Spatio-Temporal Self-Attention Network for Next POI Recommendation

Jiacheng Ni[1], Pengpeng Zhao[1](✉), Jiajie Xu[1], Junhua Fang[1], Zhixu Li[1], Xuefeng Xian[2](✉), Zhiming Cui[3], and Victor S. Sheng[4]

[1] Institute of AI, Soochow University, Suzhou, China
ppzhao@suda.edu.cn
[2] Suzhou Vocational University, Suzhou, China
xianxuefeng@jssvc.edu.cn
[3] Suzhou University of Science and Technology, Suzhou, China
[4] Texas Tech University, Lubbock, TX, USA

Abstract. Next Point-of-Interest (POI) recommendation, which aims to recommend next POIs that the user will likely visit in the near future, has become essential in Location-based Social Networks (LBSNs). Various Recurrent Neural Network (RNN) based sequential models have been proposed for next POI recommendation and achieved state-of-the-art performance, however RNN is difficult to parallelize which limits its efficiency. Recently, Self-Attention Network (SAN), which is purely based on the self-attention mechanism instead of recurrent modules, improves both performance and efficiency in various sequential tasks. However, none of the existing self-attention networks consider the spatio-temporal intervals between neighbor check-ins, which are essential for modeling user check-in behaviors in next POI recommendation. To this end, in this paper, we propose a new Spatio-Temporal Self-Attention Network (STSAN), which combines self-attention mechanisms with spatio-temporal patterns of users' check-in history. Specifically, time-specific weight matrices and distance-specific weight matrices through a decay function are used to model the spatio-temporal influence of POI pairs. Moreover, we introduce a simple but effective way to dynamically measure the importances of spatial and temporal weights to capture users' spatio-temporal preferences. Finally, we evaluate the proposed model using two real-world LBSN datasets, and the experimental results show that our model significantly outperforms the state-of-the-art approaches for next POI recommendation.

Keywords: Self-Attention Network · Point-of-Interest · Recommender system

1 Introduction

Nowadays, due to the popularity of Location-based Social Networks (LBSN), such as Foursquare and Yelp, users can share their locations and experiences

X. Wang et al. (Eds.): APWeb-WAIM 2020, LNCS 12317, pp. 409–423, 2020.
https://doi.org/10.1007/978-3-030-60259-8_30

with friends. As a result, huge amounts of check-in data have been accumulated with an increasing need of Point-of-Interest (POI) recommendation, which also gains great research interest in recent years. Different from traditional recommendation, spatio-temporal information (i.e., time intervals and geographical distances) of users' check-ins is critical in POI recommendation. However, integrating spatio-temporal transitions into recommendation is a long-term challenge.

To model users' sequential patterns, the Markov Chain based model is an early approach for sequential recommendation. Factorizing Personalized Markov Chain (FPMC) models users' sequential information through factorizing user-item matrix and utilizing item-item transitions for next basket recommendation [14]. However, the Markov assumption is difficult to establish a more effective relationship among factors. With the development of deep learning, Recurrent Neural Network (RNN) has been successfully applied to capture the sequential user behavior patterns, some examples are Long Short-Term Memory (LSTM) [8] and Gated Recurrent Units (GRU) [4].

Some recent works have extended RNN to model the spatio-temporal information, which capture the transition patterns of user check-ins, for POI recommendation and demonstrate the effectiveness. Time-LSTM equips LSTM with time gates, which are specially designed, to model time intervals [26]. ST-RNN models local temporal and spatial contexts with time-specific transition matrices for different time intervals and distance-specific transition matrices for different geographical distances [13]. HST-LSTM combines spatio-temporal influences into LSTM model naturally to mitigate the data sparsity in location prediction problem [10]. Also by enhancing LSTM network, STGN introduced spatio-temporal gates to capture spatio-temporal information between check-ins [24]. However, RNN-based models are difficult to preserve long-range dependencies. Moreover, these methods need to compute step by step (i.e., computation of the current time step should wait for the results of the last time step), which leads to these models hard to parallelize.

Recently, a new sequential model Self-Attention Network (SAN) was proposed, which is easy to parallelize and purely based on a self-attention mechanism instead of recurrent modules [16]. It achieves state-of-the-art performance and efficiency in various sequential tasks [17,22]. The essence of the self-attention network is to capture long-term dependencies by calculating the weight of attention between each pair of items in a sequence. Actually, a pure self-attention network treats a sequence as a set, essentially without considering the order of the items in a sequence. The order of the items in a sequence is extremely important for sequential modeling tasks. To model the order information of the sequence, Tan et al. [15] applied the positional embedding to encode the sequential position information for semantic role labeling. Moreover, SASRec [9] applied position embedding into the self-attention mechanism to consider the order of the items. ATRank [25] divided items' time into intervals whose length increases exponentially, where each interval represents a time granularity. However, none of the above self-attention networks take the spatio-temporal information into consideration. It is dramatically important to consider time intervals and geographical distances between neighbor items for next POI recommendation. Hence, how to integrate time intervals and geographical distances into the self-attention network is a big challenge.

To this end, in this paper, we propose a new Spatio-Temporal Self-Attention Network (STSAN) by incorporating spatio-temporal information between check-ins into a self-attention block for next POI recommendation. Specifically, we map the time and distance intervals between two check-ins to a weight between two POIs by a decay function. In this way, POI i will get a high attention score on POI j if their spatio-temporal intervals are relatively short, and vice versa. Furthermore, in order to capture the dynamic spatio-temporal preferences of different users, we combine the spatial and temporal weights adaptively and incorporate them into the self-attention block. Experimental results show that incorporating spatio-temporal information into the self-attention block can significantly improve the performance of next POI recommendation.

To summarize, our contributions are listed as follows.

- We propose a novel framework, Spatio-Temporal Self-Attention Network (STSAN), to model time and distance intervals through a decay function and incorporate the weight values into a self-attention block for next POI recommendation.
- We introduce a simple but effective way to adaptively measure the importance of spatial and temporal weight, which can capture the spatio-temporal preferences of different users.
- We conduct extensive experiments on two representative real-world datasets, i.e., Gowalla and Foursquare, to demonstrate the effectiveness of our proposed model. The experimental results show that our proposed STSAN outperforms state-of-the-art methods, especially RNN-based models.

2 Related Work

In this section, we give a brief review of POI recommendation and discuss related work from two aspects, which are traditional POI recommendation and leveraging neural networks for POI recommendation.

2.1 Traditional POI Recommendation

Matrix Factorization (MF) is a traditional method to learn users' general taste, which factorizes a user-item rating matrix into two lower dimensionality matrices, each of which represents the latent factors of users or items [11]. Cheng et al. [1] firstly fused MF with geographical and social influence by modeling the probability of a user's check-in as a Multi-center Gaussian Model for POI recommendation. Yao et al. [20] extended the traditional MF-based approach by exploiting a high-order tensor instead of a traditional user-item matrix to model multi-dimensional contextual information. Another line of work focuses on Markov Chain based methods, which estimate an item-item transition matrix and use it for predicting next item. For instance, FPMC fuses matrix factorization and first-order Markov Chains to capture the long-term preference and short term transitions respectively [14]. FPMC-LR employs FPMC to model the personalized POI transitions and aims to recommend POIs for next hours by merging consecutive check-ins in previous hours [2]. PRME, proposed by [5], uses a metric embedding method to model the sequential patterns of POIs. He et al. [6] further proposed a tensor-based latent model, which fuses the observed successive check-in behavior with the latent behavior preference of each user to address a personalized next POI recommendation problem.

2.2 Neural Networks for POI Recommendation

With the impressive achievement of deep learning methods in different domains such as computer vision and natural language processing, there exist various methods employing and extending deep neural networks for POI recommendation. Yang et al. [18] proposed a deep neural architecture named PACE, which jointly learns the embeddings of users and POIs to predict both user preferences and various context associated with users and POIs. Zhang et al. [23] presented a unified framework named NEXT to learn user's next movement intention and incorporate meta-data information and temporal contexts for next POI recommendation. Recurrent Neural Network (RNN) has been successfully employed to capture users' dynamic preferences from the sequence of check-ins. ST-RNN [13], which employs time-specific and distance-specific transition matrices to characterize dynamic time intervals and geographical distances respectively, was first proposed to model the spatial and temporal contexts for the next location prediction. Recently, HST-LSTM was proposed to mitigate the data sparsity in the location prediction problem by combining the spatio-temporal influences into the LSTM model [10]. A more recent work STGN equipped LSTM with the new time and distance gates to model time and distance intervals between neighbor check-ins and extract users' long-term and short-term interests [24]. Though RNN-based methods are efficient in modeling sequential patterns, they still suffer from several weaknesses, such as large time consuming, being hard to parallelize and preserve long-range dependencies.

3 Our Approach

In this section, we first formalize the problem statement of next POI recommendation and then present the architecture of our Spatio-Temporal Self-Attention Network (STSAN) for next POI recommendation.

3.1 Problem Statement

In the setting of next POI recommendation, we denote a set of users as $U = \left\{u_1, u_2, ..., u_{|U|}\right\}$ and a set of POIs as $V = \left\{v_1, v_2, ..., v_{|V|}\right\}$, where $|U|$ and $|V|$

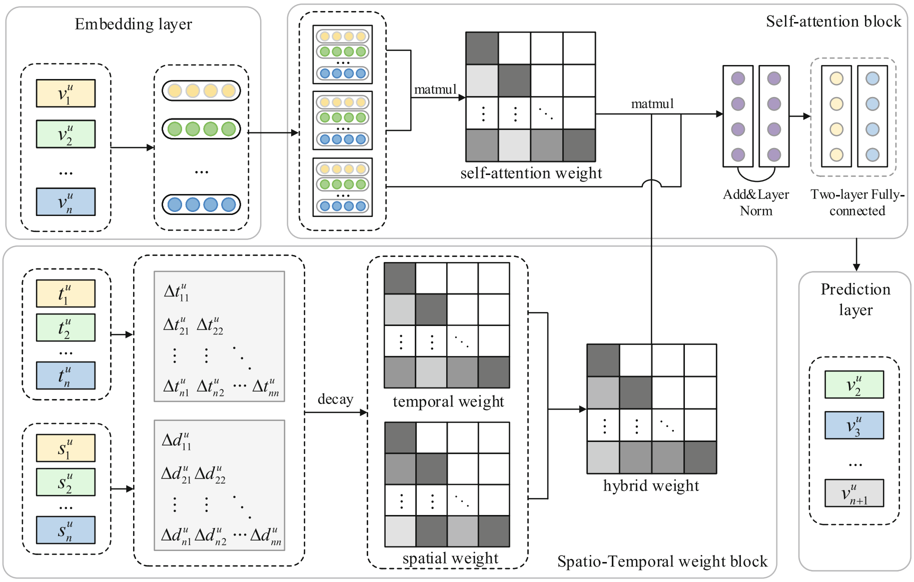


Fig. 1. The architecture of our proposed STSAN.

are the number of users and POIs respectively. For a user $u \in U$, we use $L^u = (v_1^u, v_2^u, ..., v_{|L|}^u)$ to denote a sequence of check-ins in chronological order. And each check-in record v_i^u is associated with its timestamp t_i^u and its geographic coordinates s_i^u of a POI. The goal of next POI recommendation is to predict possible top-k POIs that a user may visit at next time step, given the user historical check-ins.

3.2 Spatio-Temporal Self-Attention Network

As we mentioned above, spatial and temporal information is essential in POI recommendation. Thus, we propose a spatio-temporal self-attention network (STSAN) to integrate time and distance intervals into a self-attention block through a decay function. As shown in Fig. 1, STSAN consists of four components, i.e., Embedding layer, Spatio-Temporal weight block, Self-attention block and Prediction layer. Specifically, we first transform the sparse representation of POIs (i.e., one-hot representation) into a unique latent vector. This latent vector has a lower dimension and can capture precise semantic relationships between POIs. For spatial and temporal context, we utilize a decay function to measure the importance of time and distance intervals, forming a hybrid weight matrix. Then a user's sequential patterns are learned by a self-attention network, where the hybrid weight matrix is integrated into. Finally, we predict the next POI with a higher probability score.

Embedding Layer: As the length of user's check-in sequence is not fixed, we transform the training sequence $L^u = (v_1^u, v_2^u, ..., v_{|L|}^u)$ into a sequence with a fixed length $\hat{L}^u = (v_1^u, v_2^u, ..., v_n^u)$, where n denotes the maximum length that our model handles. If the sequence length is less than n, we employ zero-padding to fill the left side of the sequence until the sequence length is n. If the sequence length is larger than n, we just consider the most recent n check-ins. Thus we can create a POI embedding matrix $\mathbf{M} \in \mathbb{R}^{|V| \times d}$ where d is the latent dimension. Since the self-attention network ignores the positional information of previous POIs in a check-in sequence, we inject a positional matrix $\mathbf{P} \in \mathbb{R}^{n \times d}$ into the input sequence embedding. The input matrix can be defined as follows:

$$\mathbf{E} = \begin{bmatrix} \mathbf{M}_{v_1} + \mathbf{P}_1 \\ \mathbf{M}_{v_2} + \mathbf{P}_2 \\ ... \\ \mathbf{M}_{v_n} + \mathbf{P}_n \end{bmatrix} \tag{1}$$

Spatio-Temporal Weight Block: In order to capture spatio-temporal information between check-ins, given the temporal and spatial sequence associated with the user's check-ins (i.e., $(t_1^u, t_2^u, ..., t_n^u)$ and $(s_1^u, s_2^u, ..., s_n^u)$), we can calculate the temporal and spatial transition matrices $\mathbf{T}^u$ and $\mathbf{S}^u$ as follows:

$$\mathbf{T}_{ij}^u = \begin{cases} \Delta t_{ij}^u, & i \geqslant j \\ 0, & i < j \end{cases} \tag{2}$$

$$\mathbf{S}_{ij}^u = \begin{cases} \Delta d_{ij}^u, & i \geqslant j \\ 0, & i < j \end{cases} \tag{3}$$

where Δt_{ij}^u and Δd_{ij}^u are the time intervals and distance intervals between check-in v_i^u and check-in v_j^u respectively. Since the smaller the spatio-temporal intervals between two POIs, the more related the two POIs are. We use an interval-aware decay function to convert the time and distance intervals into an appropriate weight. Hence the temporal weight matrix $\hat{\mathbf{T}}^u$ and the spatial weight matrix $\hat{\mathbf{S}}^u$ can be calculated as follows:

$$\hat{\mathbf{T}}_{ij}^u = \begin{cases} g(\Delta t_{ij}^u), & i \geqslant j \\ 0, & i < j \end{cases} \tag{4}$$

$$\hat{\mathbf{S}}_{ij}^u = \begin{cases} g(\Delta d_{ij}^u), & i \geqslant j \\ 0, & i < j \end{cases} \tag{5}$$

where g is the decay function, which is defined as $g(x) = 1/log(e + x)$. Due to the nature of sequences, the model should consider only the previous POIs when predicting the current POI. Thus we employ the future blinding that ignores the influence of future POIs. That is to say, if POI v_j is behind POI v_i in a sequence, the attention score of v_i on v_j will be 0. What's more, spatial and temporal contexts are not always the same important for capturing the patterns of check-in sequence. For instance, a user may decide to visit a museum near the restaurant where he or she had dinner on the previous day. Although the time intervals of two check-ins are long (i.e., more than 24 h), the restaurant and the museum are close geographically. Thus we utilize a learnable weight factor α that the model can adjust adaptively while training to balance the influence of the spatial and temporal contexts. The hybrid weight is the adaptive combination of the temporal weight and the spatial weight, which is defined as follows:

$$\mathbf{H} = \alpha \cdot \hat{\mathbf{T}} + (1 - \alpha) \cdot \hat{\mathbf{S}}, \tag{6}$$

where $0 < \alpha < 1$. Finally we convert it through a linear projection:

$$\hat{\mathbf{H}} = \mathbf{W}\mathbf{H} + \mathbf{b}, \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a global learnable projection matrix, and $\mathbf{b} \in \mathbb{R}^{n \times n}$ is the bias, which can capture the high-order spatio-temporal transition patterns of all check-in sequences and make the model more flexible.

Self-Attention Block: We can obtain the embedding matrix $\mathbf{E}$ from the above embedding layer as the input of self-attention block, given a check-in sequence $(v_1, v_2, ..., v_n)$. In order to model the transition patterns of the sequence, we use the self-attention network proposed by [16], which can capture the relationships between POIs in the sequence. Firstly, the scaled dot-product attention is defined as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{8}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent query, key, and value respectively, d denotes the latent dimension of each POI. In the self-attention block, the query, the key and the value are equal to $\mathbf{E}$. We also convert them to three matrices through a linear projection and feed them into an attention layer:

$$\mathbf{W}_{SA} = softmax(\frac{\mathbf{E}\mathbf{W}^Q(\mathbf{E}\mathbf{W}^K)^T}{\sqrt{d}}), \tag{9}$$

$$\mathbf{F} = STSA(\mathbf{E}) = \hat{\mathbf{H}}\mathbf{W}_{SA}(\mathbf{E}\mathbf{W}^V), \tag{10}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$ are the projection matrices and $\hat{\mathbf{H}}$ is the hybrid weight matrix obtained from the spatio-temporal weight block. We argue that layer normalization is beneficial for stabilizing and accelerating at the training process [12], which is defined as follows:

$$LayerNorm(\mathbf{x}) = \tilde{\alpha} \odot \frac{\mathbf{x} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \tilde{\beta}, \tag{11}$$

where $\mathbf{x}$ is an input vector with all features of a sample, $\odot$ is an element-wise product (i.e., the Hadamard product), σ and μ are the variance and the mean of $\mathbf{x}$ respectively, $\tilde{\alpha}$ and $\tilde{\beta}$ are learned scaling factors and bias terms. Since existing methods have demonstrated that the last visited POI plays an important role on predicting next POI [7,14], we also utilize a residual connection to propagate the last POI's embedding to the final layer.

$$\hat{\mathbf{F}} = \mathbf{E} + LayerNorm(\mathbf{F}), \tag{12}$$

In order to learn more complex transitions between POIs, we apply a two-layer fully-connected layer with the ReLU activation function.

$$\mathbf{O} = ReLU(\hat{\mathbf{F}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{13}$$

where $\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2$ are model parameters.

Prediction Layer: After the self-attention block, we predict the next POI based on $\mathbf{O}_t$, given the first t POIs. We calculate the user's preference for POIs through a dot product operation as follows:

$$r_{v_i,t} = \mathbf{O}_t \mathbf{M}_{v_i}^T, \tag{14}$$

where $r_{v_i,t}$ is the relevance of POI v_i being the next POI given the first t POIs. A high score r_{v_i} means a high relevance. $\mathbf{O}_t$ denotes the t-th line of $\mathbf{O}$, and $\mathbf{M} \in \mathbb{R}^{|V| \times d}$ is a POI embedding matrix. Note that the model inputs a sequence $(v_1, v_2, ..., v_n)$ and its excepted output is a 'shifted' version of the same sequence $(v_2, v_3, ..., v_{n+1})$. After training process, we can generate next POI recommendations by the last row of matrix $\mathbf{O}$.

3.3 Network Training

During the training process, we apply the binary cross-entropy loss as the optimization objective function of our model as follows:

$$-\sum_{v_i \in L^u} \sum_{t \in [1,2,...,n]} [log(\sigma(r_{v_i,t})) + \sum_{v_j \notin L^u} log(1 - \sigma(r_{v_j,t}))], \tag{15}$$

In each training epoch, for each target POI v_i in each sequence, we randomly sample a negative POI v_j. And we use Adam to optimize the parameters in our model, which is a variant of gradient descent and can adapt the learning rate for each parameter by performing a little update for frequent parameters and heavily update for infrequent parameters.

3.4 Complexity Analysis

Space Complexity: Compared with SASRec [9], whose total number of parameters is $O(|V|d + nd + d^2)$ from the embedding layer, self-attention layers, feed-forward networks and layer normalization, our proposed model needs to consider the time and the distance intervals of all POI pairs in a user's check-in sequence. Thus the space complexity of our model inevitably grows but is acceptable, which is $O(|U|n + |V|d + nd + d^2)$.

Time Complexity: The time complexity of our model consists mostly of the spatio-temporal weight block and the self-attention block. Hence it is $O(|U|n^2 + n_{epoch}n^2d)$, where n_{epoch} is the number of epochs at the training process. If the total number of users $|U|$ is equal to $n_{epoch}d$, our model will be about twice slower than the original self-attention network. Although the time

Table 1. Statistics of the datasets after preprocessing

Dataset	#User	#POI	#Check-in	Density
Gowalla	51089	106735	3136810	0.058%
Foursquare	3376	11860	584028	1.459%

complexity of our model increases to some extent for the computation of spatial and temporal transition matrices, the parallelism nature of the self-attention network has not been destroyed. Thus our model is also much faster than those RNN-based methods, whose computation on time step t should wait for the results of time step $t - 1$.

4 Experiments

In this section, we first describe datasets, evaluation metrics and baseline methods used in our experiments. Then we evaluate the performance of STSAN compared with the state-of-the-art baseline methods and analyze our experimental results.

4.1 Datasets

We conducted experiments on two public available LBSNs datasets (i.e., *Gowalla*[1] and *Foursquare*[2]), which have user-POI interactions, timestamps of check-ins and locations of POIs. *Gowalla* is a location-based social networking website where users share their locations by checking-in and the dataset was generated worldwide from February 2009 to October 2010 [3]. *Foursquare* contains check-ins in New York and Tokyo collected from April 2012 to February 2013 [19]. Each check-in of the two datasets is associated with its timestamp and geographic coordinates. For both datasets, we remove users with fewer than 10 check-ins and POIs visited by fewer than 10 users. The statistics of the two datasets are summarized in Table 1. We sort each user's check-ins according to the chronological order and take the early 70% of users' check-ins as the training data, the last 30% as the testing data.

4.2 Evaluation Metrics and Implementation Details

To evaluate the recommendation performance of STSAN and the baseline methods, we adopt two widely used evaluation metrics, i.e., Recall and Normalized Discounted Cumulative Gain (NDCG). Recall measures the accuracy of the recommendation. For an instance in the testing set, Recall@K is 1 if the visited POI appears in the set of top-K recommended POIs, and 0 otherwise. NDCG

[1] http://snap.stanford.edu/data/loc-gowalla.html.

[2] https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

is a position-aware metric, which assigns larger weights on higher positions. In this paper, we choose $K = \{5, 10\}$ to illustrate different results of Recall@K and NDCG@K. In the default version of STSAN, we set the embedding size d to 100 on *Gowalla* and 50 on *Foursquare*. The maximum sequence length n is set to 50 on both datasets. Following [9], we implement our experiments in *Tensorflow* and apply the mini-batch Adam optimizer to optimize the parameters in our model. We set the learning rate to 0.001 initially. The number of epochs is set to 200, the batch size is 128 and we apply only one self-attention block.

4.3 Baselines

We compare our proposed model STSAN with the following representative methods, which are briefly described as follows.

- **RNN:** This is a traditional recurrent architecture, which only considers the sequence of POIs in its hidden unit while ignoring additional contextual information [21].
- **ST-RNN:** It replaces the single transition matrix in RNN to model spatio-temporal contexts by including time-specific and distance-specific transition matrices during model learning [13].
- **HST-LSTM:** It combines spatio-temporal influences into a LSTM model naturally to mitigate the data sparsity in the location prediction problem [10].
- **STGN:** Enhancing LSTM network, STGN introduces the spatio-temporal gates to capture the spatio-temporal relationships between successive check-ins [24]. We use its variation named STGCN, which uses couple input and forget gates.
- **SASRec:** This is a strong sequential model, which applies self-attention mechanisms to capture long-term sequential semantics [9].
- **T-SAN:** This is a variant of our proposed model with only temporal context.
- **S-SAN:** This is a variant of our proposed model with only spatial context.
- **STSAN:** This is our proposed model.

4.4 Performance Comparison

In this subsection, we analyze the performance of the proposed STSAN, comparing with eight baselines on two datasets. Our experimental results in terms of Recall@K and NDCG@K are shown in Table 2. From the table we can see the following observations: Compared with the standard RNN, ST-RNN, HST-LSTM, and STGN perform better on the two datasets. This confirms that incorporating time and distance information into the standard RNN architecture is critical for improving the POI recommendation performance. SASRec achieves a better performance, comparing with RNN-based methods. This confirms the advantages of self-attention mechanisms to model sequential patterns. Although ST-RNN, HST-LSTM and STGN take the spatio-temporal information into consideration,

Table 2. Experimental results of STSAN and baselines. The best performing method in each row is boldfaced.

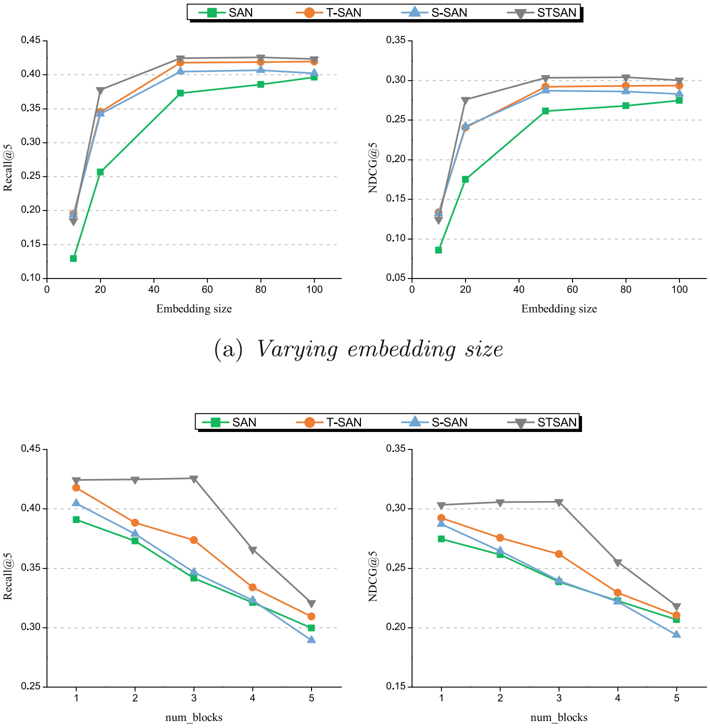
Dataset	Method	Topk = 5		Topk = 10	
		Recall	NDCG	Recall	NDCG
Gowalla	RNN	0.0893	0.0674	0.1136	0.0756
	ST-RNN	0.0967	0.0706	0.1229	0.0792
	HST-LSTM	0.1128	0.0816	0.1433	0.0905
	STGN	0.1348	0.1020	0.1714	0.1139
	SAN	0.2093	0.1440	0.2812	0.1672
	T-SAN	0.2660	0.1896	0.3418	0.2140
	S-SAN	0.2369	0.1699	0.3092	0.1934
	STSAN	**0.3113**	**0.2287**	**0.3699**	**0.2478**
Foursquare	RNN	0.1206	0.0809	0.1799	0.0999
	ST-RNN	0.1306	0.1087	0.1867	0.1197
	HST-LSTM	0.2067	0.1546	0.2662	0.1738
	STGN	0.2366	0.1736	0.3018	0.1920
	SAN	0.3966	0.2746	0.5140	0.3126
	T-SAN	0.4177	0.2922	**0.5286**	0.3282
	S-SAN	0.4046	0.2871	0.5149	0.3229
	STSAN	**0.4243**	**0.3033**	0.5221	**0.3350**

they perform worse than SASRec, which may be due to the weakness of RNN architectures. Finally, our proposed model STSAN achieves the best recommendation performance regardless of the datasets and the evaluation metrics. This proves that STSAN can better capture long-term and short-term preferences like SASRec. Although SASRec has also achieved a better result than RNN-based methods, it cannot incorporate the time and the distance intervals, which are essential for POI recommendation. Our proposed STSAN outperforms SASRec as the time and the distance intervals can be correctly combined into the self-attention block.

4.5 Discussions

In this subsection, we explore the effectiveness of spatio-temporal components in our architecture via an ablation study and investigate the influence of hyper-parameters.

Effectiveness of Spatio-Temporal Context: In order to explore the effectiveness of spatial and temporal context, we illustrate the performance of SASRec, T-SAN, S-SAN and STSAN in Table 2. SASRec applies the original self-attention block following [9]. Both T-SAN and S-SAN are variants of our proposed model with only temporal context or spatial context respectively. For T-SAN, we replace the hybrid weight matrix **H** as the temporal transition matrix



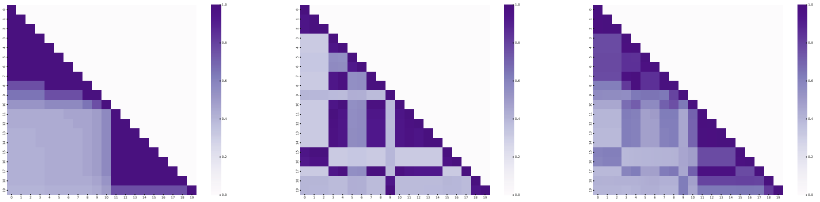
(a) *Varying embedding size*

(b) *Varying number of self-attention block*

Fig. 2. Performance with different embedding sizes and number of self-attention block.

$\hat{\mathbf{T}}$ calculated by Eq. (2). For S-SAN, the hybrid weight matrix $\mathbf{H}$ is replaced by the spatial transition matrix $\hat{\mathbf{S}}$ calculated by Eq. (3). As we can see from the experimental results, both T-SAN and S-SAN outperform SASRec. This suggests that incorporating the temporal weight and the spatial weight into self-attention block yields a significant improvement in POI recommendation. Moreover, STSAN combines spatial and temporal context through dynamically learning to give the proper weight to spatial and temporal transition matrices. Thus, it achieves the best performance among these methods. This means that time and distance intervals are both critical for improving the recommendation performances.

Influence of Hyper-parameters: Figure 2(a) shows the performance of four self-attention based models with different embedding sizes on *Foursquare*. As we can see from our experimental results, high dimensions can capture more characteristic information of POIs. On the other hand, the performance of four models is almost unchanged when the embedding size exceeds 50. This demonstrates that the model with a larger dimension cannot capture more useful patterns of POIs. The original self-attention mechanism (Transformer) proposed by [16]

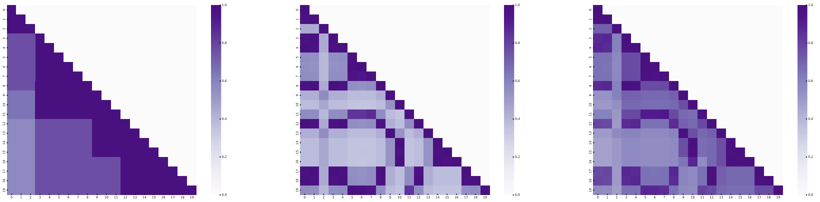


(a) Heat map of the temporal weight

(b) Heat map of the spatial weight

(c) Heat map of the hybrid weight

Fig. 3. Visualization of the spatio-temporal weight at random sampled sequences of user A on *Foursquare*.



(a) Heat map of the temporal weight

(b) Heat map of the spatial weight

(c) Heat map of the hybrid weight

Fig. 4. Visualization of the spatio-temporal weight at random sampled sequences of user B on *Foursquare*.

stacks several self-attention blocks to capture complicated sequential patterns. We conduct the experiments of our model with varying the number of self-attention blocks on *Foursquare*. Figure 2(b) shows that a larger number of self-attention blocks cannot significantly improve the recommendation performance. This may be because the hierarchical self-attention structure may increase the number of model parameters and the model may suffer over-fitting.

4.6 Visualization of Attention Weight

As we mentioned above, different users may have different spatio-temporal interests. In this subsection, we seek to reveal the different influence of time and distance intervals on the check-in sequences of different users through the visualization of three weight matrices (i.e., the temporal weight matrix $\hat{\mathbf{T}}$, the spatial weight matrix $\hat{\mathbf{S}}$ and the hybrid weight matrix $\mathbf{H}$). We randomly choose two check-in sequences among all users and convert these three weight matrices of each sequence into heat maps as shown in Fig. 3 and 4, which only shows the last 20 positions of each sequence. From the visualizations, we can conclude as follows.

Firstly, the heat map of the temporal weight indicates that more recent POIs will obtain more attention (a higher weight) due to the decay function. In reality, two POIs that a user visited in a short time tend to have similar characteristics.

Similarly, two POIs with short distance intervals are related to each other, which can be depicted from the heat map of the spatial weight. The heat map of the hybrid weight is a fusion of the two heat maps above.

Secondly, as we can see from Fig. 3(c) and Fig. 4(c), the heat map of the hybrid weight of user A is more similar to the heat map of the temporal weight. This indicates that user A tends to be more time focused. On the contrary, the heat map of the hybrid weight of user B is more similar to the spatial weight. This demonstrates that user B may prefer to walk out so that closer POIs can obtain more attention.

Overall, the visualizations of the spatio-temporal weight show the effectiveness of our proposed model in dynamically capturing users' spatial and temporal preferences.

5 Conclusion

In this paper, we proposed a spatio-temporal self-attention based model named STSAN for next POI recommendation. We incorporated the time and distance intervals between check-ins in a sequence to enhance the recommendation performance of standard self-attention networks. Specifically, we designed a decay function to obtain the weight of spatio-temporal intervals. Furthermore, we combined the spatial and the temporal weight dynamically to capture the spatio-temporal interests of the user through an adaptive factor. Extensive experimental results on two real-world datasets showed that STSAN outperforms the state-of-the-art methods. This demonstrates the effectiveness of our STSAN in modeling the spatio-temporal information into the self-attention network. In the future, we will consider richer context information, such as social relationships and textual contents to further improve the performance for next POI recommendation.

Acknowledgements. This research was partially supported by NSFC (No. 61876117, 61876217, 61872258, 61728205), Open Program of Key Lab of IIP of CAS (No. IIP 2019-1) and PAPD.

References

1. Cheng, C., Yang, H., King, I., Lyu, M.R.: Fused matrix factorization with geographical and social influence in location-based social networks. In: AAAI (2012)
2. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: successive point-of-interest recommendation. In: IJCAI, pp. 2605–2611 (2013)
3. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: SIGKDD, pp. 1082–1090. ACM (2011)
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
5. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new POI recommendation. In: IJCAI, pp. 2069–2075 (2015)

6. He, J., Li, X., Liao, L., Song, D., Cheung, W.K.: Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In: AAAI, pp. 137–143 (2016)
7. He, R., McAuley, J.: Fusing similarity models with Markov chains for sparse sequential recommendation. In: ICDM, pp. 191–200. IEEE (2016)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: ICDM, pp. 197–206. IEEE (2018)
10. Kong, D., Wu, F.: HST-LSTM: a hierarchical spatial-temporal long-short term memory network for location prediction. In: IJCAI, pp. 2341–2347 (2018)
11. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
12. Lei Ba, J., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
13. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: AAAI, pp. 194–200 (2016)
14. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp. 811–820. ACM (2010)
15. Tan, Z., Wang, M., Xie, J., Chen, Y., Shi, X.: Deep semantic role labeling with self-attention. In: AAAI, pp. 4929–4936 (2018)
16. Vaswani, A., et al.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
17. Xu, C., et al.: Graph contextualized self-attention network for session-based recommendation. In: Kraus, S. (ed.) IJCAI, pp. 3940–3946 (2019)
18. Yang, C., Bai, L., Zhang, C., Yuan, Q., Han, J.: Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In: SIGKDD, pp. 1245–1254. ACM (2017)
19. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Trans. **45**(1), 129–142 (2015)
20. Yao, L., Sheng, Q.Z., Qin, Y., Wang, X., Shemshadi, A., He, Q.: Context-aware point-of-interest recommendation using tensor factorization with social regularization. In: SIGIR, pp. 1007–1010. ACM (2015)
21. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: SIGIR, pp. 729–732. ACM (2016)
22. Zhang, T., et al.: Feature-level deeper self-attention network for sequential recommendation. In: Kraus, S. (ed.) IJCAI, pp. 4320–4326 (2019)
23. Zhang, Z., Li, C., Wu, Z., Sun, A., Ye, D., Luo, X.: Next: a neural network framework for next POI recommendation. arXiv preprint arXiv:1704.04576 (2017)
24. Zhao, P., et al.: Where to go next: a spatio-temporal gated network for next POI recommendation. In: AAAI, pp. 5877–5884 (2019)
25. Zhou, C., et al.: Atrank: an attention-based user behavior modeling framework for recommendation. In: AAAI, pp. 4564–4571 (2018)
26. Zhu, Y., et al.: What to do next: modeling user behaviors by time-LSTM. In: IJCAI, pp. 3602–3608 (2017)