposed for the prediction of a future check-in location, they often focus on seeking a higher prediction accuracy while little attention is paid to the motivations behind human movements [3,11,14]. However, explicable location prediction is of great value to location-based applications, as it can help the applications provide more targeted services and personal experience to users. In this paper, we investigate the problem of the Explicable Location Prediction (ELP) from check-in data, which is not easy due to the following challenges:

– **Complicated motivations of human movement.** Existing studies [3,4,7,9] show that human mobility is constrained by social relationships and periodic behaviors. For example, shopping can be caused by social reasons, e.g. people want to stay longer with friends, or personal reasons, e.g. people want to buy necessities for daily life, or a mix of both. We need a way to quantitatively represent a combination of multiple motivations, so as to get a better understanding of why a user moves to a location.
– **Data sparsity.** In social networks, users' check-ins are always discontinuous because they only check in at the places they are interested, which results in that for a specific user, check-in data is sparse on both space and time dimensions.

In this paper, we propose a Preference Tensor Model (PTM) for the ELP. The main idea of PTM is inspired by the observation that human movement is not random but driven and explained by a mix of social motivation and personal motivation [12]. The core component of PTM is a preference tensor of a given user, where each cell stores the probability of that user will go to a specific place at a specific time point. An ELP can be made through a retrieval from the preference tensor, as well as a motivation vector as the explanation of the prediction can be generated from two motivation tensors, a social tensor and a personal tensor which are proposed to evaluate the social motivation and personal motivation of a human movement, respectively. To address the challenge of data sparsity, we employ a reconstruction strategy for the filling of missing values. We propose a Social Tensor Decomposition Algorithm (STDA) and a Personal Tensor Decomposition Algorithm (PTDA), which are able to fill missing values of a sparse social tensor and a sparse personal tensor, respectively. Particularly, to achieve a higher accuracy, STDA fuses an additional social constraint with the decomposition.

The main contributions of this paper can be summarized as follows:

(1) We propose a Preference Tensor Model (PTM) for the explicable location prediction. By PTM, a location prediction can be made with a motivation vector as its quantitative explanation.
(2) To model the complicated motivation of a human movement, we propose two motivation tensors, a social tensor and a personal tensor. From the motivation tensors, the mixed motivation vector can be produced as the explanation of a location prediction.

(3) To fill the missing values of a sparse social tensor and a sparse personal tensor, we propose a Social Tensor Decomposition Algorithm (STDA) and and a Personal Tensor Decomposition Algorithm (PTDA). Particularly, to achieve a higher accuracy, STDA fuses an additional social constraint with the decomposition.

(4) Experimental results show that the proposed PTM outperforms the baseline methods.

The rest of the paper is organized as follows. The details of PTM are described in Sect. 2. Motivation tensors are defined in Sect. 3. Data sparsity is addressed in Sect. 4. Experimental results are presented in Sect. 5. We give a brief description of the related work in Sect. 6 and conclude in Sect. 7.

2 Preference Tensor Model

We represent an Explicable Location Prediction (ELP) as a tuple $(r, \boldsymbol{v})$, where r is the predicted location, and $\boldsymbol{v}$ is the motivation vector. The motivation vector plays a role of a quantitative explanation of the location prediction, which is defined as follow:

Definition 1. *Motivation Vector.* *A motivation vector is a vector* $\boldsymbol{v} = (w^s, w^p)$*, where* w^s *and* w^p *represents corresponding weights of social and personal motivations of a check-in.*

Now we describe the details of our PTM, and how an ELP can be made by PTM. At first, for the location prediction, we propose a Preference Tensor, to represent the affinity between a user u, a time slot t, and a location r. Specifically, a preference tensor is a 3-dimensional tensor $\mathcal{T} \in \mathbb{R}^{M \times N \times Q}$, where M, N and Q are the numbers of users, time slots, and locations, respectively. A cell $\mathcal{T}_{u,t,r}$ stores the probability of user u ($u \in \{1, \cdots, M\}$) will go to location r ($r \in \{1, \cdots, N\}$) at time slot t ($t \in \{1, \cdots, Q\}$). We divide $\mathcal{T}_{u,t,r}$ as a sum of a social term and a personal term, i.e.,

$$\mathcal{T}_{u,t,r} = \mathcal{T}^s_{u,t,r} + \mathcal{T}^p_{u,t,r}, \tag{1}$$

where $\mathcal{T}^s \in \mathbb{R}^{M \times N \times Q}$ is a social tensor and $\mathcal{T}^p \in \mathbb{R}^{M \times N \times Q}$ is a personal tensor.

The social tensor and personal tensor are two motivation tensors. We use social tensor $\mathcal{T}^s$ to represent the social ingredient of the motivation of human movement, where a cell $\mathcal{T}^s_{u,t,r}$ measures the preference for location r of user u at time slot t that can be explained under the social context of u . Similarly, the personal tensor $\mathcal{T}^p$ represents the personal ingredient of a motivation, where a cell $\mathcal{T}^p_{u,t,r}$ measures the preference for location r of user u at time slot t that can be explained by personal reasons of u. We will describe the details of how to build the social tensor and personal tensor in next section.

Once we establish the PTM (Eq. (1)), we can predict the location of user u at time t by the following equation:

$$\hat{r} = \underset{r}{\operatorname{argmax}} \boldsymbol{\mathcal{T}}_{u,t,r}, \tag{2}$$

where $\hat{r}$ is the predicted location for user u at time slot t. According to the definition of motivation vector, as the explanation of the location prediction, the motivation vector can be computed as $\boldsymbol{v} = (\frac{\boldsymbol{\mathcal{T}}^s_{u,t,\hat{r}}}{\boldsymbol{\mathcal{T}}_{u,t,\hat{r}}}, \frac{\boldsymbol{\mathcal{T}}^p_{u,t,\hat{r}}}{\boldsymbol{\mathcal{T}}_{u,t,\hat{r}}})$

3 Construction of Motivation Tensors

In this section, we describe how to construct the two motivation tensors, from a social check-in set and a personal check-in set, respectively. As tensor is a discrete structure, we need to discretize the space and time first. We partition the whole area covered by the total check-in data into grids, and use the cell number to represent the location of the check-in. We also partition one day into 24 time slots with a period of 1 h, and the time of a check-in is instead represented as the time slot number. Now we define the concepts of check-in, social check-in and personal check-in.

Definition 2. ***Check-in.*** *A check-in is a triple $Tri(u, t, l)$ representing a user u checks in at location l at time point t.*

A social check-in is a check-in caused by friendship reasons, which is defined as follow:

Definition 3. ***Social Check-in.*** *A check-in $Tri(u, t, l)$ is a social check-in, if one of u's friends checked in at the same location at time point t' and $t - t' \leq \delta$, where δ is a threshold given in advance, e.g. one week.*

Different from social check-ins, a personal check-in is caused by personal reasons, which lead to that a user checks in at a place where he/she rarely checks in together with his/her friends.

Definition 4. ***Personal Check-in.*** *A check-in $Tri(u, t, l)$ is a personal check-in, if one of u's friends checked in at l at time point t' and $t - t' > \epsilon$, or none of u's friends checked in at l before t, where ϵ is a threshold given in advance.*

Algorithm 1 fulfills the construction of the social tensor and the personal tensor in a straightforward way.

4 Dealing with Sparsity

4.1 Social Tensor Decomposition Algorithm for $\boldsymbol{\mathcal{T}}^s$

To address data sparsity of the social tensor, we propose a Social Tensor Decomposition Algorithm(STDA). Particularly, to achieve a higher accuracy of decomposition, STDA fuses a social constraint represented as a closeness matrix when decomposing $\boldsymbol{\mathcal{T}}^s$. The idea here is that if two users are close enough, they are more likely to share common interest and check in at the same locations. Based on this idea, we define the closeness matrix as follow:

Algorithm 1. Social Tensor and Personal Tensor Construction Algorithm

Input:
social check-in set $\boldsymbol{C}^s$, personal check-in set $\boldsymbol{C}^p$
Output:
social tensor $\boldsymbol{\mathcal{T}}^s \in \mathbb{R}^{M\times N\times Q}$, personal tensor $\boldsymbol{\mathcal{T}}^p \in \mathbb{R}^{M\times N\times Q}$

```
Initialize each cell of T^s and T^p with 0;
for each user i, time slot j, region k do
    n^s = the number of the occurrences of Tri(i, j, k) in C^s;
    n^p = the number of the occurrences of Tri(i, j, k) in C^p;
    N_i = the number of the check-ins of user i;
    T^s_{i,j,k} = n^s / N_i;
    T^p_{i,j,k} = n^p / N_i;
end for
```

Definition 5. ***Closeness Matrix.*** *Closeness Matrix, denoted as* $\boldsymbol{X}$*, where an entry denotes the closeness of user* i *to user* j*, and* $\boldsymbol{X}_{i,j} = \frac{|\boldsymbol{C}^s_{i,j}|}{|\boldsymbol{C}_i|}$*, where* $\boldsymbol{C}^s_{i,j}$ *is the social check-in set between user* i *and user* j*, and* $\boldsymbol{C}_i$ *is the total check-in set of user* i*.*

The objective function of STDA is defined as follow:

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{\mathcal{T}}^s, \boldsymbol{\mathcal{I}}, \mathbf{U}, \mathbf{C}, \mathbf{L}) &= \frac{1}{2}\|\boldsymbol{\mathcal{T}}^s - \boldsymbol{\mathcal{I}} \times_1 \mathbf{U} \times_2 \mathbf{C} \times_3 \mathbf{L}\|_F^2 \\ &+ \frac{\alpha}{2}\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 + \frac{\beta}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{C}\|_F^2 + \|\mathbf{L}\|_F^2 + \|\mathbf{V}\|_F^2), \end{aligned} \tag{3}$$

where $\|\cdot\|_F$ represents Frobenius norm; $\mathbf{U} \in \mathbb{R}^{M\times D}$, $\mathbf{C} \in \mathbb{R}^{N\times D}$ and $\mathbf{L} \in \mathbb{R}^{Q\times D}$ are latent factors, and D is the number of latent features; $\mathbf{V} \in \mathbb{R}^{M\times D}$ is a factor matrix of $\mathbf{X}$; $\times_i (1 \le i \le 3)$ stands for the tensor multiplication along the i-th mode; α and β are parameters controlling the contributions of different parts. The social constraint represented as $\mathbf{X}$, is imposed via $\mathbf{U}$, as it is shared between $\boldsymbol{\mathcal{T}}^s$ and $\mathbf{X}$. Equation (3) involves three terms, where the first and second terms are the errors of the matrix factorization and the tensor decomposition, and the last one is the regularization penalty. As there is no closed-form solution to the minimization of the objective function, we minimize Eq. (3) using gradient descent.

Algorithm 2 gives the procedures of STDA, where the gradients of the objective function can be computed as follows:

$$\begin{aligned} \partial_{\mathbf{U}_{i*}} F^l &= (\widetilde{\boldsymbol{\mathcal{T}}}^s_{i,j,k} - \boldsymbol{\mathcal{T}}^s_{i,j,k}) \times \boldsymbol{\mathcal{I}} \times_2 \mathbf{C}_{j*} \times_3 \mathbf{L}_{k*} + \alpha(\mathbf{U}_{i*}\mathbf{V}^T - \mathbf{X}_{i*}) + \beta\mathbf{U}_{i*}, \\ \partial_{\mathbf{C}_{j*}} F^l &= (\widetilde{\boldsymbol{\mathcal{T}}}^s_{i,j,k} - \boldsymbol{\mathcal{T}}^s_{i,j,k}) \times \boldsymbol{\mathcal{I}} \times_1 \mathbf{U}_{i*} \times_3 \mathbf{L}_{k*} + \beta\mathbf{C}_{j*}, \\ \partial_{\mathbf{L}_{k*}} F^l &= (\widetilde{\boldsymbol{\mathcal{T}}}^s_{i,j,k} - \boldsymbol{\mathcal{T}}^s_{i,j,k}) \times \boldsymbol{\mathcal{I}} \times_1 \mathbf{U}_{i*} \times_2 \mathbf{C}_{j*} + \beta\mathbf{L}_{k*}, \\ \partial_{\mathbf{V}} F^l &= (\mathbf{U}_{i*} \times \mathbf{V}^T - \mathbf{X}) \times \mathbf{U}_{i*} \end{aligned}$$

Algorithm 2. Social Tensor Decomposition Algorithm

Input:

tensor $\boldsymbol{\mathcal{T}}^s$, matrix $\mathbf{X}$, step size θ, parameters α, β,
the number of latent features D.

Output:

a filled social tensor $\widetilde{\boldsymbol{\mathcal{T}}}^s$

Initialize $\mathbf{U}$, $\mathbf{C}$, $\mathbf{L}$ and $\mathbf{V}$ with random values between 0 and 1;
Set $l = 0$;
Calculate the initial value of the cost function according to Eq. (3);
while not converged **do**
  for each $\boldsymbol{\mathcal{T}}^s_{i,j,k} \neq 0$ **do**
    $\mathbf{U}_{i*} = \mathbf{U}_{i*} - \theta \partial_{\mathbf{U}_{i*}} F^l$;
    $\mathbf{C}_{j*} = \mathbf{C}_{j*} - \theta \partial_{\mathbf{C}_{j*}} F^l$;
    $\mathbf{L}_{k*} = \mathbf{L}_{k*} - \theta \partial_{\mathbf{L}_{k*}} F^l$;
    $\mathbf{V} = \mathbf{V} - \theta \partial_{\mathbf{V}} F^l$;
    $++l$;
    Calculate the l-th iteration of the cost function $\boldsymbol{F}^l$ according to Eq. (3);
  end for
end while
$\widetilde{\boldsymbol{\mathcal{T}}}^s = \boldsymbol{\mathcal{I}} \times_1 \mathbf{U} \times_2 \mathbf{C} \times_3 \mathbf{L}$;

4.2 Personal Tensor Decomposition Algorithm for $\boldsymbol{\mathcal{T}^p}$

The objective function of PTDA is defined as follow:

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{\mathcal{T}}^p, \boldsymbol{\mathcal{I}}, \mathbf{U}, \mathbf{C}, \mathbf{L}) = {} & \frac{1}{2}\|\boldsymbol{\mathcal{T}}^p - \boldsymbol{\mathcal{I}} \times_1 \mathbf{U} \times_2 \mathbf{C} \times_3 \mathbf{L}\|_F^2 \\ & + \frac{\beta}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{C}\|_F^2 + \|\mathbf{L}\|_F^2), \end{aligned} \tag{4}$$

where $\mathbf{U}$ ($\mathbf{U} \in \mathbb{R}^{M \times D}$), $\mathbf{C}$ ($\mathbf{C} \in \mathbb{R}^{N \times D}$) and $\mathbf{L}$ ($\mathbf{L} \in \mathbb{R}^{Q \times D}$) are latent factors, and D is the number of latent features. As shown in Algorithm 3, PTDA also employs the gradient descent to find an optimal solution to Eq. (4).

In Algorithm 3, the personal tensor can be estimated as $\widetilde{\boldsymbol{\mathcal{T}}}^p = \boldsymbol{\mathcal{I}} \times_1 \mathbf{U} \times_2 \mathbf{C} \times_3 \mathbf{L}$, and the gradients of the objective function can be given as follows:

$$\begin{aligned} \partial_{\mathbf{U}_{i*}} H^l &= (\widetilde{\boldsymbol{\mathcal{T}}}^p_{i,j,k} - \boldsymbol{\mathcal{T}}^p_{i,j,k}) \times \boldsymbol{\mathcal{I}} \times_2 \mathbf{C}_{j*} \times_3 \mathbf{L}_{k*} + \beta \mathbf{U}_{i*}, \\ \partial_{\mathbf{C}_{j*}} H^l &= (\widetilde{\boldsymbol{\mathcal{T}}}^p_{i,j,k} - \boldsymbol{\mathcal{T}}^p_{i,j,k}) \times \boldsymbol{\mathcal{I}} \times_1 \mathbf{U}_{i*} \times_3 \mathbf{L}_{k*} + \beta \mathbf{C}_{j*}, \\ \partial_{\mathbf{L}_{k*}} H^l &= (\widetilde{\boldsymbol{\mathcal{T}}}^p_{i,j,k} - \boldsymbol{\mathcal{T}}^p_{i,j,k}) \times \boldsymbol{\mathcal{I}} \times_1 \mathbf{U}_{i*} \times_2 \mathbf{C}_{j*} + \beta \mathbf{L}_{k*} \end{aligned}$$

5 Experiment

To verify the performance of PTM, we compare PTM with five baselines on a real-world dataset. All of the experiments are conducted on a PC with Intel Core I5 CPU 3.2 GHZ and 16 GB main memory. The operating system is Windows 7 and all the algorithms are implemented in C#.

Algorithm 3. Personal Tensor Decomposition Algorithm

Input:
tensor $\boldsymbol{\mathcal{T}}^p$, matrix $\mathbf{X}$, step size θ, parameter β, the number of latent features D.
Output:
a filled personal tensor $\widetilde{\boldsymbol{\mathcal{T}}}^p$

Initialize $\mathbf{U}$, $\mathbf{C}$ and $\mathbf{L}$ with random values between 0 and 1;
Set $l = 0$;
Calculate the initial value of the cost function according to Eq. (4);
while not converged **do**
  for each $\boldsymbol{\mathcal{T}}^p_{i,j,k} \neq 0$ **do**
    $\mathbf{U}_{i*} = \mathbf{U}_{i*} - \theta \partial_{\mathbf{U}_{i*}} H^l$;
    $\mathbf{C}_{j*} = \mathbf{C}_{j*} - \theta \partial_{\mathbf{C}_{j*}} H^l$;
    $\mathbf{L}_{k*} = \mathbf{L}_{k*} - \theta \partial_{\mathbf{L}_{k*}} H^l$;
    $++l$;
    Calculate the l-th iteration of the cost function $\boldsymbol{H}^l$ according to Eq. (4);
  end for
end while
$\widetilde{\boldsymbol{\mathcal{T}}}^p = \boldsymbol{\mathcal{I}} \times_1 \mathbf{U} \times_2 \mathbf{C} \times_3 \mathbf{L}$;

5.1 Setting

Dataset. The dataset we use in our experiments is collected from Gowalla[1], a famous LBSN. The dataset contains 6.4 million check-ins and an undirected social network consisting of 19.7 thousand users. We randomly split the dataset into training and testing sets with an 8:2 ratio.

Metrics. We use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate the prediction accuracy of PTM, which are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}, \tag{5}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n}, \tag{6}$$

where y_i and $\hat{y}_i$ are the true and estimated values respectively, and n is the number of samples in test set.

To evaluate the explanation made by PTM, we propose a new metric, Explicable Accuracy (EAcc), which is defined as

$$EAcc = \frac{\sum I_{u,t,l}}{n}, \tag{7}$$

where $I_{u,t,l} = 1$ if $\widetilde{\boldsymbol{\mathcal{T}}}^s_{u,t,l} \geq$ (or $<$) $\widetilde{\boldsymbol{\mathcal{T}}}^p_{u,t,l}$ and $\boldsymbol{\mathcal{T}}^s_{u,t,l} \geq$ (or $<$) $\boldsymbol{\mathcal{T}}^p_{u,t,l}$ on test set, otherwise, 0. Here the idea of EAcc is to use the nonzero cells (i.e., the cells with

[1] https://snap.stanford.edu/data/.

indices (u, t, l) such that $\mathcal{T}^s_{u,t,l} \neq 0$ and $\mathcal{T}^p_{u,t,l} \neq 0$) of the original motivation tensors as the ground truth of which motivation dominates a movement, and check whether the reconstructed values keep the size order between the old values, and if they do, $I_{u,t,l}$ equals 1, otherwise 0.

Baseline Approaches. We compare PTM with the following baselines.

MF (Most-Frequent). [3] **MF** predicting a location for a user u totally relies on u's history check-ins, which can be calculated as

$$P(u, t, l) = \frac{|Tri|Tri \in \boldsymbol{C_u}, Tri.t = t, Tri.l = l|}{|\boldsymbol{C_u}|}, \tag{8}$$

where $\boldsymbol{C}_u$ is the check-in set of user u.

PMM (Periodic Mobility Model). [3] PMM makes a location prediction for a user based on a time-independent gaussian distribution of the historical locations where the user appeared before.

W3 (Who + Where + When). [14] W3 is a variant of W4 (Who + Where + When), which is a probabilistic model integrating user, location, time and activity information and can be applied to many applications. W3 makes a location prediction by maximize the following posterior probability:

$$P(l|u, s, t) = \frac{\sum_z \sum_r p(u, s, t, r, z, l)}{\sum_z \sum_r \sum_{l'} p(u, s, t, r, z, l')} \tag{9}$$

where u, s, t, r, z, l denote user, workday or holiday, time, region, topic and location, respectively.

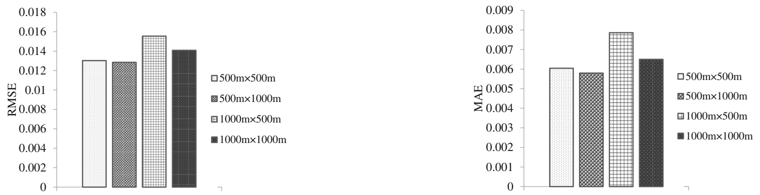
RCH (Regularity and Conformity). [11] RCH incorporates regularity and conformity of human mobility in a unified prediction model, and utilizes the interplay between the two factors. It makes a location prediction based on an estimated score for a specific location.

CATD (Context-Aware Tensor Decomposition). [2] CATD makes location predictions adopting a tensor reconstruction strategy. It constructs a sparse tensor using total check-in data and decompose it with additional context constraints collaboratively. The value of each cell in filled tensor can be considered as the probability that a user will check in at a specific region at a specific time point.

5.2 Determination of Parameters

Grid Resolution. Grid resolution in our experiments can affect the performance of PTM significantly. In our experiments, we partition the whole area covered by the total check-ins into grids based on the finding of [8]. Figure 1 shows the RMSE and MAE over different grid sizes. From Fig. 1, we can see that RMSE and MAE both reach the minimal when the grid resolution is $500\,\text{m} \times 1000\,\text{m}$.

The Number of Latent Features $\boldsymbol{D}$. We tune parameter D from 3 to 9 with respect to RMSE and MAE. The results are shown in Figs. 2(a) and 3(a), which suggest that the best choice of D is 4.



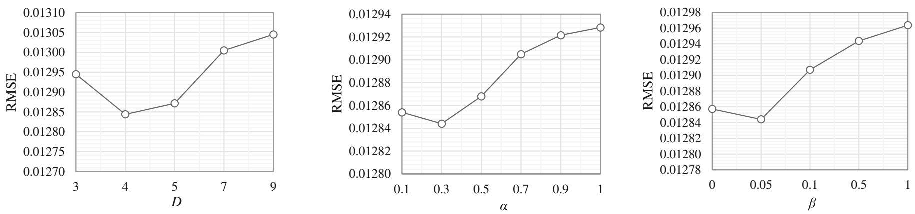
(a) RMSE variation (b) MAE variation

Fig. 1. RMSE and MAE over different grid resolutions

Contributions of Closeness Matrix and Regularization Penalty. We tune parameters α and β with respect to RMSE and MAE. The results for α are shown in Figs. 2(b) and 3(b), and the results for β shown in Figs. 2(c) and 3(c). The results suggest that $\alpha = 0.3$ and $\beta = 0.05$ is the best choices.

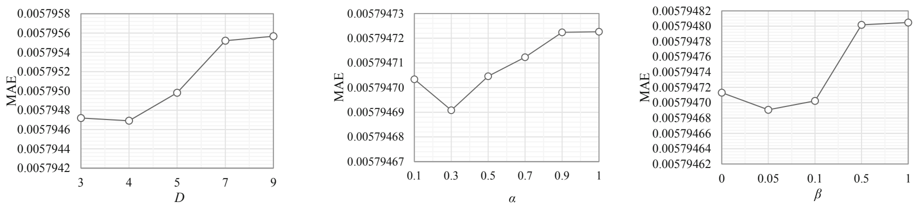
5.3 Prediction Accuracy

Figure 4 shows the RMSEs and MAEs of PTM and the baseline methods. From Fig. 4, we can see that PTM outperforms the baseline methods, which is because PTM (1) considers both the historical movements and the social relationships of users, (2) can handle the sparsity of data, (3) and can distinguish the motivations. For example, MF can not predict new locations for a user where the user never appeared before, since it only considers the history check-ins of that user.



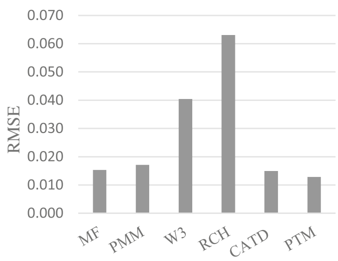
(a) (b) (c)

Fig. 2. RMSE over different D, α, β

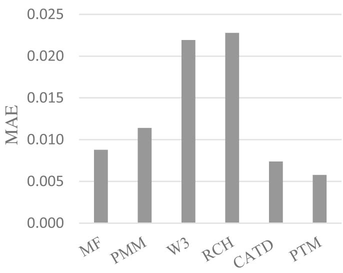


(a) (b) (c)

Fig. 3. MAE over different D, α, β



(a) RMSE



(b) MAE

Fig. 4. Accuracy comparison

The baseline methods except for CATD are designed for data that are not sparse, which makes their prediction accuracy lower than the proposed PTM on sparse data. We also note that PTM performs better than CATD, which is because CATD treats the check-ins equivalently without distinguishing social check-ins from personal ones.

5.4 Verification of Motivation Vector

To verify the effectiveness of the predication explanation, represented as motivation vectors, we compare PTM with a naive method, called Naive PTM (NPTM), with respect to EAcc. NPTM is a variant of PTM that fulfills the tensor decomposition without any constraints. Figure 5 shows the EAccs of PTM and NPTM. We can see that the EACC of PTM reaches about 91 %, which is significantly higher than the EAcc (61 %) of NPTM. We argue that this is because we fuse a closeness matrix as the social constraint to the reconstruction of a sparse social tensor.

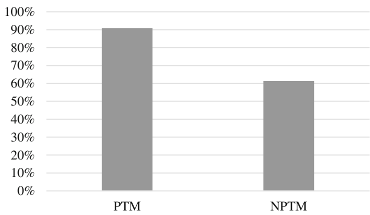


Fig. 5. EAcc comparison

6 Related Work

The existing methods for location prediction roughly fall into two categories, probabilistic model based methods and collaborative model based methods.

Probabilistic Model Based Methods. Yuan et al. [14] propose a probabilistic model W4 (short for who + where + when + what) which reveals the dependencies between user mobility and the spatial, temporal and activity factors. Cho et al. [3] propose a periodic mobility model (PMM) based on the intuition that the human mobility is periodic and often center around home and work office. Yang et al. [12] propose an approach considering both the periodicity and the sociality of human movements. However, these methods often assume that the data is not sparse, and there are sufficient check-ins to support the prediction, which makes them unable to serve our case where the data is sparse.

Collaborative Model Based Methods. Wang et al. [11] propose a hybrid model called RCH by combining both regularity and conformity of human mobility. Lian et al. [6] propose a collaborative exploration and periodically returning (CEPR) model which makes location prediction based on the periodicity of human mobility and the social relationships of users. Ye et al. [13] propose a unified POI recommendation framework which fuses user preferences with social features and geographic features of POIs. Zheng et al. [2] propose a Context-Aware Tensor Decomposition (CATD) model which can utilize additional heterogeneous data as context constraints to optimize the latent feature matrices of the tensor. However, the above methods can not quantitatively explain the motivations of a user movement.

7 Conclusions

In this paper, we propose a Preference Tensor Model (PTM) for the Explicable Location Prediction. PTM makes location prediction via a retrieval from a preference tensor, each cell of which represents how much a user prefers to a specific place at a specific time point. We propose two motivation tensors, a social tensor and a personal tensor, from which a motivation vector consisting of a social ingredient and a personal ingredient can be generated as the explanation of a location prediction. To deal with the data sparsity, we propose a Social Tensor Decomposition Algorithm (STDA) and a Personal Tensor Decomposition Algorithm (PTDA), which are able to reconstruct the sparse motivation tensors by filling the missing values of them. Particularly, to achieve a higher accuracy, STDA fuses an additional social constraint with the decomposition. At last, experimental results on a real-world dataset verify the performance of PTM.

References

1. Agarwal, A., Hosanagar, K., Smith, M.D.: Location, location, location: an analysis of profitability of position in online advertising markets. J. Mark. Res. **48**, 1057–1073 (2011)
2. Bhargava, P., Phan, T., Zhou, J., Lee, J.: Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 25–34 (2014)

3. Cho, E., Myers, S.A., Leskovec, J.: Friendship, mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1082–1090 (2011)
4. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. Nature **453**, 779–782 (2008)
5. Lan, L., Malbasa, V., Vucetic, S.: Spatial scan for disease mapping on a mobile population. In: Proceedings of the 28th Association for the Advancement of Artificial Intelligence, AAAI 2014, pp. 431–437 (2014)
6. Lian, D., Xie, X., Zheng, V.W., Yuan, N.J., Zhang, F., Chen, E.: CEPR: a collaborative exploration and periodically returning model for location prediction. ACM Trans. Intell. Syst. Technol. (TIST) **6**, 1–27 (2014)
7. Mok, D., Wellman, B., Carrasco, J.: Does distance matter in the age of the internet? Urban Stud. **47**, 2747–2783 (2010)
8. Noulas, A., Scellato, S., Mascolo, C., Pontil, M.: An empirical study of geographic user activity patterns in foursquare. Assoc. Adv. Artif. Intell. **11**, 570–573 (2011)
9. Song, C., Qu, Z., Blumm, N., Barabasi, A.L.: Limits of predictability in human mobility. Science **327**, 1018–1021 (2010)
10. Tan, Z.: Spatial advertisement competition: based on game theory. J. Appl. Math. (2014)
11. Wang, Y., Yuan, N.J., Lian, D., Xu, L., Xie, X., Chen, E., Rui, Y.: Regularity, conformity: Location prediction using heterogeneous mobility data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1275–1284 (2015)
12. Yang, N., Kong, X., Wang, F., Yu, P.S.: When and where: Predicting human movements based on socail spatial-temporal events. In: Proceedings of 2014 SIAM International Conference on Data Mining (SDM 2014), pp. 515–523 (2014)
13. Ye, M., Yin, P., Lee, W.-C.: Location recommendation for location-based social networks. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 458–461 (2010)
14. Yuan, Q., Cong, G., Ma, Z., Sun, A., Magnenat-Thalmann, N.: Who, where, when, what: discover spatio-temporal topics for twitter users. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 605–613 (2013)