A Semantic Sequential Correlation based LSTM Model for Next POI Recommendation

1st Guanhua Zhan
School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
172050093@hdu.edu.cn

2nd Jian Xu
School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
jian.xu@hdu.edu.cn

3rd Zhifeng Huang
School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
172050025@hdu.edu.cn

4th Qiang Zhang
School of Cyberspace
Hangzhou Dianzi University
Hangzhou, China
dq.z@outlook.com

5th Ming Xu
School of Cyberspace
Hangzhou Dianzi University
Hangzhou, China
mxu@hdu.edu.cn

6th Ning Zheng
School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
nzheng@hdu.edu.cn

***Abstract*—The widespread of location-based social networks has generated massive check-in sequences in chronological order. Forecasting check-in sequences is significant while challenging due to the check-ins' sparsity problem. Existing methods have followed closely to incorporate spatial and temporal context to alleviate the data sparsity problem, but neglect the semantic sequential correlation between check-ins. Howbeit, incorporating the semantic sequential correlation between check-ins for next POI recommendation encounters the challenges of semantic sequential correlation measurement and sequential behavior modeling. To measure the semantic sequential correlation, we apply a semantic sequential correlation calculation model based on a semantic correlational graph that incorporates the time intervals' influence to calculate the semantic sequential correlation. Then, we apply a novel Long Short-Term Memory (LSTM) framework equipped with two additional semantic gates that takes the additional semantic sequential correlation as the extra input to capture users' sequential behaviors and model their long short-term interest with the restrictions in the semantic level. Finally, we cluster users into different groups as an improvement of our model to achieve a more accurate recommendation. Our proposed model is evaluated on a real-world and large-scale dataset and the experimental results demonstrate that our method outperforms the state-of-the-art methods for next POI recommendation.**

***Index Terms*—Location-based social networks, next POI recommendation, LSTM**

I. Introduction

With the flourishing of Location-based Social Networks (LBSNs), such as Gowalla, Foursquare and Yelp, considerable check-in data has been generated. For instance, there are more than 12 billion check-ins generated by 50 million people in Foursquare[1], which leads to a large opportunity to analyze users' behaviors and preferences about POIs [1]. To improve users' engagement through sequential dependence of users' check-ins, and to offer a better service in making geo-targeted advertisements and coupon delivery, it is significant for both users and merchants to get the most likely POI in the next step, and this problem has attracted a lot of attention recently [2], [3]. Different from the general POI recommendation that focuses on forecasting users' preferences on POIs, the next POI recommendation is based on the users' historical check-ins in chronological order to satisfy recommendations which consider not only users' preferences, but also the dependence between POIs.

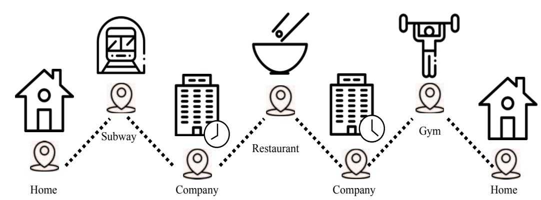

Fig. 1. A simple example of user's check-in sequence in the semantic level

However, predicting the next POI is challenging for two reasons: (1) data sparsity. Even though LBSNs generate a large number of check-ins, the number of check-ins generated by one person is very limited. For example, the density of data that can be used for POI recommendation is usually around 0.1% [4], while the density of Netflix data for movie recommendation can reach 1.2% [5]. Furthermore, the next POI recommendation is more sensitive to the data sparsity problem as sparse check-in sequences are more difficult to predict. (2) sequential behaviors modeling. Modeling users' behaviors by capturing the relations between users' actions is the key part of the next POI recommendation, effectively utilizing the correlation between sequential data affects the recommendation performance largely.

To tackle the above challenges, earlier methods of next POI recommendation rely on modeling users' mobility patterns and

[1]https://foursquare.com/about

2375-0324/19/$31.00 ©2019 IEEE
DOI 10.1109/MDM.2019.00-65

the sequential correlation between POIs by Markov Chain [6], [7]. Cheng et al. [6] propose a personalized Markov Chain taking account the spatial constraint at the same time, and Lv et al. [7] investigate the effect of living habits on a hidden Markov model to predict users' mobility patterns. However, existing Markov Chain methods is limited because of the difficulty of capturing longer sequential context and high computational complexity. Afterwards, due to recurrent neural network (RNN)'s ideal effect of modeling sequential data, Liu et al. [8] propose a model named ST-RNN which aims to predict users' mobility patterns with spatial and temporal constraints, and Kong et al. [9] propose a Spatial-Temporal Long Short-Term Memory (ST-LSTM) model which naturally combines spatial-temporal influence into LSTM to mitigate the problem of data sparsity. Nevertheless, all previous approaches neglect the powerful impact of semantic sequential correlation between check-ins.

The semantic sequence is a representation of check-in sequences in the semantic level. For instance, from a sequence shown in Fig. 1: $\{Home, Subway, Company, Restaurant, Company, Gym, Home\}$, which selected from check-ins, it is easy to imagine this user's daily trajectory and we can find that this user's check-in behaviors have a certain semantic correlation, and the semantic correlation between check-ins vary from pairs to pairs, e.g., users are more likely to go to the gym from the company than vice versa, because users are inclined to exercise after work. The greater the semantic correlation, the more likely the user will choose, this observation reveals the importance of semantic correlation. While the semantic sequential correlation is the correlation between all adjacent categories of a semantic sequence, which is expressed as a sequence of correlation in chronological order.

This paper presents an in-depth investigation on incorporating the semantic sequential correlation of check-in sequence to boost the next POI recommendation performance. To begin, we measure the semantic sequential correlation with a semantic correlational graph, in terms of the check-ins' categories as semantic information, which incorporates the time intervals' influence to it. After that, we use a semantic sequential based LSTM (LSTM-S) equipped with two additional semantic gates to capture users sequential behaviors and model their long short-term interest with the restrictions in the semantic level. Beyond that, in order to make our model more representative, we utilize the K-means method to cluster users into distinct groups then fit each group to the corresponding model we proposed.

Overall, the contributions of this paper can be summarized as follows:

- We utilize a semantic correlational graph to get the semantic correlation of all the adjacent categories that facilitates the acquisition of the semantic sequential correlation. This method first splits each user's check-in sequence into one-to-one sequences (e.g., transform $\{Company, Gym, Home\}$ into $\{Company, Gym\}$ and $\{Gym, Home\}$), then traverse the semantic correlational graph to obtain the number of directed edges and the weight of each directed edge defined by the time intervals of each one-to-one sequence. And then, the semantic correlation is generated from our semantic sequential correlation calculation model.
- We apply a novel LSTM variant LSTM-S that incorporates two extra semantic gates to capture the semantic sequential correlation of check-in sequences. Instead of directly utilizing semantic sequences to obtain the correlation, we explicitly model the semantic irrelevancy as the semantic interval, and then take the additional semantic interval sequence as the input of semantic gates.
- In order to achieve a more accurate recommendation, we cluster users into different groups using the K-means method based on the phenomenon that if users have similar preferences, their mobility patterns will tend to be similar, then fit each group into the corresponding model we proposed to boost our method's performance.
- We evaluate the proposed method by detailed experiments on a large-scale Foursquare dataset and the experimental result demonstrates the superiority of using our method.

II. Related Work

In this section, we first make a literature review of POI recommendation, then we discuss the relationship between our proposed model and previous work.

A. POI Recommendation.

POI recommendation is a significant task in LBSNs and most related to our work, it focuses on recommending POIs based on each user's check-in history with geographic information and no explicit rating information. In order to improve the performance of POI recommendation, lots of recent work has tried to mine more information from spatial context, temporal context, social context and category context.

Spatial Context. Most recent studies [10] [11] [12] have tried to incorporate the spatial context of check-ins due to the strong correlation between check-in activities and geographical distance. In order to improve the recommendation accuracy, Liu et al. [10] propose a new method to incorporate location-level influence and region-level influence based on Weighted Matrix Factorization (WMF) to deal with the data sparsity problem. For the sake of capturing the spatial clustering phenomenon (i.e., POIs visited by same users are likely to be in the same region [13]), Lian et al. [11] integrate geographical impacts through users' activity area modeling and geospatial impact propagation. Ye et al. [12] utilize a Poisson Factor Model based on geographical information to jointly learn both geographical preferences and interest preferences for users because these two preferences will interact with each other.

Temporal Context. On the one hand, different users may behave differently at different time, on the other hand, different POIs have various business hours and peak periods. Therefore it's pivotal to utilize temporal context to improve the accuracy of recommendation. Yuan et al. [14] recommend POIs for a given user at a specified time in a day through a new collaborative recommendation model that is able to incorporate temporal information. Gao et al. [15] use a time-enhanced Matrix Factorization (MF) model based on the observation that users' check-in behaviors vary from time. Zhao et al. [3] propose a time-aware trajectory embedding model to incorporate periodical temporal preference and dynamic personal preference based on distributed representation learning.

Social Context. Li et al. [16] use a two-step POI recommendation framework that (i) learns potential locations from users friends and (ii) incorporates potential locations into WMF to overcome the cold-start problem. Zhang et al. [17] integrate social factors into geographic information by using a friend-based collaborative filtering, where the similarity between friends is computed based on the distance between their residences.

Category Context. Zhang et al. [18] leverage the cumulative distribution of categories calculated by users' check-in frequency as the category impact in recommendation. Li et al. [16] calculate the categories' weights by users' preferences to affect the recommended score obtained by MF.

B. Connection to Prior Work.

We focus on successive POI recommendation to predict the most likely POI in the next step.

Earlier next location recommendation methods rely on modeling user mobility pattern and sequence correlation between POIs by Markov Chain [6], [19], but it is questionable because the sequential transitions are subject to first-order transitions owing to sparse data and computational complexity, also the particular structure of Markov Chain leads to the limitation of capturing longer sequential context. Another straightforward approach to predict the most likely visit location in the next step is taking advantage of distributed representation learning, which incorporating users' time periodicity factor [3]. Nevertheless, this method only models the periodicity of user behavior in the time level and neglects other dependency between check-ins.

Thus, better approaches based on RNN proposed due to its ideal effect of modeling sequential data, like language modeling and next-basket recommendation. In order to derive a better solution, many works have been presented. Liu et al. [8] propose a model called ST-RNN to model the temporal context and spatial context for next location recommendation by utilizing a time window in each RNN cell as well as time-specific and distance-specific transitions matrices.

However ordinary RNN model proposed before cannot ideally play the function of time-specific and distance-specific transitions matrices because of partitioning intervals into discrete bins to adapt the data sparsity problem. Furthermore, it's difficult to select the proper time window widths for different applications and the gradient vanishing problem will also arise.

Gate mechanism introduced by LSTM solves the gradient vanishing problem and can improve the accuracy of POI recommendation. Kong et al. [9] propose a Spatial-Temporal LSTM model which naturally combines spatial-temporal influence into LSTM to mitigate the problem of data sparsity and then employ an encoder-decoder manner which models the contextual historic visit information.

Yet, most of the previous successive POI recommendation methods focus only on the temporal sequential context and spatial sequential context, ignoring the semantic sequential correlation that can boost the prediction performance, we use a semantic sequential correlation calculation model to measure the correlation between adjacent check-ins' categories to get the semantic sequential correlation, and then we apply a LSTM-S model with two additional semantic gates to incorporate the semantic sequential correlation calculated before to boost our prediction performance.

III. The Proposed Model

Before describing our model for the next POI recommendation, we start with a brief introduction to describe our model's architecture as a whole. Then, we elaborate two parts of our model in detail, one is the semantic sequential correlation calculation model, and another is the LSTM-S model which incorporates the semantic sequential correlation as the extra input. In the end, we introduce our improvement method via K-means cluster.

A. Model Introduction

For a better representation, we first introduce some relevant concepts.

- **POI.** A POI is a geographic location (such as a cinema or a cafe) that has specific functions to satisfy users' requirements. In our proposed model, POI contains three attributes: unique identifier, geographic information (latitude and longitude), and semantic information (category information).
- **Check-in.** A check-in record is represented as a five-tuple (u, v, t, d, c), which means that a user u visits a POI v with geographic information d as well as category c at time t.
- **Check-in Sequence.** A check-in sequence consists of a user's consecutive check-ins. The historical check-ins of this user are utilized to be the contextual information of predicting the next POI he/she is about to visit in the next step.
- **Semantic Sequence.** A semantic sequence consists of the category information extracting from check-in sequence and meanwhile preserves its chronological order.
- **Semantic Sequential Correlation.** A semantic sequential correlation is a correlation sequence containing correlation between all adjacent categories of a semantic sequence.

Assume that there is a set of m users $U = \{u_1, u_2, u_3, ..., u_m\}$ and a set $V = \{v_1, v_2, v_3, ..., v_n\}$ contains n POIs. Each user in U has a unique historical check-in sequence $H_i^u = \left\{v_{t_1}^u, v_{t_2}^u, v_{t_3}^u, ..., v_{t_{i-1}}^u\right\}$, where t_{i-1} is the last time this user checked in, $i-1$ is the length of this user's historical check-in trajectory, and $v_{t_i}^u$ indicates a user u visit POI v at time t_i. We define the semantic sequence $C_i^u =$

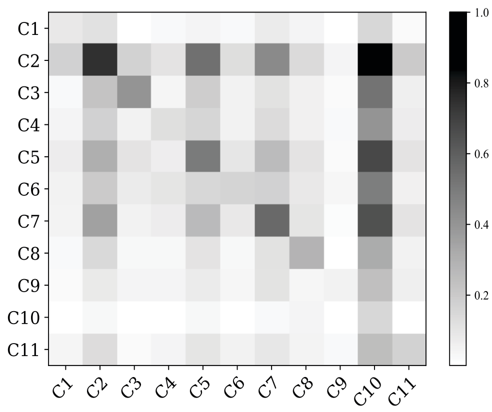

Fig. 2. A heat map of categories' relevances. Categories = C_1: Bar, C_2: Catering, C_3: Education, C_4: Entertainment, C_5: Events, C_6: Exercise, C_7: Shop, C_8: Traffic, C_9: Travel, C_{10}: Residence, C_{11}: Others

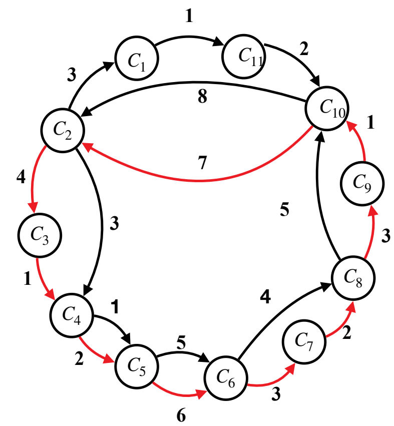

Fig. 3. An example of semantic correlational graph based on two semantic sequences: $C_2 \xrightarrow{4} C_3 \xrightarrow{1} C_4 \xrightarrow{2} C_5 \xrightarrow{6} C_6 \xrightarrow{3} C_7 \xrightarrow{2} C_8 \xrightarrow{3} C_9 \xrightarrow{1} C_{10} \xrightarrow{7} C_2$ and $C_2 \xrightarrow{3} C_4 \xrightarrow{1} C_5 \xrightarrow{5} C_6 \xrightarrow{4} C_8 \xrightarrow{5} C_{10} \xrightarrow{8} C_2 \xrightarrow{3} C_1 \xrightarrow{1} C_{11} \xrightarrow{2} C_{10}$, the numbers on the arrows represent the time intervals

$\left\{c^u_{t_1}, c^u_{t_2}, c^u_{t_3}, ..., c^u_{t_{i-1}}\right\}$ is extracted from H^u_i, in terms of the semantic information of each $v^u_{t_i}$. Then our problem can be defined as: given a user u and his/her previous $i-1$ check-in records $\left\{v^u_{t_1}, v^u_{t_2}, v^u_{t_3}, ..., v^u_{t_{i-1}}\right\}$, we recommend a POI that most likely to be visited in the next step for u with the help of mining the semantic sequential correlation from C^u_i.

In order to integrate the semantic sequential correlation into the next POI recommendation problem to boost the performance, the measurement of semantic sequential correlation and the modeling of check-in sequence with the semantic sequential correlation have become the urgent problems to be solved.

Thus we divide the structure of our model into two parts: semantic sequential correlation calculation model and the LSTM-S model for the next POI recommendation, to tackle the problems mentioned before. Also the improvement method is introduced.

Semantic Sequential Correlation Calculation Model: In this part, we measure the semantic sequential correlation through traversing a semantic correlational graph which incorporates the time intervals' influence to the semantic sequential correlation calculation.

LSTM-S for the Next POI Recommendation: In this part, We apply a LSTM variant incorporating two additional semantic gates to take the semantic sequential correlation calculated before to predict the most likely POI in the next step.

Improvement Method: For the purpose of improving the prediction performance of the next POI recommendation, we use an extra part to illustrate the improvement method of clustering users into different groups according to their preferences.

B. Semantic Sequential Correlation Calculation Model

The dataset used in our paper contains 426 fine-grained location categories which leads to a huge prediction space, in order to make check-ins' categories more representative, we aggregate them into 11 types inspired by [19]: Bar, Catering, Education, Entertainment, Events, Exercise, Shop, Traffic, Travel, Residence and Others. When we consider the next POI recommendation problem in the semantic level, the correlation between categories reflect the users' mobility patterns. Fig. 2 plots the transition probabilities between our categories.

The original method of calculating the semantic correlation is mainly based on statistics, in view of the hypothesis that the higher the transition probability between two categories in the semantic sequences, the higher the semantic correlation of the two categories. However, it is inadequate to calculate semantic correlation only considering the semantic transition probabilities. For example, there will be large errors in the calculation results when the statistical samples are insufficient.

Inspired by [2] that time intervals between users' check-ins are of significant importance in capturing the relations of users' actions, similarly, when we model users' check-ins in the semantic level, we also incorporate the time intervals' influence when capturing the semantic sequential correlation. In detail, we first leverage the semantic sequences to construct a semantic correlational graph G, then categories in the semantic sequences are represented as nodes in the semantic correlational graph. And the transition between each two categories is represented as a directed edge from the previous node to the next node. We also call the directed edge a semantic connected path because we only consider the influence between two adjacent categories, same as the method of calculating the semantic correlation using statistics. Finally, the time interval of each transition is expressed as the weight of the directed edge. Fig. 3 shows an example of a semantic correlational graph.

In order to make our calculation process more simple and clear, we introduce two rules of the semantic correlational graph, explained as follows:

- If the weights of all the semantic connected paths between

two adjacent nodes are equal, then the more connected paths between the two nodes, the greater their semantic correlation, and vice versa.
- If the number of the semantic connected paths between two adjacent nodes is equal, then the smaller the weight of the semantic connected path between two nodes, the greater their semantic correlation, and vice versa.

Based on the above rules, we traverse the semantic correlational graph, then the calculation of the semantic sequential correlation is detailed below, and the pseudocode is given in Algorithm 1.

a) Given a check-in sequence set S, each element in this set can be defined as: $S^u = \{v_1^u, v_2^u, v_3^u, ..., v_l^u\}$, where u represents the user identifier, and S^u is the check-in sequence of u, l is the length of check-in sequence. For each sequence, we first extract the semantic sequence from it and recorded as $C^u = \{c_1^u, c_2^u, ..., c_l^u\}$, then we split C^u into one-to-one sequences set O^u: $\{< c_1^u, c_2^u >, < c_2^u, c_3^u >, ..., < c_{l-1}^u, c_l^u >\}$, the angle brackets are used to emphasize the order of the sequence.

b) For each one-to-one sequence $< c_i^u, c_{i+1}^u >$ in O^u, we traverse G to get the number of directed edges n from c_i^u to c_{i+1}^u and the weight of each directed edge w_e $(1 \leq e \leq n)$ representing the nomarlized consumed time. Since we only consider the influence of the adjacent nodes, all the length of our semantic connected paths is 1, so we calculate the total length of the weighted semantic connected paths from c_i^u to c_{i+1}^u using the following formula:

$$L(c_i^u, c_{i+1}^u) = \sum_{e=1}^{n} w_e \tag{1}$$

Then the average length of the weighted path is calculated:

$$\overline{L(c_i^u, c_{i+1}^u)} = \frac{1}{n} L(c_i^u, c_{i+1}^u) \tag{2}$$

c) In view of the relevant rules we proposed before, we obtain that the average length of the weighted path is inversely proportional to the semantic correlation, while the number of weighted paths is proportional to the semantic correlation, thus the semantic correlation is calculated as follows:

$$Rel(c_i^u, c_{i+1}^u) = \frac{\log_2(n+1)}{\log_2(n+1) + \overline{L(c_i^u, c_{i+1}^u)}} \tag{3}$$

After we get the semantic correlation of each one-to-one sequence, we store these semantic correaltions in chronological order to get each user's semantic sequential correlation Seq^u from his/her check-in sequence, then store it to the semantic sequential correlation set Seq.

Algorithm 1 Semantic Sequential Correlation Calculation Algorithm

Input:
Semantic correlational graph G, check-in sequence set S generated by all the users;
Output:
Semantic sequential correlation set Seq;
for each $S^u \in S$ **do**
  Extract semantic sequence C^u;
  Split C^u into one-to-one sequences;
  Store one-to-one sequences into set O^u;
  for each one-to-one sequence $< c_i^u, c_{i+1}^u > \in O^u$ **do**
    Traversing G to calculate the number of directed edges n from category c_i^u to category c_{i+1}^u and the weight of each directed edge w_e;
    if $n > 0$ **then**
      Calculating the semantic correlation $Rel(c_i^u, c_{i+1}^u)$ through Eq.(3);
    else
      $Rel(c_i^u, c_{i+1}^u)$ = 0;
    end if
    Store $Rel(c_i^u, c_{i+1}^u)$ into semantic sequential correlation Seq^u;
  end for
  Store Seq^u into Seq;
end for
return Seq;

C. LSTM-S for the Next POI Recommendation

Normal LSTM [20] is an improved RNN model introducing the gate mechanism, which can avoid the vanishing gradient problem. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. Given the input x_t at time t, the output of LSTM hidden layer h_t is computed by following functions:

$$i_t = \sigma(x_t W_{xi} + h_{t-1} W_{hi} + b_i) \tag{4}$$

$$f_t = \sigma(x_t W_{xf} + h_{t-1} W_{hf} + b_f) \tag{5}$$

$$o_t = \sigma(x_t W_{xo} + h_{t-1} W_{ho} + b_o) \tag{6}$$

$$\widetilde{C} = tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \tag{7}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \widetilde{C} \tag{8}$$

$$h_t = o_t \cdot tanh(c_t) \tag{9}$$

We describe some definitions of these functions as follows: the first hidden layer output h_0 is equal to zero, $\sigma(\cdot)$ means the logistic sigmoid function, meanwhile $tanh(\cdot)$ is the hyperbolic tangent function. i_t, f_t, o_t reperesent the input, forget, and output gate of the t-th object respectively, these gates keep LSTM cell update, retaine, and discard data over time, and in particular, input gate i_t is utilized to choose the input data, forget gate f_t is utilized to decide which input data should be forgotten, output gate o_t is leveraged to decide whether to produce current state. Its only two inputs: x_t and h_t represent the input feature vector and the hidden output vector respectively, b_i, b_f, b_o represent the corresponding bias of each gate. Weight matrices W_{xi}, W_{xf}, W_{xo}, W_{xc} are used to connect inputs with different gates as well as the candidate cell memory $\widetilde{C}$. The cell update equation has two parts, one is the updated previous cell state which has passed through the forget gate, another is a new input state influenced by the candidate cell memory $\widetilde{C}$ through an element-wise operation. And finally hidden layer output is defined as an element-wise

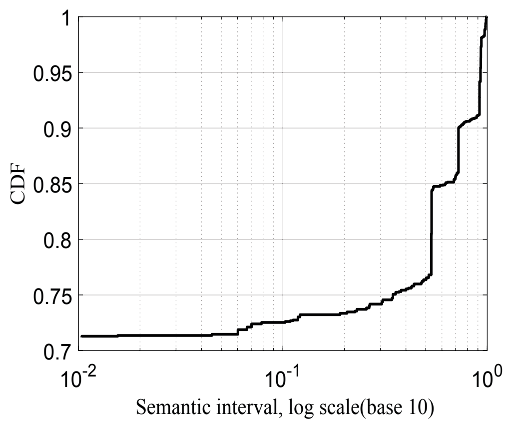

Fig. 4. CDF of semantic intervals between consecutive check-ins

operation between output gate and cell state's $tanh$ nonlinear transformation. In this standard model, x_t is used to learn the user's current short-term interest, and c_{t-1} contains users' long-term preference.

Nevertheless, the data sparsity problem will cause the cell of LSTM fails to learn well functioned gate, and through the previous description, we can see that category contexts will reflect users' behavior patterns in the semantic level. When we describe users' check-in behavior in the semantic level, the less compact the semantic correlation is, the less likely it is to happen. Thus we visualize the semantic intervals between all check-in sequences, as shown in Fig. 4, the horizontal axis represents the semantic interval values of adjacent check-ins, which means the size of semantic irrelevancy. We can see that about 72.5% semantic intervals are less than 0.1, which clearly reflects the phenomenon that users' continuous check-in actions have strong semantic correlation.

Based on the above analysis, we apply a novel structure inspired by [2] that utilize two additional semantic gates to control the influence of x_t on the current recommendation and c_{t-1} on the latter recommendation. Specifically, we take the semantic interval sequence that represents the irrelevancy between connected check-ins as the extra input, and then compared with the LSTM variant proposed before [9], which implicitly leverages the additional sequence by influencing the current input, our method leverage two additional semantic gates to control the influence of not only the last check-in, but also the cell state, finally we improve our model by coupled input and forget gates. The mathematical expressions are listed as follows:

$$S1_t = \sigma(x_t W_{xs_1} + \sigma(\Delta s_t W_{s_1}) + b_{s_1}) \tag{10}$$

$$S2_t = \sigma(x_t W_{xs_2} + \sigma(\Delta s_t W_{s_2}) + b_{s_2}) \tag{11}$$

$$\widehat{c_t} = (1 - i_t \cdot S1_t) \cdot c_{t-1} + i_t \cdot S1_t \cdot \widetilde{C} \tag{12}$$

$$c_t = (1 - i_t) \cdot c_{t-1} + i_t \cdot S2_t \cdot \widetilde{C} \tag{13}$$

$$o_t = \sigma(x_t W_{xo} + h_{t-1} W_{ho} + b_o) + \Delta s_t W_{s_o} \tag{14}$$

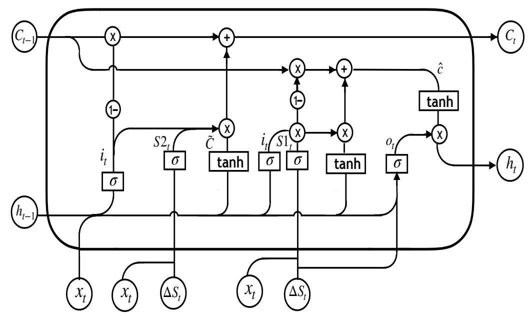

Fig. 5. Structure of semantic sequential correlation based LSTM

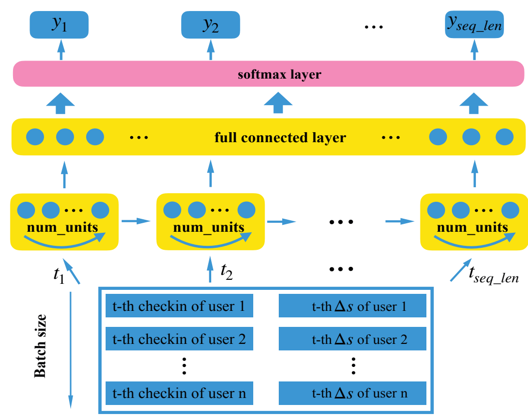

Fig. 6. Structure of LSTM-S model architecture

$$h_t = o_t \cdot tanh(\widehat{c_t}) \tag{15}$$

Where $\Delta s \in \mathbb{R}^d$ is a vector with dimension d, representing the semantic impact factor. Δs_t denote the vector representations of semantic intervals between x_{t-1} and x_t. In Eq.(10) and Eq.(11), $S1_t$ and $S2_t$ are two additional semantic gates: $S1_t$ is utilized to control the short-term preferences by modified the influence of the last check-in data on current POI recommendation, and $S2_t$ is used to preserve semantic intervals to capture users' long-term preferences for subsequent recommendation. Unlike the original LSTM directly using a c_t to transfer information to the hidden state in Eq.(9), in Eq.(12) we utilize a new cell state $\widehat{c_t}$ to store the modified $\widetilde{C}$ which is filtered by i_t and $S1_t$, then pass it to the hidden state to affect the current recommendation. While the original c_t is leveraged to receive the $S2_t$ that stores the semantic intervals, and then pass it to c_{t+1}, c_{t+2}...to save the long-term preference of users.

Note that W_{s_1} is less than 0 in Eq.(10), considering when this model encounters a large semantic interval, $S1_t$ will be relatively small, so in Eq.(12), the current input x_t will be filtered by a smaller $S1_t$, then causing less influence on current recommendation, and vice versa. However W_{s_2} does not have this limitation because $S1_t$ is only used to store semantic intervals to model long-term preference. Inspired by [21], we utilize $1 - i_t \cdot S1_t$ to replace the previous forget gate in Eq.(12) due to $S1_t$'s filter function, and replace with $1 - i_t$ in Eq.(13) due to $S1_t$'s storage function.

The LSTM architecture is shown in Fig. 5, and the model architecture is shown in Fig. 6. Note that we just utilize a simple model architecture to test the performance of our method. The parameters of model architecture are explained as follows: the num_units represents the number of hidden units, we set the number to 128 which most often used. The input of the model is a concatenation of batch size check-in id and semantic irrevelance Δs, seq_len is the length of check-in sequence, we cycle seq_len times for each check-in sesquence. The output of this model pass into a softmax layer and the probability of each check-in id is captured after that.

D. Improvement Method

Different from [22] that alleviates the data sparsity problem through discovering users' trajectories' hierarchical properties, we directly mine users' preference similarity to do that. Since users' check-in behaviors are usually affected by their own preferences, that is, if users have similar preferences, their mobility patterns will tend to be similar. So in this section, we distinguish users' behavior patterns by clustering, then each cluster will transfer to a corresponding model we proposed to avoid the low accuracy caused by putting all the check-ins into a single model.

Owing to the rich semantic information, we still learn users' distinctive preferences in the semantic level, thus the preferences of users are exploited in terms of their check-in category distributions inspired by [19]. Specifically, from getting the category information of each check-in location, we let N_c^u be the total number of check-ins of user u on category c, then the preference of user u is defined as a distinctive preference feature vector $\vec{p^u}$, where $\vec{p^u}$ = (p_1^u, p_2^u,...,p_r^u), and r is the total number of distinct categories. The vector's item p_c^u is defined as:

$$p_c^u = \frac{N_c^u}{\sum_{i=1}^{r} N_i^u} \tag{16}$$

After get all the preference vectors of users, the K-means method is utilized to cluster preference vectors of all the users. As the vector of each user indicates their unique preference so the centroid of each cluster represents the typical features of each group. When the users are clustered, the corresponding model can better fit their mobility patterns. In Section IV, we will discuss how the prediction performance of the next POI recommendation is influenced by using users' preferences.

IV. Experimental Evaluation

In this section, we give an overview of the experimental configuration, including dataset, evaluation criteria, baseline methods, and experimental setting, then construct the experiments to evaluate the performance of our proposed model.

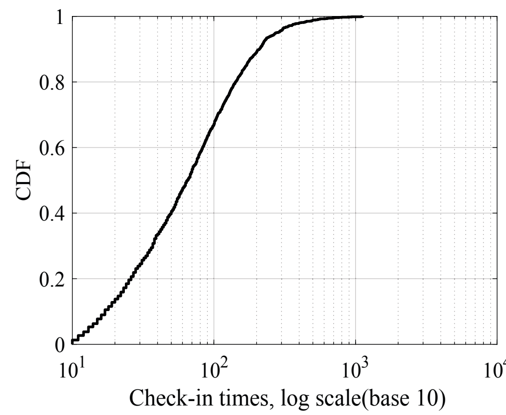

Fig. 7. CDF of check-in times of all the users in our dataset

A. Dataset

Our experimental dataset includes long-term (about 9 months from January 2013 to September 2013) check-in data collected from Foursquare, provided by [1]. It contains 33,278,683 check-ins by 22,401 users on 3,680,126 venues (in 60 cities in the United States). Those 60 cities are the most checked 60 cities by Foursquare users in the United States, each of which contains at least 10K check-ins, thus it is valuable and representative for the study of human mobility. In the preprocessing stage, we remove users whose check-in count is fewer than 10 and check-ins generated in non-residential cities (We mark the city where each user check-in most often as his/her residential city to make a more accurate recommendation). After preprocessing, our dataset contains 19,498 users with 1,172,753 check-ins. Fig. 8 shows the cumulative distribution of users' check-in times.

B. Evaluation Criteria

We use two kinds of criteria to evaluate our model and the baselines. One is $Recall@k$ which can be defined as:

$$Recall@k = \frac{n_{hit}@k}{n_{test}} \tag{17}$$

where $n_{hit}@k$ is the number of test cases that include groudtruth at top-k and n_{test} stands for the number of total test cases.

Another evaluation criteria is MRR (Mean Reciprocal Rank), $MRR@k$ is defined as the average of reciprocal ranks of groudtruth in the recommendation list, and set the reciprocal rank of groundtruth to 0 if the rank is above k. Based on previous work [8], we set k to 10 if not specified as a great value of k is usually ignored for a typical top-k recommendation.

C. Baseline Methods

Our proposed method is a semantic sequential correlation based LSTM framework for next POI recommendation, which incorporates semantic interval sequences as the extra input.

TABLE I
PERFORMANCE IN TERMS OF RECALL@10 AND MRR@10 FOR ALL METHODS WITH VARYING EPOCH VALUES

Method	Recall@10				MRR@10			
	epoch=20	*epoch=40*	*epoch=60*	*epoch=80*	*epoch=20*	*epoch=40*	*epoch=60*	*epoch=80*
BPR	0.2403	0.2917	0.3215	0.3390	0.1597	0.1947	0.2080	0.2105
LSTM	0.0256	0.6522	0.6211	0.7667	0.0060	0.4743	0.3653	0.5506
PLSTM-T	0.0311	0.5000	0.5533	0.6033	0.0131	0.1877	0.2484	0.2759
PLSTM-S	0.0211	0.5022	0.7278	0.8022	0.0063	0.2863	0.5420	0.6473
PLSTM-D	0.3022	0.4022	0.4444	0.4711	0.0883	0.1480	0.1775	0.2010
LSTM-S	0.3611	0.6256	0.8000	0.8767	0.1167	0.3479	0.5594	0.6886

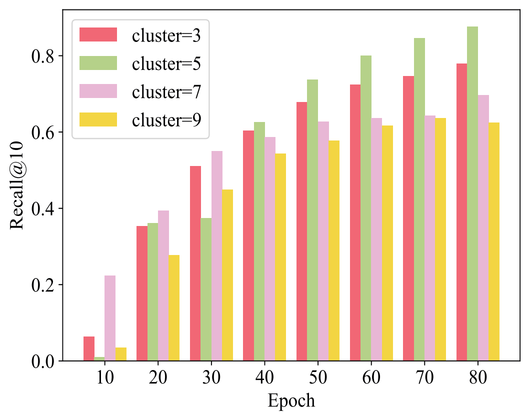

Fig. 8. Performance of Recall@10 with different cluster numbers in different epoch number using our method

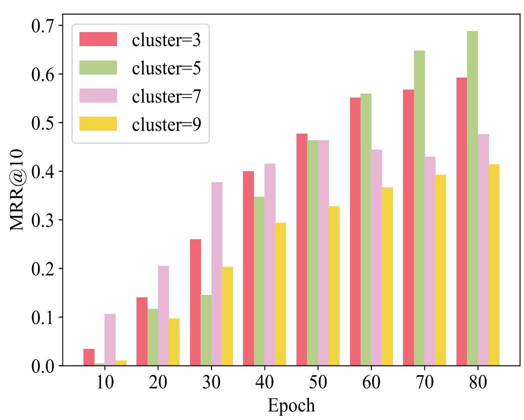

Fig. 9. Performance of MRR@10 with different cluster numbers in different epochs using our method

To prove the validity of the proposed model, we compare our model with the following prediction methods:

- BPR [23] This is the abbreviation of Bayesian Personalized Ranking, which is a learning algorithm for collaborative filtering, frequently used in recommendation with implicit feedback.
- LSTM [20] This is a general variant of RNN, which is composed of a memory cell, an input gate, an output gate and a forget gate.
- Phased LSTM [24] is a state-of-the-art LSTM architecture for modeling event based on sequential data that adds an additional time gate, in the following sections, we will use PLSTM-T to represent it.
- PLSTM-S. This is a variant of Phased LSTM that takes our semantic interval sequences as the extra input.
- PLSTM-D. This is a variant of Phased LSTM that takes distance interval as the input of time gate.

D. Experimental Setting

We sort each user's check-in records according to timestamp order, and then taking the first 80% as the training set, the remaining 20% for the test set. The learning rate of the proposed model is initialized as 0.01 and the parameters in our model are optimized by AdaGrad.

E. Performance Evaluation

Different k Performance. Firstly, we cluster the users into k clusters based on their preference vectors and test k = 3,5,7,9. The performance is shown in Fig. 8 and Fig. 9. From these figures, we can observe that: (1) Both the performance of two evaluation criteria increase with epoch numbers. (2) With the increasing of epoch numbers, the superiority of k equals 5 appears continuously. Meanwhile, the prediction performances are superior to other k values' performances on almost all the number of epoch when k = 5. (3) The performance reaches the highest when k = 5 and epoch = 80, which shows the effectiveness of user clustering when k =5. (4) The performance decreases when further increase k to 7 and 9.

So in the following experiment, we set the number of clusters to 5.

Overall Performance. We report the comparison results between our proposed model and other baselines, the overall performances are shown in Table I. From the statistics, we can conclude that:

- The performance of BPR increase steadily but no significant increase, which shows the role of LSTM in sequence recommendation of check-ins.
- The performance of the original LSTM shows great volatility, in contrast, other methods increase steadily with the epoch increasing, which shows the importance of adding context information.

- Comparing with PLSTM-T incorporating temporal information and PLSTM-D incorporating spatial information, PLSTM-S achieves a better result when epoch number is larger than 20, it proves the effectiveness of the idea of combining semantic sequential correlation.
- Without considering the LSTM with strong fluctuation, PLSTM-S and LSTM-S are superior to other methods with a certain large margin which demonstrates the generality of incorporating the semantic sequential correlation.
- LSTM-S achieves much better prediction performance than PLSTM-S in all evaluation criteria which shows the validity of our LSTM structure with two additional semantic gates to mitigate the data sparsity problem.

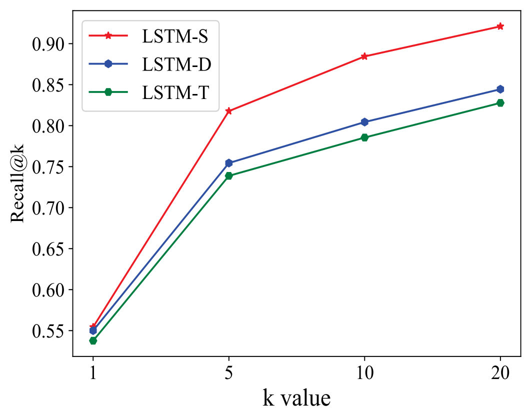

Fig. 10. Performance comparison using $Recall@k$ on our dataset with varying the values of k in LSTM-S, LSTM-D, LSTM-T

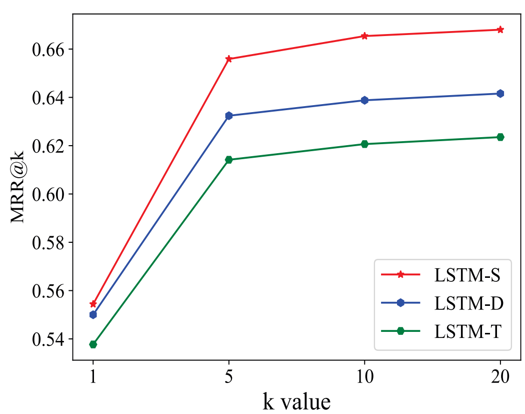

Fig. 11. Performance comparison using $MRR@k$ on our dataset with varying the values of k in LSTM-S, LSTM-D, LSTM-T

Different Factor Performance. We do experiments that consider three different factors representing temporal context, spatial context and semantic context separately. The experimental results are presented in Fig. 10 and Fig. 11. From these figures, we observe that: the performance will rise with the increasing of k values, and semantic context (LSTM-S) show great superiority than temporal context (LSTM-T) and spatial context (LSTM-D) on all k values, which demonstrates the importance of the semantic context.

V. Conclusion

This paper proposes a novel semantic sequential correlation based LSTM model in terms of the semantic interval sequences generated from semantic sequential correlation, obtained by our semantic sequential correlation calculation model. By introducing this semantic factor into the additional semantic gates, the newly added semantic context alleviate the data sparsity problem and boost the prediction performance of the next POI recommendation. The experimental results prove the effectiveness of our proposed method.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No. 61572165).

References

[1] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. Nationtelescope: Monitoring and visualizing large-scale collective behavior in lbsns. *Journal of Network and Computer Applications*, 55:170–180, 2015.

[2] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3602–3608, 2017.

[3] Wayne Xin Zhao, Ningnan Zhou, Aixin Sun, Ji-Rong Wen, Jialong Han, and Edward Y Chang. A time-aware trajectory embedding model for next-location recommendation. *Knowledge and Information Systems*, pages 1–21, 2017.

[4] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254. ACM, 2017.

[5] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2):75–79, 2007.

[6] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, pages 2605–2611, 2013.

[7] Qiujian Lv, Yuanyuan Qiao, Nirwan Ansari, Jun Liu, and Jie Yang. Big data driven hidden markov model based individual mobility prediction at points of interest. *IEEE Transactions on Vehicular Technology*, 66(6):5204–5216, 2017.

[8] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, pages 194–200, 2016.

[9] Dejiang Kong and Fei Wu. Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction. In *IJCAI*, pages 2341–2347, 2018.

[10] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 739–748. ACM, 2014.

[11] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 831–840. ACM, 2014.

[12] Bin Liu, Hui Xiong, Spiros Papadimitriou, Yanjie Fu, and Zijun Yao. A general geographical probabilistic factor model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1167–1179, 2015.

[13] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334. ACM, 2011.

[14] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.

[15] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100. ACM, 2013.

[16] Huayu Li, Yong Ge, Richang Hong, and Hengshu Zhu. Point-of-interest recommendations: Learning potential check-ins from friends. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 975–984. ACM, 2016.

[17] Jia-Dong Zhang and Chi-Yin Chow. igslr: personalized geo-social location recommendation: a kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 334–343. ACM, 2013.

[18] Jia-Dong Zhang and Chi-Yin Chow. Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 443–452. ACM, 2015.

[19] Jihang Ye, Zhe Zhu, and Hong Cheng. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 171–179. SIAM, 2013.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2017.

[22] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. Trajectory-user linking via variational autoencoder. In *IJCAI*, pages 3212–3218, 2018.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press, 2009.

[24] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, pages 3882–3890, 2016.