CANS-Net: Context-Aware Non-Successive Modeling Network for Next Point-of-Interest Recommendation

Qiang Cui*
Yafeng Zhang
Jinpeng Wang

ABSTRACT

Point-of-Interest (POI) recommendation is an important task in location-based social networks. It facilitates the sharing between users and locations. Recently, researchers tend to recommend POIs by long- and short-term interests. However, existing models are mostly based on sequential modeling of successive POIs to capture transitional regularities. A few works try to acquire user's mobility periodicity or POI's geographical influence, but they omit some other spatial-temporal factors. To this end, we propose to jointly model various spatial-temporal factors by context-aware non-successive modeling. In the long-term module, we split user's all historical check-ins into seven sequences by day of week to obtain daily interest, then we combine them by attention. This will capture temporal effect. In the short-term module, we construct four short-term sequences to acquire sequential, spatial, temporal, and spatial-temporal effects, respectively. Attention of interest-level is used to combine all factors and interests. Experiments on two real-world datasets demonstrate the state-of-the-art performance of our method.

CCS CONCEPTS

• **Information systems → Collaborative filtering**; **Recommender systems**.

KEYWORDS

context-aware, non-successive, point-of-interest

ACM Reference Format:
Qiang Cui, Yafeng Zhang, and Jinpeng Wang. 2018. CANS-Net: Context-Aware Non-Successive Modeling Network for Next Point-of-Interest Recommendation. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Location-based Social Networks (LBSNs), such as Foursquare, and Yelp, enable users to share check-in experiences and opinions on the Point-of-Interests (POIs) they have checked in. As one of the core services of LBSNs, POI recommendation [5] concentrates on mining user's check-ins, location-based information, contextual information, and social relationships to recommend POIs for users. The POI recommendation is beneficial not only to the improvement of user experiences and social networking services, but also to the advertising between potential locations and users.

A large number of methods have been proposed to solve the POI recommendation. Recently, many researchers tend to combine long- and short-term interests to boost performance. The long-term interest usually means user's interest is modeled with all his/her historical check-ins, while the short-term interest is modeled with recent check-ins [13]. These interests are usually encoded by feeding the user's successive check-ins to a encoder network, e.g., a long short-term memory (LSTM) to capture transitional regularities. The way of organizing inputs as successive check-ins works well as users' recent check-ins are more likely to impact his/her next check-in [6, 8, 10]. However, in the case when user's check-ins are abundant, it is difficult for the encoder to encode all POIs and represent them into a fixed-size vector. Moreover, only a small subset of historical check-ins are highly related to user's next visit, so the successive check-ins would weaken the signals of the related part. Only a few methods attempt to get out of this limitation by adopting non-successive POIs, such as DeepMove [3] captures periodicity from all historical POIs by attention, and LSTPM [8] exploits geographical connections among non-successive POIs by examining POI distance.

Fortunately, the context of check-in, e.g., spatial or temporal information when a user is visiting the POI recommendation system, provides some hints to the way of modeling historical check-ins. Firstly, the geographic scope of user activity is usually constrained and users are more likely to go to other locations nearby where they have been [5, 14]. For instance, some users like to visit the mall and hospital near their home. Secondly, user activity is also constrained by time. Users likely show different preferences at different times and visit similar places at similar times. For instance, some users probably go to different places for vacation only during leisure time, such as weekends. Based on the above analysis, to model the historical check-ins, the spatial-temporal context can be used to retrieve related part from user's historical check-ins, and to capture different influential factors, the historical check-ins can be organized in a context-aware non-successive way.

In this paper, we propose a **C**ontext-**A**ware **N**on-**S**uccessive modeling **Net**work (**CANS-Net**) for POI recommendation. Given spatial-temporal context at current time step, we construct various sequences with non-successive recent POIs. As for long-term interest, we first split a user's all historical check-ins into seven sequences, according to temporal context called day of week. Then

*Qiang Cui is the corresponding author.

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

we obtain seven user's daily interests by average pooling and acquire long-term interest by attention. This contributes to learn user's temporal mobility, such as periodicity. As for short-term interest, we obtain four sequences by contexts called time slot and geohash. Four LSTMs are used to obtain four short-term interests, which can capture sequential, spatial, temporal, spatial-temporal effects, respectively. As for overall interest, we firstly compute five attention weights without softmax for five interests. Then interests and weights are multiplied and concatenated together. The overall interest and a candidate POI along with context are sent to a multilayer perceptron to obtain the final probability for the POI. The main contributions are summarized as follows:

- We propose a broader assumption that a user's next check-in highly correlates with his/her context-aware recent check-ins. It can coverage more significant influential factors for POI recommendation.
- We propose several context-aware non-successive techniques according to spatial-temporal context, which can better capture user's long and short-term interests.

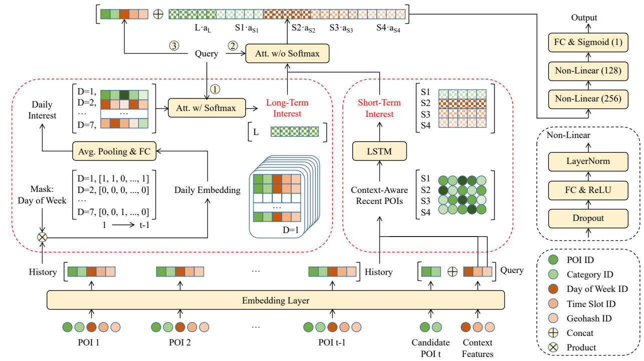

Figure 1: The architecture of CANS-Net. Seven daily interests are captured from user's all historical POIs and form long-term interest by attention. Four context-aware POI sequences are constructed and make up four short-term interests. Long- and short-term interests are concatenated by attention weights, and a MLP is used to obtain predicted probability for a candidate POI.

2 RELATED WORK

In this section, we briefly review some recent studies for POI recommendation and summarize their characteristics.

Combining long- and short-term interests attracts much attention recently and it achieves state-of-the-art. DRCF captures long-term interest by collaborative filtering and adds short-term interest by recurrent neural network (RNN) [6]. STGN modifies LSTM by devising distance gate and time gate to update interest [13]. It is worth noting that spatial-temporal relations are captured between successive POIs. DeepMove applies RNN-based method to capture transitional regularities and designs a historical attention module to exploit mobility periodicity [3]. LSPL and its extended version PLSPL use attention layer to obtain long-term interest and use LSTM to model recent sequential behaviors for short-term interest [9, 10]. LSTPM uses all trajectories with various techniques to obtain long-term interest. A LSTM and a geo-dilated LSTM are used to capture short-term interest with geographical relations between non-successive POIs [8].

Obviously, these methods promote the development of POI recommendation. As we can see, the two most popular technologies are LSTM [4] and attention. Acquiring sequential transition through successive POIs is the mainstream direction. Only a few works try to model non-successive POIs. Modeling non-successive POIs with spatial-temporal context would be probably promising.

3 METHODOLOGY

In this section, we introduce the proposed CANS-Net in detail. The overall architecture is illustrated in Fig. 1, including an embedding layer, long- and short-term modules, and a prediction layer. Four context-aware short-term sequences are illustrated in Fig. 2.

3.1 Embedding Layer

First of all, we introduce rich features. Let $U = \{u_1, u_2, ..., u_{|U|}\}$, $P = \{p_1, p_2, ..., p_{|P|}\}$, and $C = \{c_1, c_2, ..., c_{|C|}\}$ denote the set of users, POIs, and categories, respectively, where each POI has a category $c \in C$ as its metadata. Next is spatial-temporal information. As each POI has a geocoding of latitude and longitude, we transform it into the geohash-integer version. In fact, we use geohash-5 $G = \{g_1, g_2, ..., g_{|G|}\}$ to express spatial information as regions for POIs, and each POI only belongs to one geohash-5 or region. Spatial features are represented by day of week and time slot, where we have 7 days in a week $W = \{w_1, w_2, ..., w_7\}$ and 24 slots of hours in a day $M = \{m_1, m_2, ..., m_{24}\}$. Then, a user's check-in at i-th time step can be expressed as $h^i = (p^i, c^i, w^i, m^i, g^i)$.

Afterwards, we give problem formulation. Given a user's all historical check-ins $H = \{h^1, h^2, ..., h^{t-1}\}$ as well as his/her current time and region (mapped as (w^t, m^t, g^t)), our goal is to select top-k POIs that the user would check-in. For example, we already know that a user is now near a certain mall, we need to predict which POI he/she will go to.

Before making recommendations, we represent each feature by a dense vector. Each feature has a unique ID and is represented by one-hot vector at first. Each kind of feature has its own space with a learned embedding. Correspondingly, the i-th check-in's dense vector is $\boldsymbol{h}^i = \text{CONCAT}(\boldsymbol{p}^i, \boldsymbol{c}^i, \boldsymbol{w}^i, \boldsymbol{m}^i, \boldsymbol{g}^i) = [\boldsymbol{p}^i, \boldsymbol{c}^i, \boldsymbol{w}^i, \boldsymbol{m}^i, \boldsymbol{g}^i]$.

3.2 Long-Term Module

The context called day of week is used to split a user's all check-ins into seven sequences which are non-successive. We can learn seven daily interests to increase model capacity, capture daily periodicity and integrate them into the long-term interest by attention.

Daily Interest. Daily interest can acquire user's daily mobility pattern. Firstly, we construct seven mask sequences that are as long as user history. We use symbol '1' to indicate on which day a check-in occurred. For example, if a user's first check-in is on Sunday, then the first position of the mask sequence is '1' and the first positions of the other six are all '0'. In this way, we can obtain the seven mask sequences. Through the element-wise product between the seven mask sequences and the user's all historical check-ins, seven user's daily embeddings can be obtained. Next, through average pooling and then a shared fully connected layer on each daily embedding, we end up with seven daily interests, denoted as $\boldsymbol{L}_w = [\boldsymbol{l}_{w1}; \boldsymbol{l}_{w2}; ...; \boldsymbol{l}_{w7}]$. Specifically, for a user's each daily interest at each time step, we calculate the cumulative sum of the daily embedding and the cumulative sum of the mask, and then divide the two to achieve average pooling.

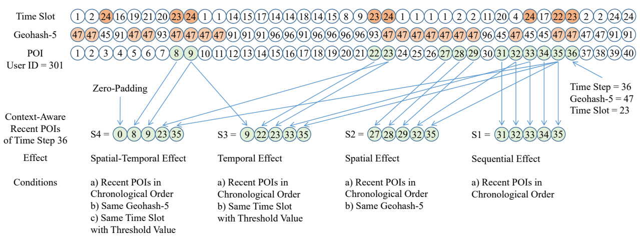

Figure 2: Illustration of obtaining four short-term sequences for a certain time step by spatial-temporal context.

Long-Term Interest. Long-term interest can extract important information from daily interest. We first construct a query $\boldsymbol{q}$, and then aggregate seven daily interests $\boldsymbol{L}_w$ through attention. A query at t-th time step is built by the concatenation of a candidate POI and the already known context (w^t, m^t, g^t):

$$\boldsymbol{q}^t = [\boldsymbol{p}^t, \boldsymbol{c}^t, \boldsymbol{w}^t, \boldsymbol{m}^t, \boldsymbol{g}^t], \tag{1}$$

where we have many candidates to construct various queries, but only one is positive. Then, we apply the widely used Bahdanau attention[1] [1] to aggregate daily interest $\boldsymbol{L}_w^t$:

$$\begin{aligned} a_{l,j}^t &= \boldsymbol{v}_{l,a}^{\mathrm{T}} \tanh(\boldsymbol{V}_{l,1}\boldsymbol{q}^t + \boldsymbol{V}_{l,2}\boldsymbol{l}_{wj}^t), \\ \boldsymbol{a}_l^t &= \mathrm{softmax}([a_{l,1}^t, a_{l,1}^t, ..., a_{l,7}^t]), \\ \boldsymbol{l}^t &= \sum_{j=1}^{7} a_{l,j}^t \boldsymbol{l}_{wj}^t, \end{aligned} \tag{2}$$

where $\boldsymbol{l}^t$ is the t-th long-term interest.

3.3 Short-Term Module

Short-term module plays an important role to capture the sequential effect, spatial effect, temporal effect, and so on. Besides, it also needs to effectively combine diverse short-term interests to capture the most important factors.

Context-Aware Recent POIs. Given current spatial-temporal context, called time slot m and region g, four short-term sequences are extracted from the user's all historical check-ins. The effect and rule condition of each sequence are illustrated in detail in Fig. 2. The first sequence S_1 is successive, and the others S_2, S_3, and S_4 are non-successive.

Separate Short-Term Interest. For each short-term sequence at t-th time step S_k^t, we apply a LSTM [4] to obtain its interest:

$$\boldsymbol{s}_k^t = \mathrm{LSTM}(S_k^t), \tag{3}$$

where $\boldsymbol{s}_k^t$ is a separate short-term interest at t-th time step. As each sequence implies a specific mobility pattern, we use four LSTMs instead of one to avoid collision.

[1]https://www.tensorflow.org/tutorials/text/nmt_with_attention

Aggregated Short-Term Interest. As short-term sequences are constructed by different rule conditions, we aggregate them by weighted concatenation instead of weighted sum. First, we still use a Bahdanau attention to compute weights.

$$a_{s,k}^t = \boldsymbol{v}_{s,a}^{\mathrm{T}} \tanh(\boldsymbol{V}_{s,1}\boldsymbol{q}^t + \boldsymbol{V}_{s,2}\boldsymbol{s}_k^t). \tag{4}$$

However, we abandon the $softmax$ on the four short-term weights [7, 15] in order to improve the expression ability of important influential factors. Then, we concatenate the weighted interests:

$$\boldsymbol{s}^t = \mathrm{CONCAT}(\{a_{s,k}^t \boldsymbol{s}_k^t \mid k \in [1, 2, 3, 4]\}), \tag{5}$$

where $\boldsymbol{s}^t$ is the t-th short-term interest.

3.4 Prediction Layer

We compute the probability for a certain candidate in a prediction layer, consisting of an input, a multilayer perceptron (MLP) and a loss.

Similar to computing aggregated short-term interest, we update Eqs. 4 and 5 to aggregate one long-term interest ($\boldsymbol{l}^t$) and four short-term interests ($\boldsymbol{s}_1^t$, $\boldsymbol{s}_2^t$, $\boldsymbol{s}_3^t$, and $\boldsymbol{s}_4^t$). Five new attention weights $a_{x,\{l,s1\sim4\}}^t$ are computed and final weighted interest is

$$\boldsymbol{x}^t = [a_{x,l}^t \boldsymbol{l}^t, a_{x,s1}^t \boldsymbol{s}_1^t, ..., a_{x,s4}^t \boldsymbol{s}_4^t]. \tag{6}$$

To avoid overfitting, we add a dropout layer for query and five interests in this attention. Before sending $\boldsymbol{x}^t$ to the MLP, we first concatenate it and query $\boldsymbol{q}^t$:

$$input^t = [\boldsymbol{q}^t, \boldsymbol{x}^t], \tag{7}$$

where $input^t$ is the input for the first layer of MLP. The following process of MLP is illustrated in detail in Fig. 1. Finally, we use the popular cross-entropy as our loss:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{8}$$

where N is the number of all candidates, including positives and negatives, $y_i \in \{0, 1\}$ and $p(y_i)$ are label and probability for a candidate respectively.

4 EXPERIMENTS

4.1 Experimental Settings

Datasets. We choose two public real-world datasets and check-ins are collected from New York city (NYC) and Tokyo (TKY)[2] [12]. Each check-in contains user ID, POI ID, category ID, latitude, longitude, and UTC time. For both datasets, we delete POIs that are checked by less than 5 times, and we only keep the most recent 500 check-ins for each user. Following the previous work [6], we adopt the leave-one-out evaluation methodology. We treat each user's last POI as ground truth, compute probabilities for all POIs during test and rank them.

Baselines. We compare our CANS-Net with several baselines, which are good at capture long- and short-term interests. **LSTM** [4] is a widely used method for sequential modeling. **STGN** [13] controls both interests by devising spatial-temporal gates. **LSTPM** [8] designs a non-local network for long-term interest and a geo-dilated LSTM for short-term interest with geographical effect. **PLSPL** [10]

[2]http://www-public.it-sudparis.eu/~zhang_da/pub/dataset_tsmc2014.zip

Table 1: Performance comparison on two datasets. The best value and the runner-up are boldfaced and underlined, respectively.

		NYC					TKY				
		Acc@1	Acc@5	Acc@10	MRR@5	MRR@10	Acc@1	Acc@5	Acc@10	MRR@5	MRR@10
Baselines	LSTM	0.1816	0.3991	0.4663	0.2610	0.2701	0.1506	0.3553	0.4511	0.2256	0.2384
	STGN	0.2071	0.4051	0.4645	0.2829	0.2909	0.2134	0.4352	0.5161	0.2956	0.3066
	LSTPM	0.1846	0.3693	0.4484	0.2549	0.2660	0.1928	0.3889	0.4622	0.2647	0.2746
	PLSPL	0.2188	0.4367	0.5162	0.3000	0.3109	0.2054	0.4252	0.5225	0.2853	0.2983
Ours	**CANS-Net$_L$**	0.3135	0.5543	0.6243	0.4055	0.4151	0.2842	0.5340	0.6287	0.3773	0.3900
	CANS-Net$_S$	0.3790	0.5798	0.6363	0.4543	0.4619	0.3464	0.5872	0.6708	0.4386	0.4498
	CANS-Net	**0.3807**	**0.5850**	**0.6454**	**0.4584**	**0.4666**	**0.3523**	**0.5963**	**0.6772**	**0.4455**	**0.4564**
	improvement (%)	74.00	33.96	25.03	52.80	50.08	71.52	37.02	29.61	50.7	48.86

uses user-based attention and LSTMs for two interests, respectively. And it adopts another user-based attention to obtain overall predicted probability.

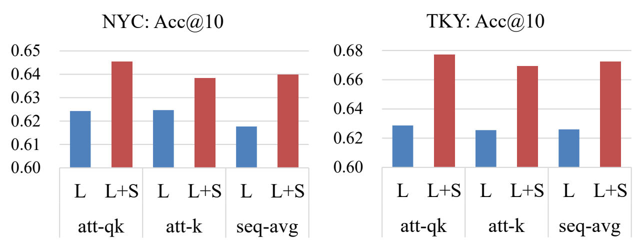

Figure 3: Examine different settings for long-term interest.

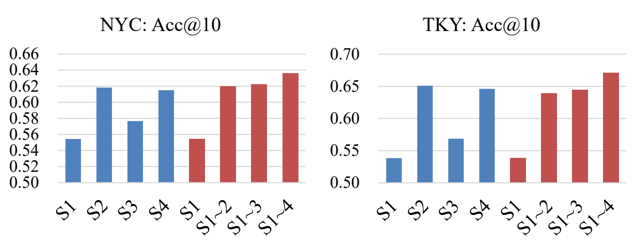

Figure 4: Ablation study of short-term interest.

Parameters and Evaluations. The dimensions of POI, category, day of week, time slot, and geohash-5 are 64, 32, 8, 16, and 32, respectively. Following the previous work [2], we make these dimensions approximately proportional to the logarithm of the number of their unique ID values. The number of negative candidates is 20. The dimension of interest is 64. The dropout rates of attention introduced in Section 3.4 are 0.0 and 0.1 on NYC and TKY, respectively, and the dropout rate of MLP is fixed as 0.1. To evaluate different methods, we apply Acc@k [11] and MRR@k. Our model has two variants, called CANS-Net$_L$ which only has long-term module and CANS-Net$_S$ which only has short-term module.

4.2 Results and Discussions

Performance. The results are listed in Table 1 and our CANS-Net significantly outperforms the others. In view of the comparison with baselines, our CANS-Net achieves a great improvement on two datasets. Specifically, for Acc@1, CANS-Net outperforms the runner-up by about 70%. For Acc@5 and Acc@10, our model improves by about 30%. The overall improvements of MRR@5 and MRR@10 are about 50%. In light of model variant, CANS-Net is able to perform better if we have more interests. The CANS-Net$_L$ and CANS-Net$_S$ have already performed well, but CANS-Net can still achieve an absolute value increase of about 0.005 when compared to CANS-Net$_S$.

Non-Successive Modeling vs. Successive Modeling in Long-Term Module. In order to examine how to deal with the user's all historical check-ins, we set up three different settings, called **att-qk**, **att-k**, and **seq-avg**. The setting att-qk is introduced in Section 3.2. The setting att-k means we do not use query in attention. The setting seq-avg averages pooling for the entire sequence. The comparison is illustrated in Fig. 3, and symbols 'L' and 'L+S' refers to the performance of CANS-Net$_L$ and CANS-Net, respectively. Obviously, the setting att-qk is better than the other two. In summary, it is best to have a query in the attention. Besides, non-successive modeling with proper techniques such as context and attention can defeat successive modeling when acquiring long-term interest.

Ablation Study of Short Interest. We compare four short-term interests to explore the four influential factors, and investigate the effectiveness of attention introduced in Section 3.3. The two kinds of results are illustrated in Fig. 4 by blue color and red color, respectively. For the first discussion, the four interests perform differently. The performance of $S2$ and $S4$ are evidently better than the other two. This means the spatial effect and spatial-temporal effect are more influential than sequential effect and temporal effect. Perhaps when a user chooses a POI in a certain region, the POIs he/she has visited before in this region have a greater impact. For the second discussion, overall performance is better than the performance of any one short-term interest. Besides, when we gradually add four short-term interests, the overall performance is getting better and better. These results show the effectiveness of our attention.

5 CONCLUSION AND FUTURE WORK

In this work, we propose a context-aware non-successive modeling network (CANS-Net) for next POI recommendation. In long-term module, we construct daily interest by day of week, and aggregate them to capture long-term interest by attention. In short-term module, we extract four short-term sequences by time slot and geohash to model sequential, spatial, temporal, and spatial-temporal effects. The attention without softmax on weights can effectively capture important factors. The experiment demonstrates the excellent modeling capability of our CANS-Net. In the future, we would design more automated modeling methodology to replace the pre-operations of constructing different sequences.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.
[3] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW*. 1459–1468.
[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[5] Md Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, Mohammed Eunus Ali, et al. 2020. A Survey on Deep Learning Based Point-Of-Interest (POI) Recommendations. *arXiv preprint arXiv:2011.10187* (2020).
[6] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A deep recurrent collaborative filtering framework for venue recommendation. In *CIKM*. 1429–1438.
[7] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. In *CIKM*. 2669–2676.
[8] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *AAAI*, Vol. 34. 214–221.
[9] Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. 2019. Long-and short-term preference learning for next POI recommendation. In *CIKM*. 2301–2304.
[10] Yuxia Wu, Ke Li, Guoshuai Zhao, and QIAN Xueming. 2020. Personalized long-and short-term preference learning for next POI recommendation. *TKDE* (2020).
[11] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based poi embedding for location-based recommendation. In *CIKM*.
[12] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.
[13] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S Sheng, and Xiaofang Zhou. 2019. Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. In *AAAI*, Vol. 33. 5877–5884.
[14] Shenglin Zhao, Irwin King, and Michael R Lyu. 2016. A survey of point-of-interest recommendation in location-based social networks. *arXiv preprint arXiv:1607.00647* (2016).
[15] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *SIGKDD*. 1059–1068.