A Context Awareness Hierarchical Attention Network for Next POI Recommendation in IoT Environment

Xuebo Liu, Jingjing Guo * and Peng Qiao *

Shanxi Information Industry Technology Research Institute Co., Ltd., Taiyuan 030012, China
* Correspondence: jane_guojingjing@163.com (J.G.); sxszdh@126.com (P.Q.)

Abstract: The rapid increase in the number of sensors in the Internet of things (IoT) environment has resulted in the continuous generation of massive and rich data in Location-Based Social Networks (LBSN). In LBSN, the next point-of-interest (POI) recommendation has become an important task, which provides the best POI recommendation according to the user’s recent check-in sequences. However, all existing methods for the next POI recommendation only focus on modeling the correlation between POIs based on users’ check-in sequences but ignore the significant fact that the next POI recommendation is a time-subtle recommendation task. In view of the fact that the attention mechanism does not comprehensively consider the influence of the user’s trajectory sequences, time information, social relations and geographic information of Point-of-Interest (POI) in the next POI recommendation field, a Context Geographical-Temporal-Social Attention Hierarchical Attention Network (CGTS-HAN) model is proposed. The model extracts context information from the user’s trajectory sequences and designs a Geographical-Temporal-Social attention network and a common attention network for learning dynamic user preferences. In particular, a bidirectional LSTM model is used to capture the temporal influence between POIs in a user’s check-in trajectory. Moreover, In the context interaction layer, a feedforward neural network is introduced to capture the interaction between users and context information, which can connect multiple context factors with users. Then an embedded layer is added after the interaction layer, and three types of vectors are established for each POI to represent its sign-in trend so as to solve the heterogeneity problem between context factors. Finally reconstructs the objective function and learns model parameters through a negative sampling algorithm. The experimental results on Foursquare and Yelp real datasets show that the AUC, precision and recall of CGTS-HAN are better than the comparison models, which proves the effectiveness and superiority of CGTS-HAN.

Keywords: context awareness; attention network; dynamic user preferences; next POI recommendation; IoT

1. Introduction

The rapid increase in the number of sensors in IoT environment has resulted in the continuous generation of massive and rich data in LBSN, which has greatly promoted the development of POIs recommendations. It is an important and challenging task to understand the mobility of users and recommend the next POI to users. The research on POIs recommendation considers user check-in data as a whole, which only considers the relationship between target users and points of interest but ignores the relationship between points of interest. In fact, there is a strong correlation between the user’s current POI and the POI to be accessed next. For example, when users obtain off work at night, they may go to restaurants, bars and other places instead of traveling. Different from ordinary POI recommendations, the next POI recommendation is to recommend the POI to visit next based on the user’s historical trajectory information and the current POI location. The next POI recommendation can enhance the user’s travel experience but also help real economists push advertisements to the target user groups. Because the next
POI recommendation is of great significance to users and enterprises, the improvement of the next POI recommendation system is popular in the academic community, and much research focuses on enhancing the recommendation performance of the next POI.

Modeling according to the check-in sequence of users is the basic work for the next POI recommendation, and POIs tend to have certain correlations among themselves [1–4]. Some previous studies adopted Markov Chain [1,5–7] matrix decomposition [2,3], tensor decomposition [8,9] or transformation model [10,11] to solve the next POI recommendation problem. In recent years, the Recurrent Neural Network (RNN) model in Deep Neural Networks [12–15] has shown a good recommendation effect in processing sequence data. MST-RNN [16] exploits the duration time dimension and semantic tag dimension of POIs in each layer of the neural network. Attention Network [17–20], as a branch of RNN, has strong recommendation performance when applied to a recommendation. In real life, the check-in sequences of users on different dates will have different time characteristics in their historical trajectories. For example, users usually check in at POIs near the company on weekdays and go to shopping malls and tourist attractions on weekends. However, these studies did not explore the diversity of time sequence characteristics.

When the POI recommendation systems encounter new users, fewer user trajectories, or users in a new area, the cold-start problem may occur, and the recommendation system cannot effectively recommend the next POI for the user. Moreover, how to extract valuable contextual information [21] (e.g., geographical location, time, social relations) from the massive data generated by IoT sensors is particularly vital.

In order to enhance the accuracy of the next POI recommendation, this paper proposes the next POI recommendation model embedded with user context check-in information, which comprehensively considers the influence of geographical, temporal and social factors. This paper studies the next point of interest recommendation problem, and the main contributions are as follows:

1. We propose the next POI recommendation model named CGTS-HAN. It uses the attention mechanism and the Bi-directional Long Short-Term Memory (BiLSTM) to establish a geo-temporal social attention network to learn user check-in sequences, which can simultaneously capture the user’s social relationships, the temporal dependence of sequence patterns, and the geographical relationships between points of interest. Taking the influence of geography, time and social factors into account and embedding the user context check-in information can effectively reduce the problem of a cold start.

2. We propose a recommendation algorithm suitable for a hierarchical model system. Incorporating the context interaction layer into CGTS-HAN, a feedforward neural network is used to learn feature intersections to describe the interaction effects of users and contexts.

3. Combining the context interaction layer and CGTS-HAN model, a Co-attention Network is developed to learn the dynamic preferences of users so that CGTS-HAN can distinguish user preference degrees in its historical check-in trajectories.

4. Experiments were conducted on two real-world datasets. The experimental results demonstrate that CGTS-HAN achieves better performance than other baseline comparison models in terms of AUC, precision and recall.

2. Related Work

The influence of sequential factors. Most next Points-of-Interest (POIs) recommendations rely on the sequence correlation in the user check-in. Wen et al. [5] calculated the global and individual transition probabilities between clusters according to the user’s check-in sequence, then used multi-order Markov chains to discover and rank subsequent clusters, and finally combined with individual preferences to generate a ranking list. Although it can model time series well, the correlation between POIs is not high. Zhang et al. [1] used N-order Markov chains to implement POIs recommendations for higher-order sequences. The authors recommend coarse-grained areas to users, such as location recommendations.
based on city streets, rather than user personalization. The recommended points of interest are not targeted. Kong et al. [15] aimed to discover users’ uncertain check-in points of interest; they extended Skip Gram to capture user preference transitions and then predicted the next POI for uncertain check-in users. Wang et al. [22] propose a location-aware POI recommendation system that user preferences by using user history trajectory and user review information. However, the collaborative filtering algorithm used in the article is likely to have a cold start problem.

The influence of geographical factors. Debnath et al. [7] mined sequential patterns from each user’s check-in location, then used Markov chains to construct transition probability matrices and combined them with spatial influences to generate space-aware location recommendations. Liu et al. [2] improved the accuracy of POIs recommendations by using the conversion mode of the user’s preference for the POIs category. Feng et al. [11] established a hierarchical binary tree according to the physical distance between POIs to reflect the influence of geographical location. However, previous efforts mainly consider the location information of check-in points as a whole and ignore their temporal relation. Using the information from users’ location history, Yang et al. [23] proposed a location-aware POI recommendation system that uses information from users’ location history and models user preferences based on their reviews. It aims to solve the user’s POIs recommendation in new regions and cities without considering the impact of time context and other time dynamics.

The influence of time factors. Considering the spatiotemporal characteristics of LBSN, Cheng et al. [3] proposed FPMC with candidates Region Constraint (FPMC-LR) method and provided new POIs for users by combining individual Markov chains and local and regional constraints. Rendle et al. [10] used Factorized Personalized Markov Chains (FPMC) to predict the next check-in interest point by expressing the short-term and long-term preferences of users. Xiong et al. [8] proposed a Bayesian probability tensor decomposition model based on time context, which dynamically acquired the potential features of users, POIs and months and could learn the global evolution of potential features. However, it is too sparse to model the time factor by month, which made the results not ideal. However, in practical recommendation applications, the recommendation results obtained by these traditional recommendation methods lack the user’s personalized requirements for POI. [24]. Liu et al. [14] used Skip Gram to train the temporal latent representation vector of POIs and proposed a time-aware POIs recommendation model. The spatio-temporal model TS-RNN proposed in this paper takes the spatio-temporal context elements into account in RNN mode to replace MF and FPMC, but the evaluation standard is still BPR.

With the continuous application of the Markov chain and factorization in the next POIs recommendation, both of them show their own limitations. The Markov chain is that it assumes strong independence among different factors, and the state of each POI in the first-order Markov chain is only related to the previous POI, which limits its performance. The limitation of tensor decomposition is that it is faced with a difficult problem which is called cold-start.

Some research work [5,9,12,13,17–19] shows that combining sequence, geography and time factors can obtain better recommendation results. Liu et al. [12] extended RNN and proposed a spatiotemporal recurrent neural network method. In this method, the time conversion matrix can be created with different time intervals to simulate the time context, and the distance conversion matrix can be created with different geographical distances to simulate the spatial context. Inspired by the Word2Vec framework, Zhao et al. [13] proposed the Geo-Teaser model, which embedded the time factor into the model to capture the time characteristics, and constructed the pairwise preference ranking at the geographical level. Then, POIs are ranked according to the preference score function, and the top-N POIs with the highest scores are recommended for users. In order to predict the access preference for the next POIs, Li et al. [17] introduced the time and multi-level context attention mechanism, which can dynamically select relevant check-in locations and distinguish context factors. The geographic-time awareness hierarchical attention
network, which is developed by Liu et al. [18], can reveal the dependencies of the overall sequence and the relationship between POIs through the BiLSTM network while using the geographic factor. Huang et al. [19] proposed a context-based self-attention network for the next POIs recommendation, which used positional encoding matrices instead of time encodings to model dynamic contextual dependencies. Guo et al. [25] proposed DeepFM, which combined factorization machine and feature embedding and sharing strategy to recommend. Among them, feature embedding and sharing strategies can avoid the establishment of feature engineering. However, the invalid second-order combination features may bring noise and adversely affect the model performance.

However, each of the above models [13,17–19,25] does not deeply mine the distance and time relationship between POIs in the trajectory when obtaining the correlation between POIs and does not add user social information into the model or framework. Research work [26,27] shows that although the influence of social relations is far less than geographical factors and time factors, it can affect the user’s check-in location selection that introduces the social factors into the next POIs recommendation.

The proposed CGTS-HAN model uses geographic factors to capture the features of POIs and their correlations in order to improve the recommendation performance of the next POIs recommendation.

3. Preliminaries
3.1. Problem Definition
This section presents five definitions [14] and specific problem statements related to the proposed next POIs recommendation problem.

Definition 1. User set. The user sets represent a set of users, denoted by \( U = \{ u_1, u_2, \ldots, u_{|U|} \} \).

Definition 2. POIs sets. The POIs sets represent a set of points of interest, denoted by \( P = \{ p_1, p_2, \ldots, p_{|P|} \} \). Each object in set of POIs consists of a 2-tuple, denoted by \((lon_i, lat_i, t_i)\), where \(lon_i\) and \(lat_i\) represent the longitude and latitude of the point of interest, respectively, and \(t_i\) represents the check-in time.

Definition 3. Time state set. The time state set denoted by \( T = \{ t_1, t_2, \ldots, t_{|T|} \} \), which is used to indicate the time points of the user’s check-in sorted by time in a day.

Definition 4. Check-in records. The check-in records denoted by \( Tr_i \). A check-in record represents a record of a user’s visit to a point of interest in one day.

Definition 5. Check-in history. The check-in history denoted by \( Tr = \{ Tr_1, Tr_2, \ldots, Tr_i, \ldots \} \). A user’s check-in history contains all his check-in records. The check-in history represents a series of check-in POI records of the user arranged in chronological order.

Problem statement: Given a user, denoted by \( u_i \), and his check-in history, denoted by \( Tr \), according to the user’s check-in history and the current point of interest, recommend the \( p_n \) for the user to visit at the next moment from \( P \).

3.2. Symbols Definition
To illustrate, Table 1 summarizes the main symbols used in this paper as follows.
Table 1. Main Notation.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_i$, $p_i$</td>
<td>Preference vectors of $u_i$, $p_i$</td>
</tr>
<tr>
<td>$T_r$</td>
<td>Sets of trajectory sequences for users</td>
</tr>
<tr>
<td>$T_{r_i}$</td>
<td>A set of check-ins for user $u_i$</td>
</tr>
<tr>
<td>$c_i$</td>
<td>Context of user $u_i$</td>
</tr>
<tr>
<td>$t_i$</td>
<td>$t_i$ latent semantic vector</td>
</tr>
<tr>
<td>$U_i, S_o, T_i$</td>
<td>Matrices of user preference, social relationship and $t_i$ latent semantic vector</td>
</tr>
<tr>
<td>$p_{r_i, S_i, p_i}$</td>
<td>The geographical predecessor vector, the geographical successor vector, the preference vector of POI</td>
</tr>
<tr>
<td>$P_r, S_r, P_i$</td>
<td>The geographical predecessor matrix, the geographical successor matrix, the preference matrix of POI</td>
</tr>
<tr>
<td>$u_i^{dp}$</td>
<td>Dynamic preferences of $u_i$</td>
</tr>
<tr>
<td>$W_1, W_2, W_3, ..., W_n$</td>
<td>Weight matrices in the model</td>
</tr>
<tr>
<td>$e_{u_i, c_i}$</td>
<td>Feature vector of the interaction between user $u_i$ and context $c_i$</td>
</tr>
<tr>
<td>$p_u$</td>
<td>Original feature vector of $u_i$</td>
</tr>
<tr>
<td>$g_{c_i}$</td>
<td>Context feature vector of $c_i$</td>
</tr>
<tr>
<td>$b_1, b_2, b_3, ..., b_k$</td>
<td>Bias terms in the model</td>
</tr>
</tbody>
</table>

4. CGTS-HAN Model

The framework of the CGTS-HAN model proposed in this paper is shown in Figure 1.

The model mainly consists of a context interaction layer, a geo-temporal social attention network and a co-attention network. The context interaction layer models the interaction between each user and their context information in the context environment and obtains the influence of each context on the user. Embedding layers are used to address the heterogeneity that exists among recommendation factors. Afterward, the model introduces a geo-temporal-social attention network to model the geographic relationships, temporal dependencies, and users’ social relationships among POIs of check-in sequences. The co-attention network is used to capture the dynamic preferences of users. Finally, we use a negative sampling algorithm to train the model. The next POI recommendation usually feeds back a sorted list of points of interest to the user, so this model first calculates the probability of the target user visiting the points of interest, then calculates the scores of candidate points of interest according to the Bayesian Equation, and finally sorts POIs to obtain an ordered list of top N POIs.

4.1. The Context Interaction Layer

In this paper, a feedforward neural network is introduced to simultaneously learn high-order features and low-order features to capture the interaction between the user and
the context and obtain the influence of the context on the user. The eigenvectors are as follows.
\[ e_{u,c_i} = f_1(p_{u,i}, g_{c_i}) \tag{1} \]

In the above formula, the \( f_1(\cdot) \) is used to represent the feature interaction function; its input is the user and the context, which are represented by \( u_i \) and \( c_i \), respectively. The \( e_{u,c_i} \) in the Equation represents the feature vector of the interaction between the user and context.

The input layer is responsible for receiving input and distributing it to the hidden layers (so called because they are invisible to the user). These hidden layers are responsible for the required calculations and output to the output layer, and the user can see the final output of the output layer. The modeling process of the context interaction layer feedforward neural network is shown in Figure 2.

\[ y = RELU(W_{uz}p_{u_i} + W_{cz}g_{c_i} + b_2) \tag{2} \]

In the above formula, \( W_{uz} \) and \( W_{cz} \) are the weight matrices of the nonlinear connection layer, \( b_2 \) is the bias term; \( RELU(\cdot) \) is the nonlinear activation function Linear Unit. After the input layer is multiplied by the weight, the result is often further processed; that is, the result is used as the input of the first layer of the hidden layer.

In order to enhance the interaction between the user and the context, the model builds three hidden layers on top of the nonlinear connection layer, which are specifically expressed as follows:
\[ y_1 = RELU_1(W_1y + b_1) \tag{3a} \]
\[ y_2 = RELU_2(W_2y_1 + b_2) \tag{3b} \]
\[ y_3 = RELU_3(W_3y_2 + b_3) \tag{3c} \]

In the above formula, \( W_1, b_1 \), and \( RELU_1(\cdot) \) represent the weight matrix, bias term and RELU activation function of the first hidden layer, respectively. The pronouns in the second and third hidden layer Equations have meanings and so on.

\( y_1, y_2, y_3 \) represent the outgoing vectors of the first, second and third hidden layers, respectively.
The outgoing vector \((y_3)\) of the third hidden layer in the model is passed to the output unit, and the output unit converts it into the feature vector of the context that acts on the user, which is expressed as follows:

\[
e_{u,c_i} = (W^u_e y_3 + b^u_e)
\]  

(4)

In the above formula, \(W^u_e\) and \(b^u_e\) represent the weight matrix and bias term of the output layer, respectively.

4.2. The Embedding Layer

The function of the embedding layer is to project the POIs into the latent semantic space, and use the matrix form to transform the geographic, temporal and social relationship factors of the points of interest in the context interaction layer, so as to relieve the heterogeneity that exists among the geographical, temporal and social factors.

Inspired by Transition [2], this paper firstly establishes a geographical predecessor vector, a geographical successor vector and a preference vector of points of interest for each POI, denoted as \(pr \in \mathbb{R}^{1 \times d}\), \(su \in \mathbb{R}^{1 \times d}\), \(pi \in \mathbb{R}^{1 \times d}\), respectively, where \(d\) is the potential dimension. The predecessor vector is used to receive the check-in trend of other POIs, and the successor vector is used to reflect the check-in trend of this POI transferring to other POIs. After that, the embedding layer transforms the established POI vector into three matrices \(Pr \in \mathbb{R}^{\lvert P \rvert \times d}\), \(Su \in \mathbb{R}^{\lvert P \rvert \times d}\), \(Pi \in \mathbb{R}^{\lvert P \rvert \times d}\), respectively. Similarly, we sequentially create a latent semantic matrix \(Ti \in \mathbb{R}^{\lvert T \rvert \times d}\) for temporal states, a social relationship matrix \(So \in \mathbb{R}^{\lvert U \rvert \times d}\) and a user preference matrix \(Ui \in \mathbb{R}^{\lvert U \rvert \times d}\) for users.

4.3. Geographical–Temporal Social Attention Networks

4.3.1. Modeling for Geographic Factors

The geographical attention network adopts the Transformer [4] mechanism. The input vector is regarded as a key-value pair by the encoder. The output of the encoder is compressed into the query by the decoder. Finally, the output query is mapped to the set of keys and values. The transformer network structure is simple, based on a self-attention mechanism, and computation is executed in parallel, which makes the Transformer efficient and requires less training time. The attention function in Transformer is described as mapping a query and a set of key value pairs to the output. Queries, keys and values are vectors. The attention weight is calculated by computing the dot product attention for each word in the sentence. The final score is a weighted sum of these values. The transformer mechanism is divided into the following four steps.

Step 1: Take the dot product of the keys for each input vector in the query. The input data consists of a geographic predecessor matrix \((Pr(query))\), a geographic successor matrix \((Su(key))\) and a POI preference matrix \((Pi(value))\).

Step 2: Scale the dot product by dividing by the square root of the dimension of the key vector.

Step 3: Use softmax to normalize the scale values. After softmax is applied, all values are positive and add up to 1.

Step 4: Apply the dot product between the normalized fraction and the value vector \(QK^T\), and then calculate the sum.

A nonlinear transformation of shared parameters is used to map \(Pr\) and \(Su\) into the same semantic space and then compute their weight matrix as follows.

\[
AG(Pr, Su) = \text{softmax}\left(\frac{QK^T}{\sqrt{n}}\right)Pi
\]  

(5)

\[
Q = \text{RELU}(PrW_Q + B_q)
\]  

(6)

\[
K = \text{RELU}(SuW_K + B_k)
\]  

(7)
Among them, Q, K are vectors of size $1 \times d$, $W_Q \in \mathbb{R}^{d \times d}$, $W_K \in \mathbb{R}^{d \times d}$, $B_q \in \mathbb{R}^{M \times d}$ and $B_k \in \mathbb{R}^{M \times d}$ are model parameters, $\sqrt{n}$ is used to scale the dot product value of $QK^T$, and the matrix output by Equation (5) represents the geographic relationship between $M$ POIs.

Since the dot product cannot model the geographic distance between POIs, this paper uses a Power Law Function (PLF) [28] based on the geographic relationship to examine the effect of geographic distance between POIs. The PLF is defined as follows.

$$P(x) = ax^{-\gamma}$$

In the above formula, $x$ and $P(x)$ are positive random variables, and $a$ and $\gamma$ are constants greater than zero. In the model, the probability value of a user visiting a point of interest follows a PLF.

The model introduces the influence of geographical factors between adjacent POIs into the attention network, then embed Equation (8) into Equation (5), and the rewritten Equation (5) is as follows.

$$\tilde{AG}(Pr, Su) = \text{softmax}\left(\frac{QK^T}{\sqrt{n}}\right)P_i \times P(x)$$

To output the geographic impact to the next stage (i.e., the temporal impact), the matrix output by the geographical attention network from Equation (9) is defined as follows.

$$\tilde{GAG}(Pr, Su, P(x)) = \tilde{AG}(Pr, Su) \times P(x)$$

4.3.2. Modeling for Time Factors

This paper captures the user’s long-term and recent interest features according to the bidirectional LSTM [29] based on the Attention mechanism. The output of the bidirectional LSTM $\tilde{GAG}$ is used as the input for the time factor modeling, and the output is a vector set $H = [h_1, h_2, ..., h_i, ..., h_T]$ with dimension $M \times d$, where $h_i$ represents the output at the time step $i$, $T$ is the sentence length $i = 1, ..., T$. This part of the network includes two sub-networks, sequence and context, which transfer time information forward and backward, respectively. Then pass the set of output vectors $H$ of the LSTM layer to the Attention layer to obtain a weight matrix for temporal factor modeling. Firstly, the model uses the tanh activation function to make the time factor available to the nonlinear model, as shown below.

$$Z = \tanh(H)$$

Then, the model process the values using the softmax function to transform $Z$ into probabilities, as shown below.

$$a_t = \text{softmax}(x^T Z)$$

Finally, the model obtains the weight matrix $r$ based on the set $H$ of output vectors and the probability $a_t$.

$$r = H a_t^T$$

In the above formula, $H \in \mathbb{R}^{M \times d}$, $x^T$ is the transpose of the parameter vector, $a_t^T$ is the transpose of the $a_t$. The representation $r$ of the sentence is formed by a weighted sum of these output vectors. Finally, The resulting classification sequence pair is denoted as follows.

$$h^* = \tanh(r)$$

4.3.3. Modeling for Social Factors

The model uses a multi-layer sub-network to obtain the attention score, and the results are as follows.

$$s_1(u_i, u_m) = \text{RELU}(...\text{RELU}(\text{RELU}(W_1^u \cdot p_{u_i} + W_2^u \cdot e_m + b_4) + b_5)...)$$

"
In the above Equation, \( W^u \) and \( W^w \) are weight coefficients, \( b_4 \) and \( b_5 \) are bias terms, and \( e_m \) is the embedded data of the user’s \( (u_i) \) neighbor user \( (u_m) \). After \( s_1(u_i, u_m) \) is calculated by Equation (15), it is normalized by the softmax function used in Equation (16), and finally, the social influence score is obtained as follows.

\[
\alpha_{i,m} = \text{softmax}(s_1(u_i, u_m)) = \frac{\exp(s_1(u_i, u_m))}{\sum_{n \in S_o} \exp(s_1(u_i, u_n))}
\]  

(16)

4.4. Co-Attention Network

This paper creates a co-attention network to capture users’ dynamic preferences. Specifically, the co-attention network uses a late fusion strategy to incorporate different weighted attention values \( (W^a) \) and nonlinear connection layers to learn the dynamic preferences of users. Given the context feature vector of \( u_i \), the sequence pair of time factors and the social influence score, and then obtain the overall effect of the context on the target user by weighting and summing them, which is represented by the dynamic preference \( (u_i^{dp}) \), which is calculated as follows.

\[
u_i^{dp} = \Psi(a_i^{dp} W^u + b_6) W^U
\]  

(17)

\[
a_i^{dp} = \text{concat}([(u_i, e_{i,m}), W^W, (t_i, h_s), (a_i, u_m), W^W])
\]  

(18)

In the above formula, \( W^u \in \mathbb{R}^{d \times d} \) and \( W^U \in \mathbb{R}^{d \times d} \) are model parameters, \( b_6 \) is bias terms, \( \Psi(\cdot) \) represents the nonlinear connection function and \( a_i^{dp} \) is the weighted joint score of the common attention network layers.

4.5. Learning and Optimization

After obtaining the user’s dynamic preference, the model uses the softmax function to generate the conditional probability distribution of the next POI \( \rho_n \), as follows.

\[
P(\rho_n | u_i, c_t, T) = \frac{\exp(u_i^{dp}, p_i^T)}{\sum_{v=1}^{[P]} \exp(u_i^{dp}, p_v^T)}
\]  

(19)

where \( P(\rho_n | u_i, c_t, T) \) is the probability distribution of user \( u_i \)'s access to POI \( \rho_n \), which is calculated according to the weighted average of user \( u_i \)'s dynamic preference \( u_i^{dp} \)'s attention weight. \( p_i^T \) is the transpose matrix of preference vectors of POI \( \rho_i \), and \( p_v^T \) is the transpose matrix of preference vectors of random POI \( \rho_v \).

Given a training dataset \( \mathcal{X} = \{ T_r, u_i, c_t, \rho_n \} \), its joint probability distribution is denoted as follows.

\[
P_\Theta(\mathcal{X}) = \prod_{x \in \mathcal{X}} P(\rho_n | u_i, c_t, T_r) = \prod_{x \in \mathcal{X}} \frac{\exp(u_i^{dp}, p_i^T)}{\sum_{v=1}^{[P]} \exp(u_i^{dp}, p_v^T)}
\]  

(20)

In the above formula, \( \mathcal{X} \) represents the training set, and the model parameter value is \( \Theta = \{ Pr, Su, Pi, So, Ti, W^Q, W^K, W^U, W^W \} \). By processing the regularization term of the above formula, the objective function is transformed into the following form.

\[
\mathcal{L} = \sum_{x \in \mathcal{X}} P(\rho_n | u_i, c_t, T_r) - \lambda \Theta
\]  

(21)

The computational cost of the above objective function will increase with the increase of POIs during optimization. Using the negative sampling method to optimize the objective function will reduce the training complexity, which can significantly improve computational
efficiency. Therefore, the model rewrites Equation (18) using the Negative Sampling technique, and the results are shown below.

$$L = \sum_{x \in X} \log \sigma(u^d_{pi} T) + \sum_{k=1, \bar{P} \sim q} K \log \sigma(-u^d_{pi} T) - \lambda \Theta$$

In the above formula, $\sigma = \frac{1}{1 + e^{-x}}$ is used to approximate the probability, and $K$ is the number of negatively sampled POIs.

4.6. Generation of Recommendation List

In the case of known $u_i$ and his check-in history $(u_i)$, the model can calculate the ranking score of each candidate POI according to the Bayesian and then recommends top-ranked POIs to the user. The score is calculated as follows.

$$\hat{r}_{pu} = P(p_n|Tr) \propto P(p_n)P(Tr|p_n) = P(p_n) \prod_{p_n \in Tr} P(p_n') | p_n$$

The ranking score of each POI in the final recommendation list of the model is formed according to the ranking score ($\hat{r}_{pu}$) of the candidate POIs and their probability distribution.

$$S_{pu} = \hat{r}_{pu} \times P(p_n | u_i, c_i, Tr)$$

The following Algorithm 1 summarizes the learning algorithm flow of CGTS-HAN, which is mainly composed of three modules: data (lines 2–9), factor modeling (lines 10–15) and prediction (lines 16–19).

---

**Algorithm 1:** The Procedure of CGTS-HAN.

Input: $\mathcal{U}$, $\mathcal{P}$, $\mathcal{T}$, $\mathcal{X}$, $\Theta$.
Output: Final POIs recommendation list.

1. Initialization: $t_i = 0$.

// Data Module
2. for each $u_i \in \mathcal{U}$ do
3. Split check-in record $T_r_i$ by $T_r$;
4. Model preference vector $u_i$;
5. Model original feature vector $p_{ui}$;
6. Model context feature vector $g_{ci}$;
7. for each $u_i \in T$ do
8. Split check-in time by day;
9. Model preference vector $t_i$;

// Factor Module
10. for each $u_i = 1; u_i \in \mathcal{U}; u_i$++ do
11. Model $a_{u_i, u_n}$;
12. for each $p_i = 1; p_i \in \mathcal{P}; p_i$++ do
13. Model GAG;
14. for each $t_i = 1; t_i \in \mathcal{T}; t_i$++ do
15. Model $h^*$

// Prediction Module
16. for each $u_i \in \mathcal{U}$ do
17. Calculate the candidate POIs scores according to Equation (23);
18. Calculate the POIs scores according to Equation (24);
19. Rank POIs and select top-N individual POIs.
5. Experiment
5.1. Processing of Datasets

In this paper, we use the published Foursquare dataset and Yelp dataset for experiments. Among them, the Foursquare dataset selects the check-in data of New York users from 1 May to 30 June 2014. The Yelp dataset selects the activity data of New York users from 1 August 1 to 30 October 2017. Moreover, we remove inactive users with less than 10 check-in locations and POIs with less than 10 check-ins from the datasets.

Table 2 shows the dataset statistics after preprocessing. In order to make the model proposed in this paper more suitable for the check-in scenario of POIs, we take 80% of the check-in trajectories of each user in the two datasets as the training sets and 20% as the test sets.

Table 2. Statistics of Datasets.

<table>
<thead>
<tr>
<th>Amount</th>
<th>Foursquare</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>users</td>
<td>48,763</td>
<td>22,754</td>
</tr>
<tr>
<td>POIs</td>
<td>18,158</td>
<td>37,879</td>
</tr>
<tr>
<td>check-in records</td>
<td>1,287,429</td>
<td>248,5027</td>
</tr>
<tr>
<td>social ties</td>
<td>118,421</td>
<td>78,647</td>
</tr>
</tbody>
</table>

According to the research results in related works, the two important factors in the recommendation of the next POIs are the distance and time between POIs. Figure 3a and Figure 3b, respectively, represent the Cumulative Distribution Function (CDF) of the distance between two adjacent POIs checked in by each user in one day on the Foursquare datasets and Yelp datasets. The role of CDF is to help us understand the imbalance of distance distribution and find out which check-in distance accounts for the largest proportion of the total.

![CDF of Distance](image Lucia.png)

Figure 3. CDF of Dis between next Check-ins POIs: (a) CDF on Foursquare; (b) CDF on Yelp.

In Figure 4, the horizontal axis represents the check-in distance between POIs, and the vertical axis represents the distance distribution ratio. It can be seen that about 85% of the consecutive check-in distances in the Foursquare and Yelp datasets are within the range of 8.2 km and 7.8 km. It is well known that weekdays and weekends in a week have different effects on check-in locations, so we divide the check-in times in the datasets into two categories: weekdays and weekends. Similarly, in order to observe the influence of the time of day on the check-in location, we divided the time of day into six periods: Early Morning (04:01–08:00), Morning (08:01–12:00), noon (Noon, 12:01–16:00), afternoon (Afternoon, 16:01–20:00), evening (Night, 20:01–24:00) and late night (Wee, 00:01–04:00). The results in Figure 4 show that the distance between consecutive check-in points on weekends is slightly larger than that during weekdays, which means that people are more inclined to go to places with farther distances between POIs on weekends.
represents the positive sample set, and \( \delta \) represents the negative sample set. Afternoon (16:01–20:00), evening (20:01–24:00) and late night (00:01–04:00) are divided into six periods: Early Morning (04:01–08:00), Morning (08:01–12:00), noon (12:01–16:00), afternoon (16:01–20:00), evening (20:01–24:00) and late night (00:01–04:00).

As shown in Figure 3, the horizontal axis represents the check-in distance between POIs, and the vertical axis represents the distance distribution ratio. It can be seen that about 85% of the check-in distance is within 8.2 km and 7.8 km. It is well known that weekdays and weekends in a week have different check-in patterns. The consecutive check-in distances in the Foursquare and Yelp datasets are within the range of 8.2 km and 7.8 km. It is well known that weekdays and weekends in a week have different check-in patterns.

To observe the influence of the time of day on the check-in location, we divided the time of day into six periods: Early Morning (04:01–08:00), Morning (08:01–12:00), noon (12:01–16:00), afternoon (16:01–20:00), evening (20:01–24:00) and late night (00:01–04:00). We observed that people are more inclined to go to places with farther distances between POIs on weekends. This means that people are more inclined to go to places with farther distances between POIs on weekends. The opposite is true if the instructed function \( \delta(p_{uij} > p_{uij'}) \) returns false. The higher the AUC, the stronger the ranking ability of the model.

In the above formula, \( J \) represents the positive sample set, and \( J' \) represents the negative sample set. When the indicator function \( \delta(p_{uij} > p_{uij'}) \) returns true, it means that the predicted probability of \( u_i \) acting on positive samples is greater than the predicted probability of negative samples. The opposite is true if the instructed function \( \delta(p_{uij} > p_{uij'}) \) returns false. The higher the AUC, the stronger the ranking ability of the model.

5.2. Experimental Evaluation

To evaluate the performance of the recommendation algorithm, we use the Area Under the ROC curve (AUC), Precision and Recall as the evaluation indicators. AUC is a common indicator for evaluating the quality of ranking lists in machine learning, and it is calculated as follows.

\[
AUC = \frac{1}{|U|} \sum_{u_i \in U} \frac{1}{|J|} \sum_{j \in |J|} \sum_{j' \in |J|} \delta(p_{uij} > p_{uij'})
\tag{25}
\]

In the above formula, \( J \) represents the positive sample set, and \( J' \) represents the negative sample set, \( u_i \) represents the \( i \)th user in the set \( U \). When the indicator function \( \delta(p_{uij} > p_{uij'}) \) returns true, it means that the predicted probability of \( u_i \) acting on positive samples is greater than the predicted probability of negative samples. The opposite is true if the instructed function \( \delta(p_{uij} > p_{uij'}) \) returns false. The higher the AUC, the stronger the ranking ability of the model.

Precision and Recall are used to compare the prediction results of all algorithms’ bias sizes.

The calculation method of Precision@N is as follows.

\[
\text{Precision@N} = \frac{1}{|U|} \sum_{u_i \in U} \frac{|P_{ui}^T \cap P_{ui}^R|}{N}
\tag{26}
\]

The calculation method of Recall@N is as follows:

\[
\text{Recall@N} = \frac{1}{|U|} \sum_{u_i \in U} \frac{|P_{ui}^T \cap P_{ui}^R|}{P_{ui}^R}
\tag{27}
\]

In the above formula, \( u_i \) represents the \( i \)th user in the set \( U \), \( P_{ui}^R \) represents the set of POIs recommended to \( u_i \) in the training set, and \( P_{ui}^T \) represents the set of POIs that \( u_i \) has checked in in the test set. \( N \) represents the number of test instances. Moreover, the higher the Precision@N and Recall@N, the more accurate and comprehensive the recommendation results are.
5.3. Compared Models

This paper compares the CGTS-HAN model proposed in this paper with the following four state-of-the-art models.

1. Geo-Teaser [13]: Geo-Teaser introduces geographic factors based on the temporal POIs embedding model, which can capture the context of check-in sequences and various temporal features of different dates and establish a geographic-level preference ranking model.

2. HST-LSTM [15]: This model combines the time factor with LSTM and adopts a hierarchical architecture to predict the next location by utilizing user historical check-in information.

3. CSAN [19]: CSAN is a multi-modal content unified framework based on an attention mechanism, which projects users’ heterogeneous behaviors into a common latent semantic space and then inputs the output results into a feature self-attention network to capture the polysemy of user behaviors.

4. DeepFM [25]: DeepFM is a new neural network structure that combines the recommendation ability of factorization machines and the feature learning ability of deep learning.

Among these methods, CSAN and DeepFM do not introduce temporal effects, while Geo-Teaser and HST-LSTM do not introduce neural networks to learn user behavior features. The parameters of each method are set as follows:

1. The weight of time and geographical influence in (2) is set to 0.5; (2), (3) and (4) the initial learning rate of the deep model is set to 0.01. In (2), the context window length and vector size of the hierarchical softmax algorithm are set to 8 and 200, respectively.

The parameter Settings involved in the CGTS-HAN model proposed in this paper are the same as the above comparison methods, and CGTS-HAN uses three layers in the context iteration layer.

5.4. Experimental Results and Analysis

5.4.1. Analysis of the Effect

Table 3 shows the AUC experimental results of different models on the two datasets. It can be seen that the AUC of the CGTS-HAN model is consistently higher than all baseline models. The AUC of CGTS-HAN on both datasets is at least 21.4% and 14.5% higher than other models.

Table 3. Experimental Results of AUC.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Foursquare</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geo-Teaser</td>
<td>0.422</td>
<td>0.473</td>
</tr>
<tr>
<td>HST-LSTM</td>
<td>0.498</td>
<td>0.520</td>
</tr>
<tr>
<td>CSAN</td>
<td>0.0584</td>
<td>0.0587</td>
</tr>
<tr>
<td>DeepFM</td>
<td>0.621</td>
<td>0.704</td>
</tr>
<tr>
<td>CGTS-HAN</td>
<td>0.835</td>
<td>0.849</td>
</tr>
</tbody>
</table>

5.4.2. Analysis of Effectiveness

This paper conduct experiments with different values of the recommendation list size and extracts the experimental results of $N = \{5, 8, 15, 30\}$ as samples. The experimental results of precision and recall are shown in Figures 5 and 6.
To balance the effects of precision and recall, we select the top 15 POIs for recommendation in the experiment. In this case, the precision of the proposed CGTS-HAN model on the Foursquare dataset drops at first and then gradually decreases. Moreover, it can be seen from Figure 6 that the Recall rate of all models increases with the increase of $N$. Models except the HST-LSTM model increase first and then gradually decrease. Additionally, it can be seen from Figure 6 that the Recall rate of all models increases with the increase of $N$.

Experiments demonstrate that adding auxiliary information of social connections helps improve POIs recommendation performance and, at the same time, confirms the effectiveness of modeling with attention networks.

### 5.4.3. Parameter Analysis

Using the control variable method, this part of the parameter analysis uses the pre-processed data of the Foursquare dataset as test samples and the pre-processed data of the Yelp dataset as train samples. The parameter settings involved in the CGTS-HAN model proposed in this paper are set as follows:

- The initial learning rate of the deep model is set to 0.01. In (2), the context window length $d$ is set to 4, and the geographical influence weight coefficient $\alpha$ and the learning rate $\beta$ to $[5,8,15,30]$ as samples. The experimental results of precision and recall are shown in Figures 5 and 6.

### Table 3.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Foursquare</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGTS-HAN</td>
<td>0.835</td>
<td>0.849</td>
</tr>
<tr>
<td>Geo-Teaser</td>
<td>0.422</td>
<td>0.473</td>
</tr>
<tr>
<td>HST-LSTM</td>
<td>0.498</td>
<td>0.520</td>
</tr>
<tr>
<td>CSAN</td>
<td>0.0584</td>
<td>0.0587</td>
</tr>
<tr>
<td>DeepFM</td>
<td>0.729</td>
<td>0.739</td>
</tr>
</tbody>
</table>

**Figure 5.** Comparison Results of Precision with $N$: (a) Precision of train on Foursquare; (b) Precision of test on Foursquare; (c) Precision of train on Yelp; (d) Precision of test on Yelp.

**Figure 6.** Comparison Results of Recall with $N$: (a) Recall of train on Foursquare; (b) Recall of test on Foursquare; (c) Recall of train on Yelp; (d) Recall of test on Yelp.

It can be seen from Figure 5 that, with the increase of $N$, the Precision of the rest of the models except the HST-LSTM model increases first and then gradually decreases. Moreover, it can be seen from Figure 6 that the Recall rate of all models increases with the increase of $N$. To balance the effects of precision and recall, we select the top 15 POIs for recommendation in the experiment. In this case, the precision of the proposed CGTS-HAN model on the Foursquare and Yelp datasets is 18% and 6% higher than that of DeepFM, and the Recall is 10% and 6% higher, respectively. This verifies the good effect of CGTS-HAN integrating sequence, geographic, time and social influencing factors. Experiments demonstrate that adding auxiliary information of social connections helps improve POIs recommendation performance.
performance and, at the same time, confirms the effectiveness of modeling with attention networks.

5.4.3. Parameter Analysis

In this experiment, the parameter sensitivities of time influence weight coefficient $\alpha$, geographical influence weight coefficient $\beta$ and learning rate $\eta$ were quantitatively analyzed by the control variable method. This part of the parameter analysis uses the preprocessed data set and does not divide the training set and the test set.

From Figure 7, we can see how the parameters affect the precision and recall of CGTS-HAN. In order not to tune $\alpha$, this paper integrates $\eta$ into $\alpha$ and $\beta$ in the tuning process (that is, $\alpha \times \eta \rightarrow \alpha$, $\beta \times \eta \rightarrow \beta$) and then tunes the two parameters of $\alpha$ and $\beta$ to weigh the influence of time and geographical factors. In order to ensure convergence, we make the values of $\alpha$ and $\beta$ as small as possible in the experiment. At the beginning of the tuning, we assume that the values of $\alpha$ and $\eta$ are equal and change the learning rate by tuning $\alpha$. The experimental results show that when $\alpha$ increases, the precision and recall rate of the experimental model maintain an upward trend as a whole, and the growth rate slows down as the coefficient value increases. When $\alpha$ is equal to 0.05, the precision and recall performance of the model in this paper is balanced and reaches the best. Therefore, we assume that $\alpha$ is equal to 0.05 and remains unchanged by tuning the $\beta$ value to observe the changes of $\frac{\beta}{\alpha}$ in the precision and recall rate of the model in the $[0, 2]$ interval. Figure 7 shows that the CGTS-HAN can achieve the best recommendation performance when $\frac{\beta}{\alpha} \in [0.5, 1]$.

![Figure 7](image_url)

**Figure 7.** Parameter effect on $\alpha$ and $\beta$: (a) Precision on CGTS-HAN; (b) Recall of on CGTS-HAN.

5.4.4. Time Complexity Analysis

Complexity. For one check-in trajectory, learning the temporal embedding model costs $O(T \cdot K \cdot d)$, where $T$, $K$ and $d$ denote the context window size, the number of negative samples and the latent vector dimension, respectively. For the dynamic preference ($u_{dp}^i$) learning in Algorithm 1, we sample $m$ unvisited POIs, which can generate maximum $O(m^2)$. For each trajectory, the learning procedures cost $O(m^2 \cdot d)$. Therefore, the complexity of CGTS-HAN is $O(m^2 \cdot d \cdot T \cdot r)$, where $T \cdot r$ denote the user trajectory length.

Scalability. Generally, POIs are sparse in LBSN, and the number of check-in POIs is greater than the number of unchecked POIs. Furthermore, in order to make the model more efficient, we use Negative Sampling for optimization. The calculation time of each iteration of Equation (21) is approximately $O([(T \cdot r \cdot K) \cdot |X'|])$, where $K$ is the number of negative samples, and $|X'|$ is the training set size. In fact, the values of $T \cdot r$ and $K$ both satisfy a relation $(T \cdot r, K < |X'|)$. Therefore, as the check-in trajectories distribution of POIs in LBSNs follows PLF, the time complexity of CGTS-HAN proposed in this paper is linearly related to the number of check-in training sets $|X'|$, which also guarantees that CGTS-HAN to parallel the parameter updates and scalable on large datasets.
6. Conclusions and Future Directions

This work studies the use of contextual information and interest preferences of users’ location-based social networks to recommend the next POI for users in the IoT environment. This paper proposed a new next POI recommendation model named CGTS-HAN, which can learn the contextual features of users’ POI more accurately than other models. The model aims to recommend the next POI according to the user’s historical trajectory, integrates the user’s current geographical location and time and considers its social influence to model the user’s dynamic preferences. In particular, the model can infer users’ interest preferences based on their social information to provide high-quality recommendations in the IoT environment when users have little or no activity history. Experiments are conducted on two datasets based on geo-social information, and the results demonstrate the effectiveness and efficiency of the method.

At present, the CGTS-HAN model uses offline training data, and future work will consider developing effective online training data to improve the accuracy of recommendation results. At the same time, sophisticated distributed representation methods can be developed to improve the next POI recommendation task.

Author Contributions: Conceptualization, X.L. and J.G.; methodology, X.L.; software, X.L. and J.G.; validation, X.L., J.G. and P.Q.; formal analysis, X.L.; investigation, J.G. and P.Q.; resources, P.Q.; data curation, X.L.; writing—original draft preparation, X.L. and J.G.; writing—review and editing, X.L., J.G. and P.Q.; visualization, J.G. and P.Q.; supervision, P.Q.; project administration, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References