TADSAM:A Time-Aware Dynamic Self-Attention Model for Next Point-of-Interest Recommendation

Peng Liu[1] **Yange Guo**[1] **Xianxian Li** *[1] **Yutian Zheng**[1] **Yuan Liang***†[2]

[1] Guangxi Key Lab of Multi-source Information Mining and Security,Guangxi Normal University
[2] School of Computer Science and Engineering Beihang University
liupeng@gxnu.edu.cn,gyg2019010376@stu.gxnu.edu.cn
lixx@gxnu.edu.cn,zytiansoft@126.com,liangyuan120@buaa.edu.cn

Abstract

In recent years, the next Point-of-Interest (POI) recommendation has been widely concerned with the prosperity of location-based social networks. Its purpose is to predict users' next activities according to their current time and place. However, it mainly faces two challenges: 1) with the change of time, the historical behaviors of users show diversity and complexity, and 2) due to the sparsity of user trajectory data, it is hard to capture users' preferences in the corresponding time pattern. Therefore, we propose a Time-Aware Dynamic Self-Attention Model TADSAM to predict the next decision-making activities of users in the future. Firstly, we use an extended self-attention mechanism to deal with the complex check-in records of the user. Secondly, considering the influence of time, we divide the user check-in records into different time windows and develop a personalized weight calculation method to exploit temporal patterns of the user's behaviors. Experimental results demonstrate that our method outperforms the novel models for the next POI recommendation on sparse check-in data.

Introduction

The rapid development of the mobile internet makes location-based social networks (LBSN) such as Gowalla and Foursquare more and more popular in our daily lives. The large amount of user data generated by LBSN makes it possible to recommend the next POI to users by understanding users' behavior.

The traditional POI recommendation recommends the appropriate POIs to users according to the general static preferences of users while ignoring recent visits. On the contrary, the next POI recommendation determines the next decision-making activity of the user according to the current location and previous information of visits. At the same time, many studies have achieved good results in the next POI recommendation (Chang et al. 2018; Yu et al. 2020; Luo, Liu, and Liu 2021). However, these methods only predict where the user goes next but do not when the user makes the next decision. We think it is crucial to predict when the user goes in the next POI recommendation. For example, the user's next

Figure 1: An example of the behavior distribution of three users in a day

behavior may not occur until a week or a month later, so our next POI recommendation is almost meaningless.

Therefore, the time factor of user visits plays a vital role. In the long run, the user's trajectory behavior is periodic. During working days, users will consider POI activities near the company but think POI or other entertainment places near home during the weekends. Analogously, the interest distribution of users within a day also changes dynamically. We give the sequence trajectories of three users in a day, as shown in Figure 1. From the historical tracks of the three users, we find that they have roughly been to the same place or participated in similar activities. Suppose you want to recommend the next POI to User 3. If only based on the previous historical trajectory and users with similar activities, the previous model will easily recommend the gym to users. However, User 3 may not want to go to the gym at 8:00 p.m., which is not a satisfactory recommendation result for User 3. It can be seen that time plays a crucial role in POI recommendation in both the long term and short term.

Previous work has conducted extensive researches on POI recommendation with time contexts, which are classified into two types. One type is developed based on the matrix factorization method. The entire set of possible timestamps is mapped to a limited time slot (Zhao et al. 2017), assuming the user's signature timestamp has the same impact in

*These authors contributed equally.
†Corresponding author

the same period. Then construct a low-dimensional space to capture the characteristics of time slots and combine them with other preference types. However, it is difficult to determine which time granularity will be used for time mapping slots. Therefore, it will also affect the model to make accurate recommendations. The other type mainly focuses on sequential transitions, such as the Markov model and the RNN model. In the early POI recommendation, the research found that the user's next behavior is easily affected by the user's current behavior. Many studies have begun to use the Markov chain model to pay attention to the sequential relationship of users' historical trajectory (Chen et al. 2018; Rendle et al. 2010). However, when dealing with user sequential tracks, the Markov chain model can only predict the user's next behavior through a single behavior of the user's historical trajectory, which often reduces the accuracy of recommendations. Subsequently, the recurrent neural network model with memory mechanism has become a research hotshot model in POI recommendation. There are various time-aware models for the next POI recommendation (Zhao et al. 2016; Li, Shen, and Zhu 2018). The RNN model has achieved good results in the sequential recommendation. However, the next POI recommendation's sequential dependence is powerful, and the RNN model is unstable in dealing with long sequential dynamics, so there is information loss during the calculation. Moreover, the weakness of parallel computing ability result in easy establishing false dependencies.

In recent years, a new and original attention mechanism has emerged (Vaswani et al. 2017). It has been employed extensively in machine learning and NLP, which has achieved good results. The novelty of it is that it captures the dependence of long sequences. Intuitively, it has powerful parallel computing ability. Therefore, inspired by the mechanism, we propose a time-aware dynamic self-attention network TADSAM to solve the above limitations in the next POI recommendation. TADSAM uses a multi-head attention mechanism to translate between check-in sequences. To consider the influence of time context, we use the relative time matrix of user trajectory to expand the vanilla self-attention mechanism. Moreover, to better explore the behavior pattern of the user's historical trajectory sequence, we divide the user's historical check-in records into different time windows and use the personalized weight calculation method to obtain the user's current preference. The specific contributions are as follows:

- We propose a time-aware-dynamic self-attention network method. We use a relative temporal matrix to expand the vanilla self-attention mechanism to explore the relationship between user check-in records. Through this method, we can not only explore the complex and diverse behavior patterns of users but also understand the changes in users' interests.
- We divide the user's historical check-in sequence into different time windows and design a personalized weight calculation method to explore the time pattern of user behavior more effectively.
- We combine the popularity and geographical impact of POI to improve the system's performance to overcome the problem of data sparsity and facilitate the retrieval of the system.
- We conducted complex comparative experiments on two public data sets to evaluate the performance of the method. The results show that our proposed method is superior to the advanced method recommended by the next POI for sparse data.

Related work

In early studies, the POI recommendation only recommended the first few popular POIs to users (Ye, Zhu, and Cheng 2013), which provides a reference for users to select the next location. But this recommendation method is too simple to meet the needs of users. At present, researchers propose many models for POI recommendation, such as the classical collaborative filtering model (Xiong et al. 2010; Ye et al. 2011) and the matrix factorization model (Liu and Aberer 2013; Li et al. 2021). (Ye et al. 2011) combine the geographical and social influence of users based on collaborative filtering. (Liu and Aberer 2013) propose the method based on the matrix factorization, which considers context and social information. (Li et al. 2021) calculate the context-aware similarity between different users based on matrix factorization technology to capture the impact of temporal and spatial characteristics on users.

Although these methods can effectively alleviate the problem of data sparsity, it does not mine the relationship deeply between user behaviors. Research shows the user's historical trajectory is a sequential relationship, which is user's next behavior decision will be affected by the previous behavior. The sequential model method was prosed based on the Markov chain to predict the possibility of the user's next behavior (Chen et al. 2018; Rendle, Freudenthaler, and Schmidt-Thieme 2010) in the early stage of the next point-of-interest recommendation. But the Markov chain model mainly focuses on the transition probability between two visits and does not understand the relationship among the sign-in records of the user's all trajectory. Challenged by the defects of Markov models, the recurrent neural network model with memory mechanism has received widespread attention. The spatiotemporal method (Zhao et al. 2016) defines the relationship among users, time, and local for the next POI recommendation. HST-LSTM (Kong and Wu 2018) integrates spatiotemporal influence into the LSTM model, which alleviates the problem of data sparsity. (Sun et al. 2020) module two LSTM models to obtain users' long-term and short-term preferences, respectively. (Liu et al. 2021) propose a category-aware gated recurrent unit (GRU) model to mitigate the negative impact of sparse check-in data, capture long-range dependence between user check-ins.

In recent years, the application of the attention mechanism in machine translation and natural language processing has attracted widespread attention. (Kang and McAuley 2018) capture the relationships between the user's sequential behaviors by using a self-attention mechanism. (Zheng and Tao 2020) utilize two attention mechanisms to dynami-

cally obtain the user's long-term and short-term preferences, respectively. STAN (Luo, Liu, and Liu 2021) uses a bi-layer attention architecture that firstly aggregates spatiotemporal correlation within user trajectory and then recalls the target with consideration of personalized item frequency. However, these methods neither explore the time relationship between the user trajectory sign-in records nor the time pattern of the user's behaviors. Inspired by (Chen et al. 2019; Yu et al. 2020), we propose a time-aware dynamic model to address the problems.

Problem Formulation

We define the set of user by U, and the set of location by L. In LSBN, each POI is a unique geographic identifier associated with longitude and latitude. A check-in record of each user u_i is a 3-tuple $r_i = (u_i, l_i, t_i)$, which means that user u_i visits POI l_i at the time point t_i. We denote the check-in trajectory of each user by $\mathrm{Traj}\,(u_i) = \{r_1, r_2, \ldots, r_{m_i}\}$. We convert each user's trajectory sequence into a fixed sequence length, i.e. $\mathrm{Seq}\,(u_i) = \{r_1, r_2, \ldots, r_k\}$, where k denotes the maximum length that we consider. In addition, we take the time interval between two locations as the element of relative time matrix. Each element Δ^t_{ij} is represented as $\Delta^t_{ij} = |t_i - t_j|$:

$$\Delta^t = \begin{bmatrix} \Delta^t_{11} & \Delta^t_{12} & \cdots & \Delta^t_{1k} \\ \Delta^t_{21} & \Delta^t_{22} & \cdots & \Delta^t_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta^t_{k1} & \Delta^t_{k2} & \cdots & \Delta^t_{kk} \end{bmatrix} \tag{1}$$

Definition 1 (Next POI recommendation)

Given the user's check-in trajectory $\mathrm{Traj}\,(u_i)$ and prediction time t_{N+1}, the task of the next POI recommendation is to predict the possibility of the POI l_i that user will visit at the time t_{N+1}.

The Model Architecture

We first introduce the components of the dynamic model, where include the embedding layer, self-attention layer, time pattern layer, and prediction layer. The overall architecture of our model is shown in Figure 2. To learn the parameters, we use the binary cross-entropy loss as the objective function.

Embedding Layer

The embedded layer is composed of the user's historical trajectory matrix and the relative time matrix. The embedding layer mainly studies the latent representation of the trajectory matrix and time relative matrix of users. We create four embedding indicators, i.e. $e^u \in \mathrm{R}^d, e^l \in \mathrm{R}^d, e^t \in \mathrm{R}^d$ and $\mathrm{E}(\Delta^{\mathrm{t}}) \in \mathrm{R}^{k\times d}$, to represent user, POI, discrete-time and time relation matrix embeddings, respectively, where d represents the dimension of embedding space. The original timestamp of the user's check-in record is discrete, so we map time to 168 dimensions, corresponding to 168 hours a week, which helps understand the specific time when the user visits a place and reflects the periodicity

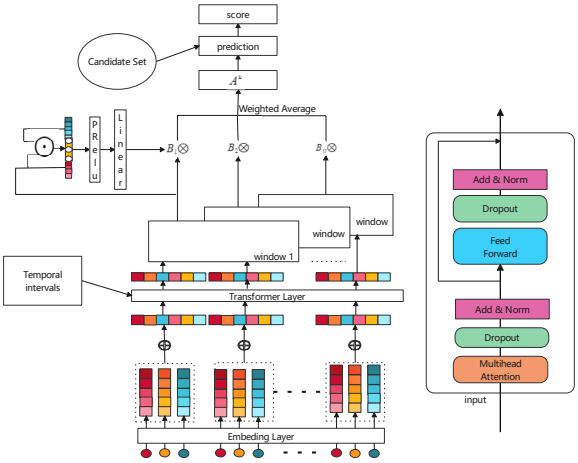


Figure 2: Time-aware dynamic self-attention model and internal structure of self-attention block

.

of the user's trajectory. The trajectory embedding of each user sequence $\mathrm{Seq}(u_i) = \{r_1, r_2, \ldots, r_k\}$ is represented as $\mathrm{E^u} = [e^{r_1}, e^{r_2}, \ldots, e^{r_k}] \in \mathrm{R}^{k\times d}$.

The time relative matrix expands the vanilla self-attention mechanism effectively reflects the relationship of user trajectory sequence. The relatively short time indicates a strong dependence between the two sign-in sites.

Extended Self-attention Layer

We propose a time-aware self-attention model to input a time dynamic trajectory sequence. Research shows that the successive check-ins of users are time-sensitive. Considering the same POI recommendation at different times, the subsequent check-in of users may be different. Inspired by the relative position self-attention mechanism (Shaw, Uszkoreit, and Vaswani 2018), we extend the self-attention to consider the time intervals between two access locations in a sequence. We take the relative time matrix as the relative position of self-attention, which can capture the relationship between the items of the user track sequence. The relative position of self-attention will be more efficient than the absolute position (Vaswani et al. 2017).

The self-attention layer includes two sub-layers: the self-attention layer with multiple heads and the feedforward network layer (FFN). Firstly, we use the multi-head attention mechanism to analyze users' history tracks with time information from different angles for mining users' various preferences. The specific implementation is as follows:

$$\mathrm{G^u} = \mathrm{MSA}(\mathrm{E^u}) = concat\left(head^1, head^2, \cdots, head^n\right)\mathrm{W_H} \tag{2}$$

where $\mathrm{W_H} \in \mathrm{R}^{nd_v\times d}$ is a transformation matrix for modeling the correlations of different heads, n is the number of heads, $d_v = d/n$, concat function connects several vectors and $\mathrm{G^u} = [g^u_1, g^u_2, \cdots, g^u_k] \in \mathrm{R}^{k\times d}$ is the output of MSA. $head^n$ is the output of n-th head. The number of the head is adaptively adjusted according to the size of the dataset. Then the improved self-attention mechanism is calculated

as follows:

$$head^n = \sum_{i=0}^{k} \alpha_{ij} \left(e^{r_i} \mathrm{W_V} + \Delta_{ij} \right) \tag{3}$$

where $\mathrm{W_V} \in \mathrm{R}^{d \times d}$ represents input matrix of value. We calculated each weighting coefficient by using a softmax function, as follows:

$$\alpha_{ij} = softmax \left(\frac{e^{r_i} \mathrm{W_Q} \left(e^{r_j} \mathrm{W_K} + \Delta_{ij} \right)^{\mathrm{T}}}{\sqrt{d_v}} \right) \tag{4}$$

Where $\mathrm{W_Q}, \mathrm{W_K} \in \mathrm{R}^{d \times d}$ represent input matrix of query and key respectively. $\sqrt{d_v}$ is to prevent the inner product from being too large.

Our time-aware attention layer linearly combines the users' historical trajectory information and the relative time information with self-adaptive weight. After the time-aware attention layer, we apply the feedforward network to endow nonlinearly the model, which improves the model ability of expression.

$$\mathrm{FFN(G^u)} = \mathrm{ReLU(G^u W_1 + b_1) W_2 + b_2} \tag{5}$$

As discussed in (Kang and McAuley 2018), after the superposition of the extended multi-head self-attention layer and feedforward network, there will be more problems, such as overfitting, unstable training process, and the need for more training time. Therefore, we add layer normalization, residual connection, and dropout technique to solve these problems after the multi-head attention mechanism and the feedforward network layer, which expressed as follows:

$$\mathrm{SA(E^u)} = \mathrm{LN(Dropout(MSA(E^u)) + E^u)} \tag{6}$$

$$\mathrm{A^u} = \mathrm{LN(Dropout(FFN(G^u)) + G^u)} \tag{7}$$

In the above two expressions, $\mathrm{SA\ (E^u)}$ represents the results of the multi-head self-attention layer processed by dropout technology, residual linking, and layer normalization. Similarly, $\mathrm{A^u}$ is the result of processing by FFN, which is also equivalent to the final output. The output is the user's complex preference matrix with temporal information and correlations from a check-in sequence. Algorithm 1 represents an extended self-attention network to extract users' diverse preferences.

Time Partten Layer

The user's long-term behaviors contain some core interests, which will not change with time and have a certain periodicity. In the self-attention layer, we discuss the time relationship between the two visit sites, which is far from enough. Research shows that affected by the clock, the user's preferences are different at different periods of the day. We further explore the time pattern of the user's preferences. It is unwise to divide the day evenly because the user's daily activities distribute unevenly. Referring to the method of (Yu et al. 2020), we determine the span of this period by using the probability density of the user's visits behavior during the day. The CatDM (Yu et al. 2020) adopts a personalized weight calculation method to accurately obtain the interests of each time window, which has achieved good results. Therefore, we use the same way to explore the time

Algorithm 1: Training trajectory sequence of users by the extended self attention

Input: Check in trajectory matrix of user, $\mathrm{E^u}$; Relative time matrix of user trajectory, $\mathrm{E}(\Delta)$
Parameter: $\mathrm{W_H}, \mathrm{W_Q}, \mathrm{W_K}, \mathrm{W_V}, \mathrm{W_1}, \mathrm{W_2}, \mathrm{b_1}, \mathrm{b_2}$
Output: Diverse preference representation matrix of user, $\mathrm{A_u}$;

while $\mathrm{E^u}$,Δ^{t} **do**
  for self-attention **do**
    $\mathrm{E^u}$ trajectory matrix of user
    $\mathrm{E}(\Delta)$ trajectory matrix of user
    Get $\mathrm{G^u}$ by using Equation(2)
  end for
  for feedforward **do**
    Characteristic output $\mathrm{G^u}$ of multi head attention mechanism
    Get $\mathrm{A^u}$ by using Equation(7)
  end for
end while
return $\mathrm{A^u}$

pattern of user preferences. We designed multiple windows to distinguish the user's behavior and extract the clock influence for different periods within a day, as shown in Figure 2. Each window $\mathrm{W_i^u}$ is defined as follows:

$$\mathrm{W_i^u} = \left\{ a_z^u \,|t_z^u \in \left[t_i^s, t_{i+1}^s \right] \right\} \tag{8}$$

$$\int_{t_i^s}^{t_{i+1}^s} f(x_t) dx = \frac{1}{12} \tag{9}$$

Here, we determine the probability distribution and probability density of the user behavior in two experimental data sets. We assume that $f(x_t)$ is the probability distribution of the user's behavior, and $\int_{t_i^s}^{t_{i+1}^s} f(x_t) dx$ is the probability density. We divide 12-time windows, and we let the probability density of each time window be $\frac{1}{12}$. Let the equivalent probability density of the time windows determine the time interval of each time window.

We can think of each window as a sub-sequence of the user's behaviors, which means that each time window represents the user's short-term preferences in the period. To capture the user's interests of each time window, we use a window state to combine the user's check-in behaviors for each time window. Let $|W_i^u|$ represent the number of check-ins within each time window. Each time window state W_i with size d is as follows:

$$\mathrm{W_i} = \frac{\sum_{n=1}^{|\mathrm{W_i^u}|} a_n^u}{|\mathrm{W_i^u}|} \tag{10}$$

Next POI Prediction

We know that POI recommendation is affected by many factors, such as geographic influence, category influence, temporal and spatial influence, et al. Therefore, to facilitate the model to search for accurate POIs, we incorporate geographic factors and popularity. Given a POI candidate set $\mathrm{L} = \{l_1, l_2, \ldots, l_{\mathrm{L}}\}$. According to the geographical factors

and popularity of POI, we gain a group of new POI candidate sets with short distance and high popularity. Firstly, we should delete some POIs that are far away from the user according to the distance of the POI. On the POI recommendation, the user behaviors follow a power-law distribution, We filter some POIs far from the user's current location and obtain a batch of initially filtered POI candidate sets. Because the POI is visited frequently, the POI is more popular. So, we eliminate some unpopular POIs according to the popularity of POIs. The popularity of a point of interest equals the result, which divides the number of users visiting the point of interest by the number of users visiting all points of interest in a day. Finally, we get a batch of new closer and more popular POI candidate sets: $\mathrm{L}' = \{l_1, l_2, \ldots, l_c\}$. Then the embedded location candidates $E(\mathrm{L}') = \{e_{l_1}, e_{l_2}, \ldots, e_{l_c}\}$. We filter out POIs that are more than 5 kilometers away from the user's current location and have a popularity of 0.5. The score of each interest point in the new candidate interest point set is calculated, and the system will recommend the top-K interest points to users according to the scores; the score r_u^{poi} of each candidate interest point is as follows:

$$r_u^{poi} = p_u \cdot e_{l_i}^{\mathrm{T}} + e_t \cdot e_{l_i}^{\mathrm{T}} \tag{11}$$

The above function is a scoring function. We select a candidate POI and calculate the possibility of the user accessing the candidate POI according to the user's current preference and the time to be predicted. For the above formula, the first term indicates the user's preference for the candidate location, the second term shows the correlation between the time that we will forecast and the candidate poi. The score is higher, and the user is more likely to visit the candidate POI. Finally, a top-K recommendation list is presented and sent to the user according to the calculated score.

Model Training

In this work, we design a personalized recommendation system model based on a temporal-aware self-attention mechanism. Next, we use the scores of the recommendation system to delimit an objective optimization function. We use the binary cross-entropy loss function to train the model, and the objective function is as follows:

$$Loss = -\sum_{u \in U} \left(\sum_{\mathcal{P}_u} \log\left(\sigma\left(r_u^{poi}\right)\right) + \sum_{\mathcal{N}_u} \left(1 - \sigma\left(r_u^{poi}\right)\right) \right) \tag{12}$$

In the above objective function, we randomly select a location that the user doesn't visit from the filtered POI as a negative sample. We use dropout technology to prevent overfitting of the optimization function.

Experiments

Experimental Setup

Dataset. In this section, we will use two public datasets to experiment with the effectiveness of our proposed model, i.e., foursquare and Gowalla. Each check-in record of datasets contains a user ID, POI ID, timestamp of a user who has visited the location, and longitude and latitude of the side. In these two datasets, we first need to filter out those users with less than ten check-in records to ensure the validity of the models. The two processed datasets in the Table 1 are as shown:

Datasets	Users	POIs	Check-ins
Forsquare	1083	9989	179468
Gowalla	4438	11235	761566

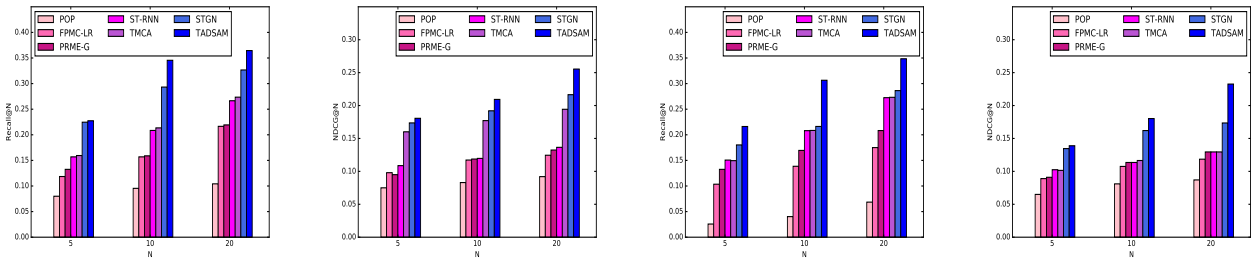
Table 1: The statistics of two datasets.

Baseline Methods. In this section, we select several baselines to compare with our model to prove the effectiveness of our proposed model. Several baseline methods are as follows:

- $\mathbf{POP}$: The method will recommend the most popular POIs to users without considering the user's personal preferences and recent visits (Ye, Zhu, and Cheng 2013).
- $\mathbf{FPMC-LR}$: This method is a personalized POI recommendation, which adopts the combination of matrix decomposition and Markov chain (Cheng et al. 2013).
- $\mathbf{PRME-G}$: The method of the personalized POI recommendation is novel, in which not only modules the whole sequence of users' trajectories, but also considers the metric embedding of the distance between the current location and the next location (Feng et al. 2015).
- $\mathbf{STRNN}$: It has based on the RNN model for the next POI recommendation. This recommendation uses the time interval and geographical distance of users' successive check-ins to module the RNN model (Liu et al. 2016).
- $\mathbf{TMCA}$: It is based on LSTM, and the method integrates a variety of contextual information with attention mechanisms, including Spatio-temporal information and category information. For equality, we remove the category information (Li, Shen, and Zhu 2018).
- $\mathbf{STGN}$: It is an improvement of LSTM. It integrates time gate and distance gate to model the Spatio-temporal context information of successive check-ins (Zhao et al. 2020).

Evaluation Metrics and Implementation Details. To evaluate the recommendation performance, we use two general evaluation metrics, i.e., Recall@N and NDCG@N. Recall@N measures the correctness of POI in top-K recommended POIs, and NDCG@N measures the quality of the top-K ranking list. The two evaluation metrics are higher, and the recommendation performance is better. In this paper, we choose N=$\{5, 10, 20\}$ to represent different results of Recall@N and NDCG@N.

In this experiment, we set the initial learning rate to 0.0035 and the dropout rate to 0.75. We set the embedded dimension d of each model as 32 and the maximum sequence length as 64. We use TensorFlow to complete the experiments and use the minimum batch Adam optimizer to optimize our model. We set the batch size 128 and a self-attention block.



(a) Recall@N on Foursquare. (b) NDCG@N on Foursquare. (c) Recall@N on Gowalla.. (d) NDCG@N on Gowalla

Figure 3: Performence comparison of models on two datasets

Dataset	Variants	Recall@5	Recall@10	Recall@20	NDCG@5	NDCG@10	NDCG20
Forsquare	$\mathrm{TADSAM_{NW}}$	0.1765	0.2143	0.2642	0.1482	0.1635	0.1863
	$\mathrm{TADSAM_{AW}}$	0.1985	0.2693	0.2836	0.1642	0.1860	0.2175
	$\mathrm{TADSAM_{NT}}$	0.2023	0.2813	0.2945	0.1760	0.1963	0.2245
	TADSAM	0.2274	0.3455	0.3622	0.1806	0.2093	0.2554
Gowalla	$\mathrm{TADSAM_{NW}}$	0.1645	0.2062	0.2473	0.1062	0.1426	0.1763
	$\mathrm{TADSAM_{AW}}$	0.2064	0.2893	0.3285	0.1143	0.1596	0.1895
	$\mathrm{TASAM_{NT}}$	0.2075	0.2932	0.3342	0.1249	0.1642	0.1968
	TADSAM	0.2162	0.3064	0.3485	0.1389	0.1802	0.2326

Table 2: Performance comparison of different variants of TADSAM

Performance Comparison

In this section, we analyze the performance of our proposed model TADSAM and compare it with other baselines, as shown in Figure 3. The relevant indicators of our model in both data sets are higher than all baselines. For example, in the Foursquare dataset, compared with the best baseline STGN, TADSAM improves by 17.8% and 9.01% in Recall@N and NDCG@N, respectively. At the same time, our model improves by 21.9% and 17.6% on the same metrics in the Gowalla dataset. In addition, we observe the comparison results of all baselines. The performance of FPMC-LR and PRME-G are better than Popu, which indicates the necessity of considering sequential information in the next POI recommendation. The ST-RNN, TMCA, and STGN perform better than FPMC-LR and PRME-G, which proves the effectiveness of LSTM in the sequential recommendation and the necessity of modeling the whole check-in sequence. From these models, we can see that spatiotemporal context information also has a particular impact on recommendation performance. Our model performs better than all baselines, which also shows the advantage of the self-attention mechanism in modeling the whole check-in sequence.

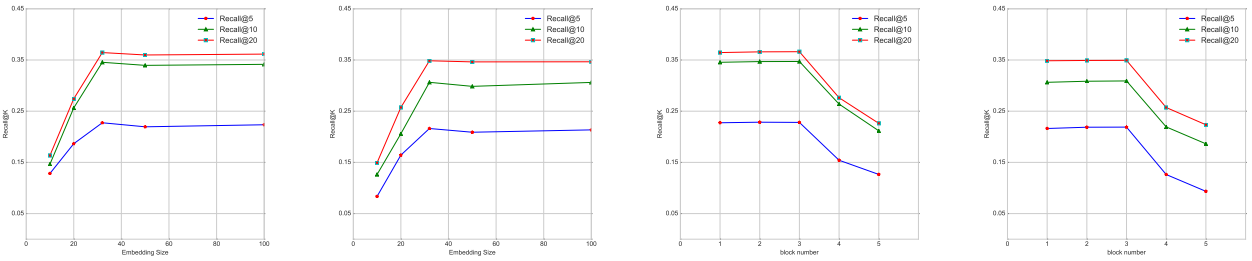
Impact of Different Components of TADSAM

titlespacing*subsection 0pt1.25ex plus 1ex minus.2ex.5ex plus .2ex To verify the effectiveness of different parts of our proposed model, we further conduct variant experiments of the model. We set the time window to understand users' preferences in different periods, which also alleviates the problem caused by data sparsity. Therefore, we need to experiment to prove its effectiveness. $\mathrm{TADSAM_{NW}}$ represents a simple model without divided windows. In this paper, we use the probability density of users' check-ins behavior on the day to allocate the time windows. To verify the effectiveness of the allocation method, we divide the time window evenly. We use $\mathrm{TADSAM_{AW}}$ to represent the time window model that has been divided average and a time interval matrix to capture the relationship between two sign-in behaviors of users. To verify the effectiveness of the time matrix, we use the vanilla self-attention mechanism to model the whole check-ins sequence of the user to capture the user's preferences, which we named $\mathrm{TADSAM_{NT}}$. The specific experimental results in Table 2 as shown. From the experimental results of three variants, our model is better than all variants, which shows the effectiveness of different components of the model.

As can be seen from the above Table 2, $\mathrm{TADSAM_{NW}}$ performance is significantly worse than the other two variants, which also proves that having different time windows can more accurately understand users' preferences. The experimental results of $\mathrm{TADSAM_{AW}}$ show that the unbalance time windows can appropriately improve the performance of the model. From the results of $\mathrm{TADSAM_{NT}}$, the time context information has some impact on the recommendation performance of the model. Overall, our model performance is better than all variants, and it also proves the effectiveness of different components of the model.

Influence of Hyper-parameters

There are two crucial Hyper-parameters in our proposed model TADSAM: embedded dimension and number of self-attention blocks. As shown in figure 4, due to space constraints, we only list the Recall@N experimental results on Foursquare and Gowalla. From the experimental results,



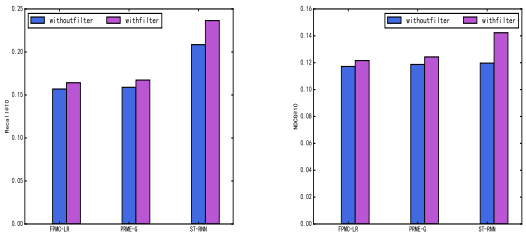
(a) d on Foursquare. (b) d on Gowalla. (c) self attention block number on Foursquare (d) self attention block number on Gowalla

Figure 4: Performence comparison of models on two datasets

the model can better extract the characteristic information of POI with the dimensionality increases. However, the model's performance tends to be stable when the size of the dimension exceeds 32. We also test the number of self-attention blocks, and the experimental results show that the increase of self-attention blocks can not improve the recommendation performance of the model. This situation may be an overfitting problem caused by the increase of model parameters with the self-attention block.

Analyze the impact of filtering

In this paper, we filter a batch of candidate sets into two layers to reduce the search space of the model. To prove the rationality of filtering, we also added filtering to other comparison baseline methods. Because ST-RNN, TMCA, and STGN are proposed based on RNN, we only conduct the comparative experiment on ST-RNN. Figure 5 shows the comparison results between filtered and unfiltered in the Foursquare dataset.



(a) Recall@10 (b) NDCG@10

Figure 5: Performance of approaches with Filter

Analysis of results

The above experimental results fully prove the effectiveness of our proposed model. In the experimental results, we found that our model is much better than the method based on the RNN model. Because the RNN model gradually reveals some disadvantages with the increasing length of the user trajectory sequence. In addition, the parallel computing ability of the RNN model is poor. It is easy to establish false dependencies because it will over assume that any adjacent item in the sequence is highly dependent. Our model applies a self-attention network. The self-attention of the powerful parallel computing ability captures internal relationships easily between the check-ins. We use the time relative matrix of user trajectory to expand the vanilla self-attention mechanism, which captures the time context information of user trajectory. To further explore the time pattern of user trajectory, we add different time windows to determine the user's preferences in each period. By comparing the results of component experiments of the model, the addition of time windows has significantly improved the recommendation performance.

Conclusion

In this paper, we construct a time-aware dynamic model for the next POI recommendation task. Firstly, we use the relative time matrix to expand self-attention blocks, which establish a dynamic temporal relationship and extract the user's diverse interest preferences. Secondly, we divide the obtained preference features into different periods and use a clever linear combination method to calculate the user's preferences. Finally, we calculate the probability of users visiting each candidate POI according to the filtered candidate POIs.

Acknowledgments

The research was supported in part by the National Natural Science Foundation of China (No. 62166004), the Guangxi Natural Science Foundation (Nos. 2018JJA170082and 2020GXNSFAA297075), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Center of Multisource Information Integration and Intelligent Processing, Shandong Provincial Natural Science Foundation (No. ZR2019BF029), the Innovation Project of Guangxi Graduate Education(No. JXXYYJSCXXM-2021-012), the Guangxi Key Laboratory of Trusted Software (No. KX202037), the Project of Guangxi Science and Technology (No. GuiKeAD 20297054), and the Guangxi Natural Science Foundation Project (No. 2020GXNSFBA297108).

References

Chang, B.; Park, Y.; Park, D.; Kim, S.; and Kang, J. 2018. Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation. In *IJCAI*, 3301–3307.

Chen, C.; Liu, Z.; Zhao, P.; Zhou, J.; and Li, X. 2018. Privacy preserving point-of-interest recommendation using decentralized matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chen, Q.; Zhao, H.; Li, W.; Huang, P.; and Ou, W. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, 1–4.

Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Twenty-Third international joint conference on Artificial Intelligence*.

Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q. 2015. Personalized ranking metric embedding for next new poi recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE.

Kong, D.; and Wu, F. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction. In *IJCAI*, volume 18, 2341–2347.

Li, M.; Zheng, W.; Xiao, Y.; Zhu, K.; and Huang, W. 2021. Exploring Temporal and Spatial Features for Next POI Recommendation in LBSNs. *IEEE Access*, 9: 35997–36007.

Li, R.; Shen, Y.; and Zhu, Y. 2018. Next point-of-interest recommendation with temporal and multi-level context attention. In *2018 IEEE International Conference on Data Mining (ICDM)*, 1110–1115. IEEE.

Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI conference on artificial intelligence*.

Liu, X.; and Aberer, K. 2013. Soco: a social network aided context-aware recommender system. In *Proceedings of the 22nd international conference on World Wide Web*, 781–802.

Liu, Y.; Pei, A.; Wang, F.; Yang, Y.; Zhang, X.; Wang, H.; Dai, H.; Qi, L.; and Ma, R. 2021. An attention-based category-aware GRU model for the next POI recommendation. *International Journal of Intelligent Systems*.

Luo, Y.; Liu, Q.; and Liu, Z. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *Proceedings of the Web Conference 2021*, 2177–2185.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, 811–820.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q. V. H.; and Yin, H. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 214–221.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Xiong, L.; Chen, X.; Huang, T.-K.; Schneider, J.; and Carbonell, J. G. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM international conference on data mining*, 211–222. SIAM.

Ye, J.; Zhu, Z.; and Cheng, H. 2013. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, 171–179. SIAM.

Ye, M.; Yin, P.; Lee, W.-C.; and Lee, D.-L. 2011. 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*, 325–334.

Yu, F.; Cui, L.; Guo, W.; Lu, X.; Li, Q.; and Lu, H. 2020. A category-aware deep model for successive poi recommendation on sparse check-in data. In *Proceedings of the web conference 2020*, 1264–1274.

Yu, Y.; and Chen, X. 2015. A survey of point-of-interest recommendation in location-based social networks. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Zhao, P.; Luo, A.; Liu, Y.; Zhuang, F.; Xu, J.; Li, Z.; Sheng, V. S.; and Zhou, X. 2020. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Zhao, S.; Zhao, T.; King, I.; and Lyu, M. R. 2017. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th international conference on world wide web companion*, 153–162.

Zhao, S.; Zhao, T.; Yang, H.; Lyu, M. R.; and King, I. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Thirtieth AAAI conference on artificial intelligence*.

Zheng, C.; and Tao, D. 2020. Attention-Based Dynamic Preference Model for Next Point-of-Interest Recommendation. In *International Conference on Wireless Algorithms, Systems, and Applications*, 768–780. Springer.