Exploring The Interaction Effects for Temporal Spatial Behavior Prediction

Huan Yang*
School of Software, Shandong University
zhw1608163028@gmail.com

Tianyuan Liu*
School of Software, Shandong University
zodiacg@foxmail.com

Yuqing Sun†
School of Software, Shandong University
sun_yuqing@sdu.edu.cn

Elisa Bertino
Department of Computer Science, Purdue University
bertino@cs.purdue.edu

ABSTRACT

In location based services, predicting users' temporal-spatial behavior is critical for accurate recommendation. In this paper, we adopt a joint embedding (JointE) model to learn the representations of user, location, and users' action in the same latent space. The functionality of a location is the critical factor influencing different elements of the behavior and is learned by an embedding vector encoding crowd behaviors. A user personalized preference is learned from the user historical behaviors and has two features. One is the combination of action and location, which is learned by maximizing the semantic consistency of the observed behaviors. The other is the periodic preference. Inspired by the notion of periodical temporal rules, we introduce the concept of temporal pattern to describe how often users visit places so as to reduce the high temporal variance of behaviors. A projection matrix is introduced to combine the temporal patterns with location functionality. A user behavior is predicted by the joint probability on behavior elements. We conduct experiments against two representative datasets. The results show that our approach outperforms other approaches.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**; *Neural networks.*

KEYWORDS

behavior prediction; embedding; latent correlation

ACM Reference Format:
Huan Yang, Tianyuan Liu, Yuqing Sun, and Elisa Bertino. 2019. Exploring The Interaction Effects for Temporal Spatial Behavior Prediction. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357963

*Both authors contributed equally to this research.
†Corresponding author.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357963

1 INTRODUCTION

Location based services (LBS) are today used in many different application domains. LBS platforms provide users with opportunities for sharing point-of-interests (POIs), products, and comments. They greatly enhance user experience and help merchants to accurately target advertisements and recommend products. Predicting user temporal-spatial behavior is a fundamental task for accurate recommendations. According to the Revealed Preference Theory in Economics[1], a user behaviors reveal her inherent demands and the disposable budget like income or time. A user preferences actually are the combination on which places she likes to go, how often to be there and what actions taken there, rather than a single element of behavior. For an instance, two users both prefer western foods, but they may choose different restaurants due to the consumption level or go the same restaurant with different frequencies or with different menu choices. So it is necessary to learn not only the expected location of the user, but also the times at which the user will be at the location and the activities the user will do at the location at a certain time. A recommender system with such capability would be able to provide very accurate recommendation.

The works more closely related to this problem are the approaches for POIs recommendation on either a visit time or a location [5, 6, 14, 24]. But such approaches do not predict other elements of user behavior, such as the actions carried by the user, i.e. shopping or social activities etc., and thus are unable to predict these elements together. Other related works focus on user action prediction [9, 21]. Such approaches take into account the user purchase history as a user-item rating matrix and predict items that the user would buy by using matrix factorization [10, 20]. However, these methods are designed for online transactions and do not take the spatial factors into account, thus they are unable to predict the user temporal-spatial behavior. For example, in online to offline (O2O) applications, the location of a shop is an important factor for predicting whether the user would visit this shop.

In practice, however, it is much harder to jointly predict multiple behavioral elements than to predict a single behavior element as traditional recommendation methods do. The reason is that user behavior representations adopted by existing approaches provide very little semantics and thus it is difficult to correlate the various elements of a behavior. Users' action preferences also vary and are temporal-spatially specific. For example, an individual may go to a park once a week and go to her office every weekday, but only

[1]https://en.wikipedia.org/wiki/Revealed_preference

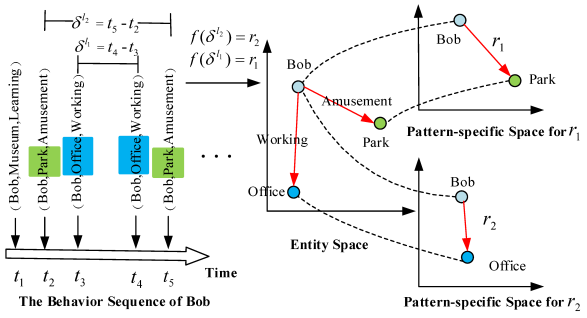


Figure 1: The Embedding Process on User Behavior Preference.

once a month to a museum. Even for the same user there is a high variance in his preference.

To address these challenges, we propose a joint embedding model to learn the correlated elements in user behaviors as latent vectors in the same space. Inspired by the Consumer Demand Theory[2], the location functionalities are regarded as the driven factor and learned by embedding vectors encoding crowd behaviors. Such vectors are used to join other behavior elements together. A user personalized preference is learned from the user historical behaviors encoded by the embeddings, and includes two parts as shown in Figure 1, where the left gives an example of *Bob's* behavior sequence and the right gives the model. The first part of model is the combination of action and location, which is represented by the expectation of action vectors at each location in the entity space and is learned by maximizing the semantic consistency of the observed behaviors. The second part is the periodic preference. Since user's behaviors have a high variance in time, it is hard to directly model the temporal elements of the behaviors in the form of time intervals. Inspired by the notion of temporal periodic rules, we introduce a novel scheme, based on temporal patterns, to represent how often a user visits a place. A projection matrix is introduced to combine the temporal pattern with the location functionality. Such an approach supports a unified representation of user preferences in the pattern-specific space. We conduct experiments on two real-world datasets to verify our proposed approach. The evaluation results show that our approach outperforms related state-of-the-art methods. *To the best of our knowledge, our approach is the first able to predict user behavior by combining multiple aspects together.*

The rest of this paper is organized as follows. Section 2 discusses related works. Section 3 and 4 introduce the notations and discusses the proposed model. Section 5 and 6 analyze the datasets and present the experimental results, respectively. Finally, we conclude this work.

2 RELATED WORKS

The work closely related to ours is the work on POI recommendations. Several approaches have been proposed to predict the location an individual would visit by learning from visit histories of a similar group of individuals. POI recommendation approaches are typically based on the collaborative filtering techniques [22]. Zheng *et al.* apply a collective matrix factorization method to mine interesting places and recommend them to the users [27, 28]. However, these prediction models do not take into account temporal information and thus are unable to accurately predict the time point of future behaviors. Zhang *et al.* adopt Markov models for prediction by regarding visit locations as states in a Markov chain, whereas the transition probability is assumed to be the same for all users [2, 25]. In these models, different elements in a behavior are considered independently, and thus the models are unable to capture contextual information from the entire behavior sequence. Recently, the *word2vec* framework has been proposed for POI recommendation [18]. Inspired by the words' contextual correlations in sentences, Feng *et al.* construct a geographical binary tree to incorporate spatial elements; the nearby POIs are assigned to nodes that are close in the binary tree [4]. Other techniques for POI recommendation incorporate geographical influence [11, 22] and temporal influence [15, 26]. Unlike such approaches, we take into account concrete action types, besides the traditional physical points, and incorporate the temporal variance into the learning objective.

Recent approaches to event prediction are also related to our work, such as approaches for predicting the type of a future event based on the observed sequence of events [3, 13]. Recurrent Neural Networks (RNN), assuming that the temporal dependencies change monotonously in a sequence, have been successfully applied in predicting sequential events. In the context of healthcare, Liu *et al.* have designed a method to predict clinical events by using an extension of LSTM[13]. Such approaches do not link spatial and temporal elements together to predict future event. Differently, we take into account these information such that our method can learn the rich semantic embeddings of different behavior elements.

Approaches for item recommendation are also related to our work, as our approach can be used for recommending an item as user action in her next visit to some location. Rendle *et al.* propose factorizing personalized Markov chains to model the transition probability between item pairs; this model is popular and often chosen as the baseline method [19]. With the growing popularity of language models, embedding-based methods have been increasingly used in item recommendation techniques. The recommendation model by Wu *et al.* is based on an embedding of users and items in a common latent space. The transition probability from one item to another is related to the Euclidean distance of the two items in the latent space [21]. A major limitation of such approaches is that they do not predict temporal elements in user behaviors[19, 21].

Inspired by the Translational Invariance in Geometry, the translation based models project the entities and the relations in a knowledge graph into a continuous latent space [1, 12]. A triple (h, r, t) in a knowledge graph indicates the fact of the entities h and t being associated with relationship r. Taking the triples (h, r, t) as inputs, the embedding vectors $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$ are learned by following the principle $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ since a relation vector $\boldsymbol{r}$ is regarded as a translation operation in the space, and $\boldsymbol{t}$ is the nearest neighbor of $\boldsymbol{h} + \boldsymbol{r}$. These works focus on the task of predicting the possibility of whether an entity has a specific relation with another entity that is not given in the knowledge graph. Although our embedding method is similar on modeling the relationships between behavior elements, we solve a different problem and integrate the sequential relations between

[2]https://en.wikipedia.org/wiki/Microeconomics

behaviors such that we can learn the functional and geographical semantics from crowd behaviors. In the prediction period, our model seems related to some generative models [8, 23] that provide a probability distribution over all possible events. These methods focus on mining the geographical specific semantics or patterns from crowd behaviors. For example, Yin *et al.* investigate the latent semantic regions in which the messages are posted with the same topic preference[23]. But they do not consider a user's preference and do not take into account the temporal factor. Thus they are not appropriate to predict a user behavior.

To summarize, we jointly predict user temporal-spatial behavior by combining multiple aspects together. We consider both group common behaviors and personalized preferences.

3 PROBLEM STATEMENT AND THE PROPOSED MODEL

3.1 Notions

Let U and L denote the set of users and the set of locations, respectively. Let A denote the set of user action types that are part of users' behavior. For example, buying tickets at a cinema and watching a movie are two action types. Let E denote the set of events. An event $e =< u, l, a, t >$ is an action executed by $u \in U$ at $l \in L$ at time point t. For convenience, we denote element x of event e by $e(x)$. The behavior history of a user u is represented as a sequence of temporal-spatial events $S^u = [e_1, e_2, ..., e_{|S^u|}]$, such that for any $i, j \in \{1, 2, ..., |S^u|\}$, $i < j$, $e_i(t) < e_j(t)$ holds. If there are several actions executed by a user on the same day at the same location, we treat them as different events.

Definition 1. ***U**ser **B**ehavioral **P**rediction (**UBP** for short) Problem. Given a user set U, a location set L, an action set A, and users' behavior histories set $\hat{S} = \{S^u | u \in U\}$, the User Behavioral Prediction Problem aims to predict which location a user v will visit, $v \in U$, and when, and what action the user would take there.*

3.2 The Embedding Model

We propose a joint embedding (JointE) model to combine the correlated elements of a behavior to solve the UBP problem, where the representations of user, location, and action are jointly learned in the same continuous space, denoted by $\boldsymbol{u}, \boldsymbol{l}, \boldsymbol{a} \in R^d$. Our approach is based on three considerations. 1) The functionality of a place is the critical factor of a behavior, since when combined with temporal patterns, it reveals people' inherent requirements or the intended purpose of the users. 2) Each user has her own behavior specificities, such as action type, location, and temporal pattern. The user specific behavior accurately reflects the location where a user goes, with which frequency, and what the user does at the location. 3) Groups of individuals often share similar patterns, that can be learned from collected data about their behaviors. This is often the basis of recommendation systems and reflects the proverb *Everyone thinks one of a kind, but in fact there are thousands of similar people.*

This process is implemented by the following three steps. The **first step** is to compute the correlations between users, actions, and locations. A user may perform different actions at the same location. A user behavior specificities are learned from historical data and modeled by the combination of action preferences and temporal patterns at different locations. For example, at the same shopping mall, a user may shop at a supermarket, go to a restaurant for lunch or watch a movie at a cinema. A user action preference vector $\boldsymbol{a}_l^u$ where $l \in L$ is calculated as the expectation of action vectors, $\boldsymbol{a}_l^u = \sum_{i=1}^{|A|} w_{l,i}^u \boldsymbol{a_i}$, where $\boldsymbol{a_i}$ denotes the vector of action $a_i \in A$ and $w_{l,i}^u$ denotes the weight of a_i. It is computed by the frequency of action a_i executed by u at l, $w_{l,i}^u = \frac{|\{e=(u,l,a,t)|e \in S^u, e(l)=l, e(a)=a_i\}|}{|\{e=(u,l,a,t)|e \in S^u, e(l)=l\}|}$. Given a user u, the set of her ever visited locations is denoted by L^u; the correlations between users, actions, and locations are obtained by the user's action preference. The vectors of u and a should be connected by the vector of l, namely $\boldsymbol{u} + \boldsymbol{a}_l^u = \boldsymbol{l}$. The loss function is defined as follows:

$$\ell_1 = \sum_{u \in U} \sum_{l \in L^u} ||\boldsymbol{u} + \boldsymbol{a}_l^u - \boldsymbol{l}||^2 \tag{1}$$

The **second step** is to compute the temporal correlations of user behavior elements. To model the high variance of time intervals in behaviors, we use temporal patterns to describe how often a user visits a location, which is defined as a set of discretized and comparable scales, $\Re_t = \{r_1, r_2, \cdots, r_{|\Re_t|}\}$. An intuitive example of periodical patterns is $\{never, seldom, sometimes, often, always\}$, which can be transformed into a set of certain time period according to different temporal granularities, such as weeks or months. We adopt a mapping function $f(\cdot)$ to transform a time interval into a temporal pattern based on the periodical visits to a place. A projection matrix for each temporal pattern is introduced to combine it with location functionality. A temporal pattern can be embedded into either the same space R^d in which user, location, and action are embedded, or into another space $R^{d'}$, where $d' \neq d, d' \in N^+$. Correspondingly, $M^r \in R^{d \times d}$ or $M^r \in R^{d \times d'}$. Our goal is to minimize the distance between $\boldsymbol{u}, \boldsymbol{r}$ and $\boldsymbol{l}$ in the pattern-specific space. The loss function is:

$$\ell_2 = \sum_{e=(u,a,l,r) \in E} ||\boldsymbol{u} \cdot M^r + \boldsymbol{r} - \boldsymbol{l} \cdot M^r||^2 \tag{2}$$

There are two advantages in using the pattern-specific projection matrix rather than directly embedding the temporal pattern r in the same continuous space. One is the capability of representing flexible temporal patterns into the same semantic space with other comparably stable user behaviors. In practice, a user preference is reflected by the functionality of the location, which changes less over time. However, the time points of behaviors are more dynamic and stochastic, both with respect to different users and to the same user over time. So the pattern-specific projection matrix helps represent such variations in a uniform way. The other is the capability of distinguishing the embeddings of users who have the same temporal pattern at the same location but with different action types. Their representations should be close in the temporal pattern space but far in the entity space.

The **third step** is to compute the semantic correlations between locations in user behavior sequences. We first define a context window with a size $c \in N^+$ to represent the location correlations in successive events. We use the notation l_i to denote the i-th location in a user behavior history. Given a user behavior history S^u and the i-th event e_i in S^u, the set of locations in the context window of

l_i is denoted by $C(l) = \{l_{i-c}, \ldots, l_{i-1}, l_{i+1}, \ldots, l_{i+c}\}$. Locations in a sequence should be closer with respect to both the semantic and geographical aspect than those not occurring in the sequence. The context vector $\boldsymbol{l_c}$ is represented by the expectation of location vectors in the context window, namely $\boldsymbol{l_c} = \frac{1}{2c}(\boldsymbol{l_{i-c}} + \ldots + \boldsymbol{l_{i+c}})$. The goal is to maximize the context locations conditional occurrence likelihood for all sequences. The probability $p(l|C(l))$ is defined by the softmax function: $p(l|C(l)) = \frac{exp(\boldsymbol{l} \cdot \boldsymbol{l_c})}{\sum_{l' \in L} exp(\boldsymbol{l'} \cdot \boldsymbol{l_c})}$. We adopt the log-posterior probability as the loss function over observed locations:

$$\ell_3 = -\sum_{u \in U} \sum_{l \in L^u} log\ p(l|C(l)) \tag{3}$$

Based on the previous formulations, we can define the objective function for computing the semantic correlations of elements in user behavior sequences. Let $\Theta = \{U, L, A, \mathfrak{R}_t\}$ denote the parameters of the model, which are learned by the joint optimization objective:

$$\Theta^* = \arg\min_{e \in E} \{\alpha \cdot \ell_1 + \beta \cdot \ell_2 + (1 - \alpha - \beta) \cdot \ell_3 + \lambda ||\Theta||_2^2\} \tag{4}$$

where $||\Theta||_2^2$ is the regularization component, α, β and λ are super adjustment parameters. The joint optimization process is presented in the supplementary section 3.3.

3.3 The Joint Optimization

We present the process for learning the parameters. The embeddings are learned by jointly optimizing the objective function given by Eq. 4. The objective ℓ_1 aims to minimize the connecting error of the locations that are visited. We tend to generate negative samples by replacing the location rather than the user. In this way, the chance of generating false negative samples can be reduced since the average number of users visiting each location is much larger than the average number of locations visited by each user. Given a user u, the set of his/her ever visited locations is L^u. The set of locations that u never visited, based on the historical records, is denoted by $L_n^u = L \setminus L^u$. For each event (u, l, a, t), we draw k negative location samples by random selection from L_n^u based on the probability distribution of locations for the training set. Let $f_a(u, l) = ||\boldsymbol{u} + \boldsymbol{a_l^u} - \boldsymbol{l}||^2$ denote the distance between $\boldsymbol{u} + \boldsymbol{a_l^u}$ and $\boldsymbol{l}$. ℓ_1 is then re-written as:

$$\ell_1 = \sum_{l \in L^u, l' \in L_n^u} (f_a(u, l) - f_a(u, l')) \tag{5}$$

We transform the function ℓ_1 (Eq.5) into the form of hinge loss:

$$\ell_1 = \sum_{l \in L^u, l' \in L_n^u} max(0, \gamma + f_a(u, l) - f_a(u, l')) \tag{6}$$

where γ is the margin parameter. Similarly, ℓ_2 (Eq.2) is rewritten as

$$\ell_2 = \sum_{l \in L^u, l' \in L_n^u} max(0, \gamma + f_r(u, l) - f_r(u, l')) \tag{7}$$

where $f_r(u, l) = ||\boldsymbol{u} \cdot M^r + \boldsymbol{r} - \boldsymbol{l} \cdot M^r||^2$ is the connecting error between $\boldsymbol{u}, \boldsymbol{r}$ and $\boldsymbol{l}$ in the pattern-specific space. The objective ℓ_3 aims to capture the sequential influence between locations. We adopt the negative sampling technique to train the model efficiently[17].

$$\ell_3 = -\sum_{l \in L^u, l_c \in C(l), l' \in L_n^u} (log\sigma(\boldsymbol{l} \cdot \boldsymbol{l_c}) + log\sigma(-\boldsymbol{l} \cdot \boldsymbol{l'})) \tag{8}$$

We leverage the stochastic gradient descent (SGD) algorithm to optimize the parameters. Each parameter is updated by $\Theta_i \leftarrow \Theta_i - \eta \frac{\partial \ell}{\partial \Theta_i}$, where η is the learning step.

We take $\boldsymbol{l}$, $\boldsymbol{u}$, $\boldsymbol{a_i}$ and $\boldsymbol{r}$ as examples to explain the gradient function for Θ in ℓ_1, ℓ_2.

$$\begin{aligned}
\frac{\partial \ell_1}{\partial \boldsymbol{u_i}} &= 2(\sum_{k=1}^{|A|} w_{l,k}^u \boldsymbol{a_{k,i}} - \sum_{k=1}^{|A|} w_{l',k}^u \boldsymbol{a_{k,i}} + \boldsymbol{l'_i} - \boldsymbol{l_i}) \\
\frac{\partial \ell_1}{\partial \boldsymbol{l_i}} &= -2(\sum_{k=1}^{|A|} w_{l,k}^u \boldsymbol{a_{k,i}} - \boldsymbol{l_i}) \\
\frac{\partial \ell_1}{\partial \boldsymbol{a_{i,j}}} &= 2w_{l,i}^u (\sum_{k=1}^{|A|} w_{l,k}^u \boldsymbol{a_{k,i}} + \boldsymbol{u_j} - \boldsymbol{l_j}) \\
&\quad - 2w_{l',i}^u (\sum_{k=1}^{|A|} w_{l',k}^u \boldsymbol{a_{k,i}} + \boldsymbol{u_j} - \boldsymbol{l'_j}) \\
\frac{\partial \ell_2}{\partial \boldsymbol{r_j}} &= 2\sum_{i=1}^{d} \boldsymbol{l'_i} \cdot M_{ij}^r - 2\sum_{i=1}^{d} \boldsymbol{l_i} \cdot M_{ij}^r
\end{aligned} \tag{9}$$

Since ℓ_3 involves only location embeddings and the other parts involve multiple elements, we split the whole optimization process into two sub-processes and iteratively execute them. The process is balanced by a hyper parameter $\rho \in (0, 1)$. In each iteration, we select an optimization sub-process according to whether a random variable $x \in (0, 1)$ is smaller than ρ, then update the parameters for the selected sub-process. Details are shown in Algorithm 1. The convergence condition is satisfied when the loss decrease is within a threshold.

The overall model complexity is $O(d(|U| + |A| + |L|) + (d + 1)d'|\mathfrak{R}_t|)$. Although the training process is time consuming, it is performed only once. In practice, the common parameters can be reused, such as $U, A, L, \mathfrak{R}_t$. For a new user who never appeared in the model, the model complexity of learning the user vector is $O(d)$, which is very efficient.

4 USER BEHAVIOR PREDICTION

In this section, we introduce two prediction models based on the embeddings of user behavior elements. Here, for the ease of calculation, we discretize continuous time t to temporal patterns. Our problem can be formulated as: Our goal is to predict user u's next action a_t and next temporal pattern r_t, given the next location l_t and the historical behavior sequence S^u.

4.1 Probabilistic Inference Model

Our goal is to estimate the joint probability of the elements and select the most likely behavior from the set of behavior candidates. Let $\mathbb{E}$ denote the combination of all elements in behaviors. The probability distribution is modeled as the mixture of location preference and each element-level preference at the location.

$$\begin{aligned}
e_u^* &= \arg\max_{e \in \mathbb{E}} p(e|S^u, \Theta) \\
&= \arg\max_{e=(u,l,a,t)} p(l|S^u, \Theta) \cdot p(a|u, l, \Theta) \cdot p(t|u, l, \Theta)
\end{aligned} \tag{10}$$

Each element is computed as follows.

Algorithm 1 Model Training

Input: training set $\hat{S} = \{S^u | u \in U\}$, user set U, location set L, action set A, pattern set $\Re_t$, embedding dimensions d, d', context window size: c, negative sample size: k, learning step: η, optimization part selection: ρ.
Output: all parameters in Θ.

/*initialization*/
for $i \in U \cup L \cup A$ **do**
  $\boldsymbol{i} \leftarrow uniform(-\frac{6}{\sqrt{d}}, \frac{6}{\sqrt{d}}), \boldsymbol{i} \leftarrow \frac{\boldsymbol{i}}{||\boldsymbol{i}||}$
end for
for $r \in \Re_t$ **do**
  $\boldsymbol{r} \leftarrow uniform(-\frac{6}{\sqrt{d'}}, \frac{6}{\sqrt{d'}}), \boldsymbol{r} \leftarrow \frac{\boldsymbol{r}}{||\boldsymbol{r}||}$
  $M^r \leftarrow M^r_{ij} = 1 \; if \; i = j, otherwise \; 0$
end for
/*optimization*/
repeat
  Sample $u \in U$ randomly
  draw n locations and $k * n$ negative samples
  $x = random(0, 1)$
  if $x < \rho$ **then**
    update parameters in loss functions ℓ_1, ℓ_2
  else
    update parameters in loss function ℓ_3
  end if $\ell = \alpha \cdot \ell_1 + \beta \cdot \ell_2 + (1 - \alpha - \beta) \cdot \ell_3 + \lambda ||\Theta||_2^2$
until ℓ converges
return Θ

1) The probability distribution over locations in the next behavior is calculated by

$$p(l|S^u, \Theta) = p(l_{i+1}|l_i^u, \Theta) = \frac{exp(\boldsymbol{l_{i+1}} \cdot \boldsymbol{l_i^u})}{\sum_{l' \in L} exp(\boldsymbol{l'} \cdot \boldsymbol{l_i^u})} \tag{11}$$

where i is the size of S^u, $\boldsymbol{l_{i+1}}$ is the latent vector of l_{i+1}, l_i^u is the set of locations u recently visited, $\boldsymbol{l_i^u}$ is the expectation of location vectors in l_i^u.

2) $p(a|u, l, \Theta)$ denotes the probability distribution of *action type* at l in the next behavior. Formally,

$$p(a|u, l, \Theta) = \frac{exp[(\boldsymbol{u} + \boldsymbol{a}) \cdot \boldsymbol{l}]}{\sum_{a' \in A} exp[(\boldsymbol{u} + \boldsymbol{a'}) \cdot \boldsymbol{l}]} \tag{12}$$

3) $p(r|u, l, \Theta)$ denotes the probability distribution of *temporal pattern* towards l of the next behavior. Formally,

$$p(r|u, l, \Theta) = \frac{exp[(\boldsymbol{u} \cdot M^r + \boldsymbol{r}) \cdot (\boldsymbol{l} \cdot M^r)]}{\sum_{r' \in \Re_t} exp[(\boldsymbol{u} \cdot M^{r'} + \boldsymbol{r'}) \cdot (\boldsymbol{l} \cdot M^{r'})]} \tag{13}$$

4.2 Attention-Based Model

We also try the Attention Based Recurrent Neural Network Model (ARNN) so as to capture the users' dynamic preferences. This model is based on the equations 10 and 11, and combine the predication on action and pattern in equations 12 and 13. As presented in Figure 2, there are three parts. First, based on the embeddings learned by the JointE model, a user behavior event at step t is modeled as the vector $\boldsymbol{X_t}$ by concatenating the embeddings of behavior elements. The second part is a RNN network where a user behavior sequence of $\boldsymbol{X_t}$ is fed into the network. The output of

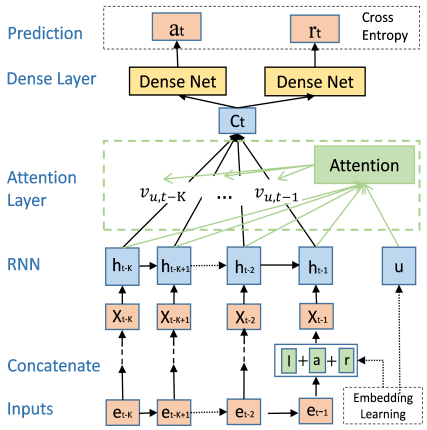


Figure 2: Attention-Based Neural Network Prediction Model

Table 1: Statistics for the datasets used in the evaluation.

Dataset	#records	#users	#locations	#action
Koubei	579,993	19,977	1,104	11
Gas	581,367	35,418	693	8

hidden state in RNN is denoted by $\boldsymbol{h_t}$. The attention value on each step is computed against the hidden state $\boldsymbol{h_t}$ and the user vector $\boldsymbol{u}$, denote by $v_{u,t}$, which represents a user's dynamic preference. The third part takes the hidden state on step t and the attention on several previous behaviors as input, and then predicates the next behavior by $softmax$ function on a dense net. Details are given below.

$$\begin{aligned} \boldsymbol{X_t} &= concanate[\boldsymbol{l}, \boldsymbol{a}, \boldsymbol{r}] \\ \boldsymbol{h_{t+1}} &= tanh(\boldsymbol{M^h h_t} + \boldsymbol{M^x X_t} + \boldsymbol{b^h}) \\ v_{u,t-k} &= \frac{e^{(\boldsymbol{h_{t-k}} \boldsymbol{M^v} + \boldsymbol{b^v})\boldsymbol{u}}}{\sum_{k=1}^{K} e^{(\boldsymbol{h_{t-k}} \boldsymbol{M^v} + \boldsymbol{b^v})\boldsymbol{u}}} \\ \boldsymbol{C_t} &= \sum_{k=1}^{K} v_{t,t-k} \boldsymbol{h_{t-k}} \end{aligned} \tag{14}$$

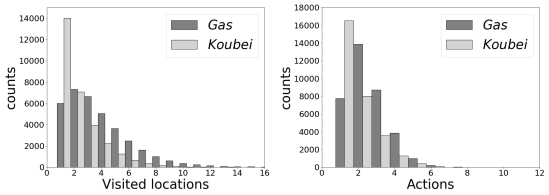
where K denotes the size of time window, $\boldsymbol{M}$ and $\boldsymbol{b}$ represent the weight matrices and bias vectors.

5 DATA DRIVEN MODEL SETTING

5.1 Datasets

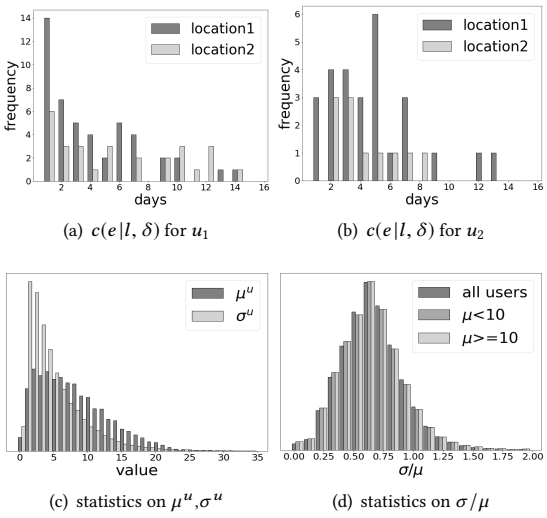
We use two representative real datasets with both temporal and spatial features for user behaviors. The first dataset, *Koubei*, is collected from a popular O2O service platform from Jun. 2016 to Oct. 2016. It contains 579,993 records involving 1,104 locations and 19,977 users' payment data. We retain the users with more than 15 records. The information about merchants includes location information and category information, such as barbecue, buffet, hot pot and etc. The data set and source code of this paper can be obtained from https://github.com/yghn14/JointE.

The *Gas* dataset records user transactions in gas station, including car fuel filling and buying goods in the station store, which were collected from a province branch of PetroChina from Jan. 2017



(a) (b)

Figure 3: The statistics on user behaviors in the two datasets. (a) The x-axis denotes the number of locations a user ever visited, and the y-axis denotes the number of users. (b) The x-axis denotes the number of action types in user historical records and the y-axis denotes the number of users.



(a) $c(e|l,\delta)$ for u_1 (b) $c(e|l,\delta)$ for u_2

(c) statistics on μ^u,σ^u (d) statistics on σ/μ

Figure 4: The correlations between elements of behaviors in the *Gas* dataset. (a)-(b) The x-axis denotes intervals and the y-axis denotes the frequency $c(e|l,\delta)$. (c) The statistics of δ for users. (d) The probability distributions of value σ/μ for: (1) all the users; (2) users with $\mu < 10$; and (3) users with $\mu >= 10$.

to Dec. 2017. It includes both online transactions and offline behaviors. Each record includes the details of each transaction, i.e., the time, location (i.e., latitude and longitude), product category(i.e., fuel, car accessory and food), and the amount. We filter the user data based on the number of historical records per individual, and retain those with more than 15 records. Totally, there are 581,367 records, involving 35,418 users and 693 places. The statistics about the datasets are shown in Table1.

5.2 User Behavior Statistics for Model Adaption

We analyze the datasets to gather some basic statistics on the user behaviors so as to understand their semantics. We first investigate the spatial elements of user behaviors based on locations and actions associated with each user in Figure 3. Although user preferences look stable with respect to locations and actions since each preference is associated with only a few places and actions, these correlations are actually uncertain.

For the temporal aspect, we quantify the user periodical patterns on each location. We first randomly select two users from the *Gas* dataset, denoted by u_1 and u_2, and count their periodical visits to each location within a given period by $c(e|l,\delta) = |\{e \in S^u | l \in L^u, e(\delta) = \delta\}|$. The results in Figure 4(a)-4(b) show that even for the same user, temporal patterns have high variance. To understand a user's overall behavior specificity, we compute the expectation μ^u and variance σ^u on time intervals for u. The statistics on the *Gas* dataset in Figure 4(c) show large differences between users, thus showing that modeling temporal patterns is challenging.

For each user u, we obtain the location-specific temporal intervals δ^l in S^u and the mean and standard deviation of this set, denoted by μ, σ. Based on the notion of periodical rules, we randomly chose a threshold (for example, 10) and classify users into two groups by μ (for example, $\mu \geq 10$ and $\mu < 10$). The statistics for σ/μ are shown in Figure 4(d). We can see that the statistics for these groups are approximated to the same probability distribution. These results show that it is better to use temporal patterns instead time intervals.

A user's periodic patterns with respect to a location are computed by a mapping function $f(\delta_i^l) \rightarrow r : \mathfrak{R}_t$. This encoding method helps not only in modeling user temporal factors according to a unified statistical scheme, but also in taking into account user-specific periodic preferences.

6 EXPERIMENTS

6.1 Baseline Models and Metrics

The algorithms are implemented in Python, and all experiments are performed on a x64 machine with 2.5GHz intel Core i7 CPU and 16GB RAM. We report here results from the experiments on the *Gas* and *Koubei* datasets. For each user, the behavior sequence is partitioned into two parts, 80% for training and 20% for testing. To verify the effectiveness of our model, we select several state-of-the-art methods as comparison. The following models are compared in our evaluations:

JointE. This is the proposed model described in previous sections, which learns the embeddings of behavior elements using a joint-objective optimization. Temporal patterns are learned from location-specific time intervals, the mapping function is $f(\delta_i^l) \rightarrow r : \mathfrak{R}_t$. For the super parameters, we tried several settings and choose the best combination, which are c=1,k=2,d=d'=20. We find that the adjustment on the parameters α, β, λ has little influences on the results. Thus we adopt $\alpha = \beta = \lambda = 0.25$ to guarantee each part has an equal importance in the optimization. Under these settings, we learn the latent embeddings for users, locations, actions, and temporal patterns.

JointE-n. This is a specific form of our proposed method, where the temporal pattern is redefined while the remaining parts of the model are identical to JointE. Temporal patterns are learned from the normal time intervals, the mapping function is $f(\delta_i) \rightarrow r : \mathfrak{R}_t$. The learning process and parameter settings are the same as JointE.

Table 2: Comparison on the performance in solving UBP.

Dataset	Method	Behavior			Action			Pattern			Location		
		P@1	P@5	P@10	P@1	P@3	P@5	P@1	P@2	P@3	P@1	P@2	P@3
Gas	JointE	**0.559**	**0.825**	**0.906**	0.606	**0.836**	**0.957**	**0.853**	**0.923**	**0.963**	0.351	0.352	0.353
	JointE-n	0.373	0.635	0.758	**0.611**	0.797	0.953	0.660	0.840	0.922	**0.452**	0.453	0.454
	FPMC	0.283	0.284	0.528	0.256	0.577	0.794	0.351	0.656	0.849	0.367	**0.462**	**0.516**
	LSTM	0.429	0.757	0.858	0.510	0.810	0.945	0.801	0.891	0.952	0.395	0.437	0.458
	MLP	0.324	0.571	0.689	0.530	0.724	0.875	0.735	0.799	0.893	0.036	0.088	0.111
	STELLAR	-	-	-	-	-	-	0.339	0.532	0.783	0.311	0.411	0.499
Koubei	JointE	**0.476**	**0.750**	**0.938**	**0.528**	**0.832**	**0.931**	**0.850**	**0.921**	**0.980**	**0.627**	**0.629**	**0.630**
	JointE-n	0.410	0.665	0.756	0.445	0.778	0.843	0.513	0.718	0.876	0.575	0.576	0.578
	FPMC	0.255	0.257	0.267	0.207	0.535	0.723	0.405	0.649	0.844	0.393	0.497	0.555
	LSTM	0.382	0.741	0.840	0.482	0.824	0.910	0.813	0.894	0.925	0.336	0.399	0.421
	MLP	0.379	0.660	0.669	0.456	0.783	0.820	0.775	0.876	0.892	0.073	0.103	0.122
	STELLAR	-	-	-	-	-	-	0.339	0.605	0.873	0.310	0.439	0.516

Factorized Personalized Markov Chains (FPMC)[19]. Rendle *et al.* embed users' preferences and their personalized Markov chain to provide next basket item prediction. The expected element of behavior is predicted based on the latest behavior. The latent dimensions is set to d =20, which is the same as our method.

Long Short-Term Memory Neural Networks (LSTM)[7]. LSTM is acknowledged as one of the best methods for predicting sequential data. The inputs to LSTM are users' behavior sequences with the same length, and the output is the element to be predicted. The implementation of LSTM is based on the machine learning framework TensorFlow. The number of hidden-nodes is 100.

Multi-Layer Perceptron (MLP) is widely used in conventional prediction systems and is an efficient method for task prediction. We use all elements of behavior as the input layer and the expected elements as the output. The network includes 3 hidden-layers and each layer includes 100 hidden-nodes.

Spatial-Temporal Latent Ranking (STELLAR)[26] has been widely used in POI prediction. It considers user-location interaction, location-location interaction, and time-location interaction. It predicts the location and temporal pattern based on the latest behavior; action and behavior information is not considered in this approach. The latent dimensions is set to d =20, at which it approaches the best results.

A widely adopted evaluation metric is the top-K similar candidates for a target behavior, which verifies whether the true behavior is in the results. The function $hit@K(e) \in \{0, 1\}$ is used to indicate whether the real behavior e is in the top-K recommendation list. Let E_{test} denote the set of cases for prediction. We adopt the precision metric $P@K$ to quantify the prediction results.

$$P@K = \frac{\sum_{e \in E_{test}} hit@K(e)}{|E_{test}|} \tag{15}$$

6.2 Evaluation on Behavior Prediction

The prediction on an event $e = (u, a, r, l)$ is computed against Eq.10, i.e. $p(e|u) = p(a|u, l)p(r|u, l)p(l|S^u, \Theta)$ or $p(e|u, l) = p(a|u, l)p(r|u, l)$ for some location l. A successful behavior result is justified by the ground truth behavior e being in the top-K recommendation list sorted in a descending order based on the prediction values. The results in the first three columns of Table 2 show that an increasing K increases the performance for all methods. JointE has the best performance. Consider the metric $P@1$, JointE outperforms the other methods by 30% and 16% on two datasets, respectively. In comparison with the other version of our method, JointE-n takes into account general intervals with all behaviors of a user, JointE learns location-specific temporal patterns that can reveals a user's specific behaviors.

Table 3: Comparison for Behavior Prediction (P@1).

	Gas			Koubei		
$H^u(l)$	$(0, \sim)$	$(0.5, \sim)$	$(1, \sim)$	$(0, \sim)$	$(0.5, \sim)$	$(1, \sim)$
#users	19467	15516	7983	5059	3045	711
JointE	0.551	0.537	0.516	0.418	0.395	0.372
ARNN	0.588	0.581	0.567	0.461	0.412	0.391
Imprmt Ratio	6%	8%	9%	10%	4%	5%

Evaluation on Attention-based Model. We use the task of behavior prediction to quantitatively evaluate JointE and ARNN methods. We classify all users into three groups against $H^u(l)$, namely: $(0, \sim)$,$(0.5, \sim)$ and $(1, \sim)$, and conduct experiments on different settings. The performance comparison by P@1 is shown in Table 3, where the last row shows the improvements of ARNN over JointE. Comparing with JointE, ARNN is more appropriate to capture a user's dynamic preferences. Concretely, ARNN shows at least an increase of 6% on *Gas* dataset and 4% on *Koubei* dataset improvement of P@1.

To further compare our methods against the other methods, we perform the following tasks.

Action Prediction. For a user u, this task is to predict the user action type at a specific location l. The results in columns 4, 5, and 6 of Table 2 show that our methods outperform the other methods. For example, JointE achieves 0.606 in *Gas* and 0.528 in *KouBei* at

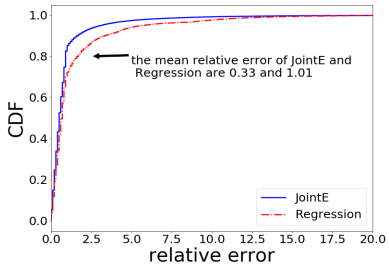


Figure 5: The comparison with respect to the prediction of temporal intervals with the CDF of relative error.

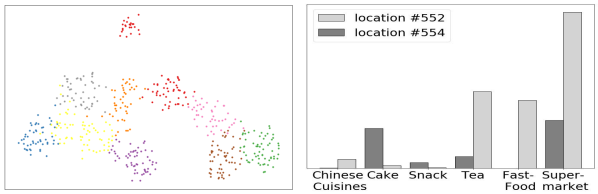


Figure 6: (a)The clustering results on locations based on the learned embeddings in the *Koubei* dataset, k=10. (b) The distribution of users' action types in two locations in the *Koubei* dataset.

metric $P@1$. We notice that all models perform better for *Gas* than for *Koubei*. A reason could be that the functionalities of locations in *Gas* are simpler than in *Koubei*, so that user actions at each location are more stable.

Temporal Pattern Prediction. For a given user u, this task predicts the point in time of next behavior at location l. We first predict a temporal pattern $r \in \Re_t$, namely, a period of time, and then map it onto a certain time point. The experimental results in columns 7, 8, 9 of Table 2 show that JointE outperforms other state-of-the-art latent ranking methods and neural network models for both *Gas* and *Koubei*. Although STELLAR can capture the temporal effect in a concrete scale, such as day, week, and month, it does not work well in capturing temporal intervals. The results show that FPMC and STELLAR perform worse in the temporal pattren perdiction since they predict temporal patterns given a user's recent checkd-in behavior that may be irrelevant to the next.

Our work is the first approach to propose an efficient method able to accurately predict temporal patterns, which are considered difficult to predict. To further understand the exact temporal interval for a temporal pattern in JointE, we predict the next time interval as a regression problem. For a user u, the predicted temporal pattern by our method is transferred to a concrete time point by the reverse function $f^{-1}(r) = \delta$. The comparison method we adopt is the linear regression (Regression). We use the metric $relative\ error = |\frac{prediction-truth}{truth}|$. From the results in Figure 5, we can see that our model captures the temporal interval effects better.

Location Prediction. This task is to predict a user's next visited location; the results are listed at the last three columns of Table 2. In the *Koubei* dataset, our methods outperform the other methods with respect to different metrics. On the *Gas* dataset, JointE-n performs the best with respect to metric $P@1$, while FPMC has a better performance with respect to metrics $P@2$ and $P@3$. The reason is that FPMC combines the user preference into the Markov transition function between locations. Comparably, our method embeds multiple elements into location vectors, such as the action and the temporal elements, so it is appropriate for a combined behavior prediction.

Understanding the Semantics of Embedding. To help understand the prediction results, we extract semantic information from the learned latent embeddings and discuss how the semantics help solve UBP. We provide an intuitive view of the embeddings for locations by visualizing them using tSNE[16].

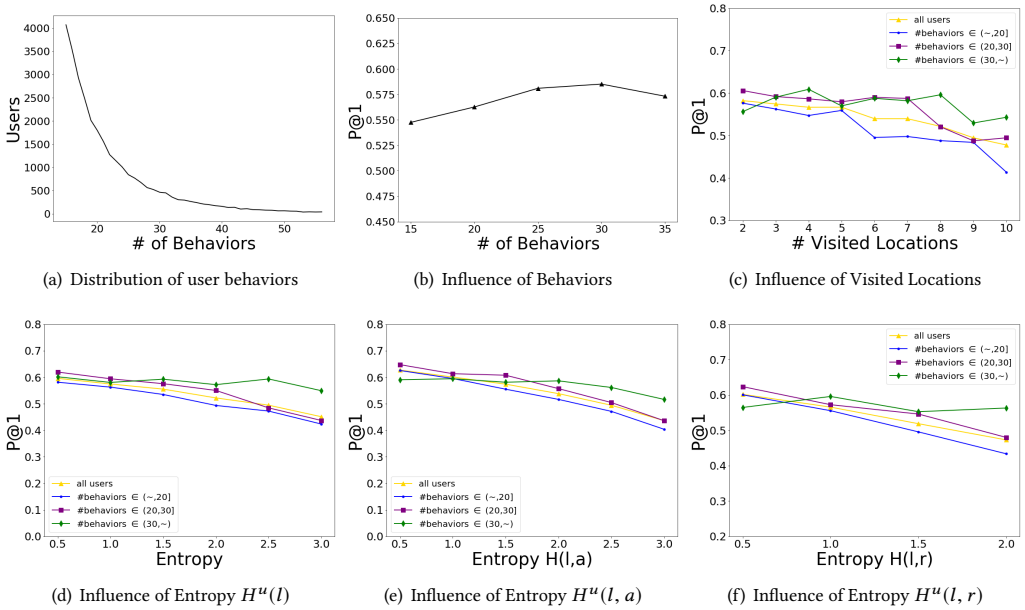
For the *Koubei* dataset, we cluster the locations into k clusters based on their embeddings and color them differently by cluster labels in Figure 6(a). We then randomly choose two locations from different clusters in *Koubei* that are geographically close (the closer locations have numbers that are closer) and count the frequencies of actions associated with these locations for all users. From the results listed in Figure 6(b), where the x-axis represents the actions and the y-axis represents the proportion of actions, we can see that there are different preferences with respect to actions at these locations. For example, the proportion of Chinese restaurants in location #552 is obviously higher than in location #554. Such differences are learned in the embeddings so that they are classified into two clusters.

Those results show that the location-specific periodic information is useful in enhancing the performance of behavior prediction tasks and that learning joint representations is more effective for modeling the elements of user behavior. The embeddings involve not only the functional characters of locations but also the temporal and action specificities.

6.3 Discussion on Parameter Influence and Model Limitation

Solving the UBP problem highly relies on the dynamics of user behaviors. We thus discuss how these characteristics influence the performance of our method (JointE) on behavior prediction and report the results for the two datasets in Figure 7 and Figure 8, respectively. We first analyze the influence of the number of user behaviors. The statistics in Figure 7(a) and Figure 8(a) show that the number of behaviors follows a long-tail distribution. We conduct experiments on different behavior threshold for user selection. The results in Figure 7(b) and Figure 8(b) show that an increasing threshold leads to better results except that a very large threshold may result in few users remaining in the dataset that then makes the results unstable.

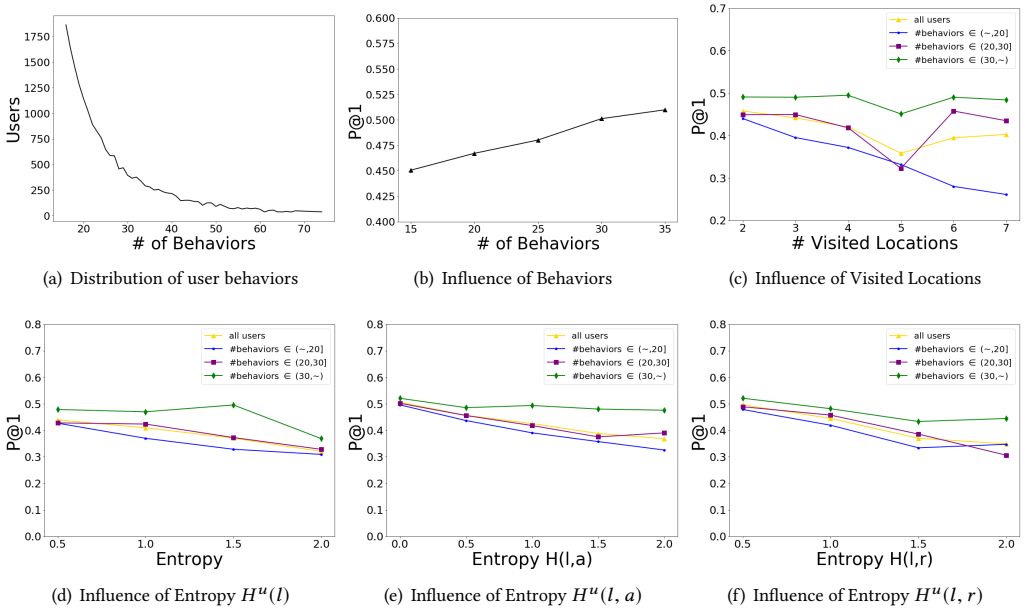
We then analyze the impact of uncertainty about user behaviors. We classify all users into three groups against $|S^u|$, namely: $(\sim, 20]$, $(20, 30]$ and $(30, \sim)$, and conduct experiments on different settings. We first evaluate the influence of visited locations $|L^u|$. As we increase $|L^u|$, the accuracy decreases gradually as shown in Figure 7(c). Since there is more uncertainty in user behaviors with larger $|L^u|$. However, this is not the case for Koubei dataset. The accuracy is decreased with varying $|L^u|$ from 2 to 5 but rises



(a) Distribution of user behaviors (b) Influence of Behaviors (c) Influence of Visited Locations

(d) Influence of Entropy $H^u(l)$ (e) Influence of Entropy $H^u(l,a)$ (f) Influence of Entropy $H^u(l,r)$

Figure 7: Influence of parameters and settings for the *Gas* dataset.



(a) Distribution of user behaviors (b) Influence of Behaviors (c) Influence of Visited Locations

(d) Influence of Entropy $H^u(l)$ (e) Influence of Entropy $H^u(l,a)$ (f) Influence of Entropy $H^u(l,r)$

Figure 8: Influence of parameters and settings for the *Koubei* dataset.

when it is larger than 5 in Figure 8(c). This is because a large $|L^u|$ also means that the user has enough behaviors, thus the model can enhance the prediction accuracy. Then we use the entropy $H^u(l) = -\sum_{l \in L^u} p_l \log p_l$ to quantify the uncertainty and present the results for two datasets in Figures 7(d) and 8(d), respectively. Generally, the performance for $P@1$ decreases when the uncertainty increases, and increases as the behavior threshold increases.

We also analyze how the uncertainty between location and action influences the performance. Let $P_1 = \{(l,a)|l \in L, a \in A\}$, $P_2 = \{(l,r)|l \in L, r \in \Re_t\}$ represent all possible combinations of locations and actions or patterns, respectively. The entropy $H(l,a) = -\sum_{(l,a)\in P_1} p(l,a) \log p(l,a)$ represents the uncertainty between location and action for user u, where $p(l,a) =$

$\frac{|\{e=(u,l,a,t)|e\in S^u, e(l)=l, e(a)=a\}|}{|\{e=(u,l,a,t)|e\in S^u\}|}$. Similarly, the entropy $H(l,r) = -\sum_{(l,r)\in P_2} p(l,r)\log p(l,r)$ quantifies the uncertainty between location and pattern. We conduct experiments on different settings for $H(l,a)$ and $H(l,r)$. The results in Figures 7(e)-7(f) and 8(e)- 8(f) show that, as the value of entropy increases, the performance decreases since there are more behavioral patterns in user behaviors. The trends are opposite when the behavior threshold increases.

Our method predicts temporal-spatial behaviors by taking advantage of location functionality and temporal patterns. The crowd behaviors reflect the functionality of a location. The location functionality, on the other hand, influences individuals' behaviors, and plays an important role in predicting future behaviors. The predictive capability of our method is also limited to the choice of temporal patterns; a temporal pattern represents a type of periodic properties and each location is characterized by various types of temporal patterns. Similarly, each location is characterized by various action types. A user preference can be learned well should we have enough user behaviors. Thus our model has the power to predict future event precisely.

7 CONCLUSION

In this paper, we investigate the problem of predicting a user temporal-spatial behavior. To understand the semantics of the different behavior elements, a novel embedding model is proposed, in which the embeddings of users, locations, and actions are learned in the same continuous space. Location functionality is the critical factor for connecting different elements of the behavior, which is learned from crowd behaviors. We introduce a temporal pattern scheme to represent how often users visit locations. We conduct experiments against two representative datasets and the results show that our approach outperforms state-of-the-art methods. We also analyze the semantics of embeddings from the perspective of location.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2018YFC0831401), the National Natural Science Foundation of China (91646119), the Major Project of NSF Shandong Province (ZR2018ZB0420), and the Key Research and Development Program of Shandong Province (2017GGX10114).

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*. 2787–2795.
[2] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: successive point-of-interest recommendation. In *International Joint Conference on Artificial Intelligence*.
[3] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1555–1564.
[4] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical Latent Representation for Predicting Future Visitors.. In *AAAI*. 102–108.
[5] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *International Conference on Artificial Intelligence*. 2069–2075.
[6] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware point of interest recommendation on location-based social networks. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 1721–1727.
[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. 9 (12 1997), 1735–80.
[8] Younghoon Kim, Jiawei Han, and Cangzhou Yuan. 2015. TOPTRAC: topical trajectory pattern mining. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 587–596.
[9] Noam Koenigstein, Gideon Dror, and Yehuda Koren. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *ACM Conference on Recommender Systems*. 165–172.
[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.
[11] Xutao Li, Gao Cong, Xiao Li Li, Tuan Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation. (2015), 433–442.
[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion.. In *AAAI*, Vol. 15. 2181–2187.
[13] Luchen Liu, Jianhao Shen, Ming Zhang, Zichang Wang, and Jian Tang. 2018. Learning the Joint Representation of Heterogeneous Temporal Events for Clinical Endpoint Prediction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.
[14] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the context of locations for personalized location recommendations. In *International Joint Conference on Artificial Intelligence*. 1188–1194.
[15] Yanchi Liu, Chuanren Liu, Bin Liu, Meng Qu, and Hui Xiong. 2016. Unified point-of-interest recommendation with temporal interval assessment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1015–1024.
[16] Laurens Van Der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 2605 (2008), 2579–2605.
[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Computer Science* (2013).
[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. 26 (2013), 3111–3119.
[19] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *International Conference on World Wide Web*. 811–820.
[20] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. (2015), 403–412.
[21] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized next-song recommendation in online karaokes. In *ACM Conference on Recommender Systems*. 137–140.
[22] Mao Ye, Peifeng Yin, Wang Chien Lee, and Dik Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. 325–334.
[23] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. 2011. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*. ACM, 247–256.
[24] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 363–372.
[25] Jia Dong Zhang, Yanhua Li, and Yanhua Li. 2014. LORE: exploiting sequential influence for location recommendations. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems*. 103–112.
[26] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation.. In *AAAI*. 315–322.
[27] Vincent Wenchen Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative Filtering Meets Mobile Recommendation: A User-Centered Approach. In *Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, Usa, July*.
[28] Vincent W. Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, Usa, April*. 1029–1038.