Next POI Recommendation Method Based on Category Preference and Attention Mechanism in LBSNs

Xueying Wang[1], Yanheng Liu[1,2], Xu Zhou[2,3](✉), Zhaoqi Leng[4], and Xican Wang[4]

[1] College of Computer Science and Technology, Jilin University, Changchun, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China
zhoux16@jlu.edu.cn
[3] Center for Computer Fundamental Education, Jilin University, Changchun, China
[4] College of Software, Jilin University, Changchun, China

Abstract. Focusing on learning the user's behavioral characteristics during check-in activities, the next point of interest (POI) recommendation is to predict user's destination to visit next. It is important for both the location-based service providers and users. Most of the existing studies have not taken full advantage of spatio-temporal information and user category preference, these are very important for analyzing user preference. Therefore, we propose a next POI recommendation algorithm named as CPAM that integrates category preference and attention mechanism to comprehensively structure user mobility patterns. We adopt the LSTM with multi-level attention mechanism to get user POI preference, which studies the weight of different contextual information of each check-in, and the different influence of each check-in the sequence to the next POI. In addition, we use LSTM to capture the user's category transition preference to further improve the accuracy of recommendation. The experiment results on two real-world Foursquare datasets demonstrate that CPAM has better performance than the state-of-the art methods in terms of two commonly used metrics.

Keywords: LSTM · Next POI recommendation · Contextual information · Location-based social networks · Attention mechanism

1 Introduction

With the rapid development of mobile networks, location-based social networks (LBSNs) are also widely used in recent years, such as Foursquare and Facebook [4]. Users can share their location and life by checking in locations. According to users' historical check-in information, it is convenient to construct users' movement trajectory and dig out their movement patterns. The next point of interest (POI) recommendation has become one of the most important tasks in

B. Li et al. (Eds.): APWeb-WAIM 2022, LNCS 13422, pp. 12–19, 2023.
https://doi.org/10.1007/978-3-031-25198-6_2

LBSNs and has a broad range of applications. Its primary objective is to predict the next POI that a user is likely to visit at a given time based on the user's check-in sequence [10]. The next POI recommendation plays a significant role in location-based services, and it can not only promotes customer experiences, but also helps improve the quality of location-related business services [2].

User's transition preference for POI category reflects user's mobility patterns at category level, in order to take full advantage of contextual information, we propose a next POI recommendation algorithm (CPAM) that combines category preference and attention mechanism. Experimental results on two real-world datasets demonstrate that CPAM algorithm is significantly better than other six comparative algorithms in terms of Recall and Map.

2 Related Work

Earlier approaches are to model the user's movement patterns through Markov chains to solve the sparse problem [6]. But existing Markov chain based methods are difficult to capture longer sequence contexts. In recent years, there has been a trend of methods applying deep learning for recommendation system. For example, Liu et al. proposed the ST-RNN which considers spatio-temporal information on the basis of RNN [5]. But RNN is not suitable for building long sequences. Subsequently, Zhang et al. proposed iMTL with multi-task learning framework based on LSTM [11], which comprehensively considered the category and space-time information in trajectory sequence. In addition, Some studies found that aggregating different contextual information (such as social relationship, time, location, etc.) into POI recommendation methods can alleviate data sparseness [12]. Attention mechanism can capture the degree of influence of different components [1]. It is also widely used for the next POI recommendation. Combining LSTM and attention mechanism can distinguish the differing degrees of influences that each time step may have on the next check-in. Huang et al. proposed ATST-LSTM, which adds attention mechanism on the basis of LSTM [3]. Li et al. proposed a codec framework, which could automatically learn the deep spatio-temporal representation of historical check-ins, but it did not consider the impact of spatio-temporal transition on check-in [14]. Wu et al. considered the long and short term preferences of users separately, and integrated attention mechanism, geographical location and category information of POI into the LSTM network [13]. The above studies all employ the attention mechanism to achieve better next POI recommendation performance.

3 Proposed Method

The model is mainly composed of three modules, as shown in Fig. 1. (1) Category module based on LSTM is to obtain the user's preference representation at category level; (2) POI module based on self-attention LSTM network to get user's preference representation at POI level; (3) Output layer is to generate a ranked list of next POIs.

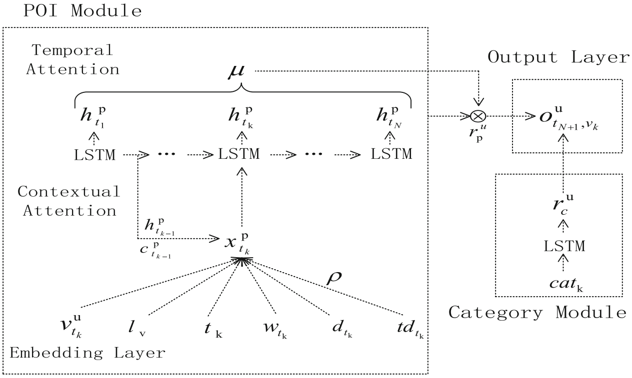

Fig. 1. The proposed CPAM framework

3.1 Category Module

Category sequence is the representation of check-in sequence at the semantic level. It reflects user's mobile preference at category level. Category module is designed to infer user category preference and participates in POI recommendation as an auxiliary function.

We learn the user's category preference $\mathbf{r}_c^u$ from category sequence $C_u = \left\{C_{t_1}^u, C_{t_2}^u, \cdots, C_{t_N}^u\right\}$, each element of C_u is denoted as $C_{t_k}^u = (u, cat_v)$. It indicates that the user u visits a POI v of category cat_v at time t_k. The latent vector of the category module is defined as follows.

$$\mathbf{x}_{t_k}^c = \mathbf{W}^C \mathbf{cat}_v + \mathbf{b}^C \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{E \times E}$ is the weight matrix, where E is the dimension of the hidden vector, $\mathbf{b} \in \mathbb{R}^E$ is bias. Then, $\mathbf{x}_{t_k}^c$ is input into the LSTM network to infer the hidden state $\mathbf{h}_{t_k}^c$ of user u.

$$\mathbf{h}_{t_k}^c = LSTM\left(\mathbf{x}_{t_k}^c, \mathbf{h}_{t_{k-1}}^c\right) \tag{2}$$

$$\mathbf{r}_c^u = \mathbf{h}_{t_N}^c \tag{3}$$

where $LSTM(\cdot)$ captures the sequential correlation of categories, $\mathbf{h}_{t_{k-1}}^c$ is the LSTM hidden state, which indicates the check-in category up to t_{k-1}.

3.2 POI Module

Embedding Layer. The historical check-in sequence of user u consists of the check-in tuple $A_{t_k}^u = \left(u, v_{t_k}^u, l_v, cat_v, t_k, w_{t_k}\right)$, we use it to learn the user's preference at the POI level. the latent vector of the embedding layer of the POI preference module is defined as follows:

$$\tilde{\mathbf{x}}_{t_k}^p = \mathbf{W}_v \mathbf{v}_{t_k}^u + \mathbf{W}_l \mathbf{l}_v + \mathbf{W}_t \mathbf{t}_k + \mathbf{W}_w \mathbf{w}_{t_k} + \mathbf{W}_d \mathbf{d}_{t_k} + \mathbf{W}_{td} \mathbf{td}_{t_k} + \mathbf{b} \tag{4}$$

where $\mathbf{v}_{t_k}^u$ is POI number, $\mathbf{l}_v$ is POI location, $\mathbf{t}_k$ is access timestamp, $\mathbf{w}_{t_k}$ is the day of the week, $\mathbf{d}_{t_k}$ is distance between $l_{t_k}^u$ and $l_{t_{k-1}}^u$, $\mathbf{td}_{t_k}$ is time difference between t_k and t_{k-1}. $\tilde{\mathbf{x}}_{t_k}^p$ is sent to the contextual attention layer.

Contextual Attention. Each feature of the embedded layer marks an attribute of the current check-in, and the extent to which these attributes affect the current check-in is different. Therefore, the proportion of different contextual information is studied with contextual attention mechanism in the current check-in.

$\tilde{\mathbf{x}}(i, t_k)$ represents the i-th attribute of the k-th historical check-in. $\rho(i, t_k)$ indicates the weight of the i-th feature in the k-th check-in. The softmax function is used for normalization.

$$\tilde{\rho}(i, t_k) = tanh\left(\mathbf{W}_i\left[\mathbf{h}_{t_{k-1}}^p, \mathbf{c}_{t_{k-1}}^p\right] + \mathbf{W}_i^{\tilde{x}}\tilde{\mathbf{x}}(i, t_k) + \mathbf{b}_i\right) \tag{5}$$

$$\rho(i, t_k) = \frac{exp(\tilde{\rho}(i, t_k))}{\sum_{i=1}^{I} exp(\tilde{\rho}(i, t_k))}, 1 \leq i \leq I \tag{6}$$

where I is the number of attributes, $\mathbf{h}_{t_{k-1}}^p$ is the LSTM hidden state, $\mathbf{c}_{t_{k-1}}^p$ is the LSTM cell state. Then, $\tilde{\mathbf{x}}(i, t_k)$ is multiplied by $\rho(i, t_k)$ to obtain the embedding vector, the updated attribute embedding vector is connected to obtain the aggregation $\mathbf{x}_{t_k}^p$ of the embedding layer based on contextual attention mechanism. $\mathbf{x}_{t_k}^p$ is sent to LSTM to infer the hidden state $\mathbf{h}_{t_k}^p$ at t_k.

$$\mathbf{x}(i, t_k) = \tilde{\mathbf{x}}(i, t_k) \times \rho(i, t_k) \tag{7}$$

$$\mathbf{x}_{t_k}^p = \sum_{i=1}^{I} \mathbf{W}(i)\,\mathbf{x}(i, t_k) + \mathbf{b} \tag{8}$$

$$\mathbf{h}_{t_k}^p = LSTM\left(\mathbf{x}_{t_k}^p, \mathbf{h}_{t_{k-1}}^p\right) \tag{9}$$

Temporal Attention. We use the temporal attention mechanism to adaptively select relevant historical check-ins activities to achieve a better recommendation of the next POI.

Let $\mathbf{H}^p$ be a matrix composed of all hidden vectors $\left\{\mathbf{h}_{t_1}^p, \mathbf{h}_{t_2}^p, \cdots \mathbf{h}_{t_N}^p\right\}$, where N is the length of the historical check-in sequence. The weight vector μ of historical check-in is generated through the temporal attention mechanism.

$$\mu = \frac{exp\left(g\left(\mathbf{h}_{t_k}^p, \mathbf{q}^u\right)\right)}{\sum_{i=1}^{N} exp\left(g\left(\mathbf{h}_{t_k}^p, \mathbf{q}^u\right)\right)} \tag{10}$$

the attention function $g\left(\mathbf{h}_{t_k}^p, \mathbf{q}^u\right)$ is as follows.

$$g\left(\mathbf{h}_{t_k}^p, \mathbf{q}^u\right) = \frac{\mathbf{h}_{t_k}^p\left(\mathbf{q}^u\right)^T}{\sqrt{E}} \tag{11}$$

where $\mathbf{q}^u$ is the query information. Then multiply the resulting weight vector μ by $\mathbf{H}^p$ to get user u's preference representation at the POI level.

$$\mathbf{r}_p^u = \sum_{k=1}^{N} \mu_k \mathbf{h}_{t_k}^p \tag{12}$$

3.3 Output Layer

We filter out a suitable POI for each user from all the accessed POIs, which must meet any of the following conditions: (1) the POI is the one that the user has visited before; (2) the POI is close to the POI that the user recently accessed to; (3) it is the POI that is visited most by all users, i.e., popular POI.

In the output layer, we calculate the POI preference obtained by the POI module and the category preference obtained by the category module with the selected POI v_k, and use the Softmax function to perform normalization, and the probability of all candidate POI is obtained as bellows.

$$o_{t_{N+1},v_k}^u = \frac{exp\left(\mathbf{r}_p^u \mathbf{v}_k \times \mathbf{r}_c^u \mathbf{cat}_v\right)}{\sum_{k=1}^{N} exp\left(\mathbf{r}_p^u \mathbf{v}_k \times \mathbf{r}_c^u \mathbf{cat}_v\right)} \tag{13}$$

3.4 Network Training

Bayesian Personalized Ranking (BPR) is used to define loss function for training the LSTM network in the category and POI modules [7], since BPR trains network models by learning pair-wise sorting and can effectively utilize information about POIs that the user does not visit. The data used for the category and POI modules consists of a set of triplets sampled from the original data, each triplet containing the user u and a pair of positive and negative samples.

The loss function of the category module is:

$$lc = \sum_{(c>c')\in\Omega_c} ln\left(1 + e^{-\left(o_t^c - o_t^{c'}\right)}\right) \tag{14}$$

where c' is the negative category of c, Ω_c is the training example, o_t^c is the predicted probability of user u visiting the POI of category c at time t, and $o_t^{c'}$ is the predicted probability of user u visiting the POI of category c'.

The loss function of the POI module is:

$$lp = \sum_{(v>v')\in\Omega_p} ln\left(1 + e^{-\left(o_t^v - o_t^{v'}\right)}\right) \tag{15}$$

By integrating the loss functions and regularization terms of the two modules, we strive to minimize the total loss function:

$$l = lc + lp + \frac{\varepsilon}{2}\left|\left|\Theta^2\right|\right| \tag{16}$$

where ε is the regularization coefficient, Θ is the set of model parameters to learn. AdaGrad is employed to optimize network parameters since it can significantly improve the robustness of stochastic gradient descent.

4 Experiments

To verify the proposed method, we compare it with six baselines on two public real world check-in datasets named as Charlotte (CHA) and New York (NYC) from Foursquare. All the algorithms are coded in Python 3.8 and the framework is TensorFlow 2.3.1.

4.1 Datasets

The check-in data of CHA [11] is collected from January 2012 to December 2013 and the check-in data of NYC [9] is collected from April 2012 to February 2013. The CHA dataset includes 20,939 check-in records and NYC dataset includes 227,428 check-in records.

In this study, each check-in record consists of user, POI, the POI location, the check-in timestamp, the POI category, and the day of the week. Similar to the work of Zhang et al. [11], we use the first 90% of check-ins of each user as the training set and the last 10% as the test set.

4.2 Results and Analysis

We demonstrate the effectiveness of the CPAM method compared to the following six methods: PMF [8], ST-RNN [5], Time-LSTM [14], ATST-LSTM [3], LSPL [13], iMTL [11]. To investigate the effectiveness of CPAM, we focused on answering two research questions. **RQ1:** Can the performance of CPAM be improved by using attention mechanisms and category preference? **RQ2:** Can each component of CPAM help improve recommendation performance?

Table 1. The recommendation result of different methods on CHA and NYC dataset

Datasets	**CHA**				**NYC**			
Criteria	***Rec@5***	***Rec@10***	***MAP@5***	***MAP@10***	***Rec@5***	***Rec@10***	***MAP@5***	***MAP@10***
PMF	0.0868	0.1343	0.0181	0.0413	0.0322	0.125	0.0222	0.0263
ST-RNN	0.0890	0.1879	0.0333	0.061	0.0476	0.1964	0.025	0.0312
Time-LSTM	0.0943	0.2142	0.0625	0.0709	0.0794	0.2238	0.0372	0.0558
ATST-LSTM	0.1703	0.3083	0.0699	0.0819	0.1824	0.3269	0.0721	0.0821
iMTL	0.2138	0.3634	0.0833	0.0909	0.2184	0.3801	0.099	0.1057
LSPL	0.2539	0.3701	0.0909	0.1057	0.2702	0.3901	0.0925	0.1129
CPAM	**0.2785**	**0.4016**	**0.0921**	**0.1162**	**0.2777**	**0.4484**	**0.101**	**0.1234**

Answer to RQ1: Table 1 shows the performance of all methods, and the results of two evaluation indicators when k is set to 5 and 10 are listed. It is found that the recall and MAP value of Time-LSTM is higher than that of ST-RNN, which infers that LSTM has better performance than RNN in long sequence modeling. What's more, ATST-LSTM performs better compared with Time-LSTM, which indicates that adding the spatio information and attention mechanism of check-in sequence is beneficial to the modeling of POI check-in sequence. Compared to baseline methods, CPAM considers users' preferences for POI and category at the same time and it mines as much information contained in user check-in sequences as possible. So CPAM we proposed has a better recommendation performance.

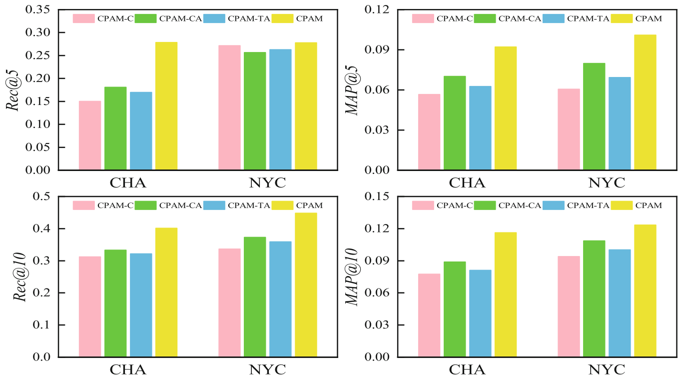

Fig. 2. The recommendation performance comparison of CPAM and its variants on CHA and NYC dataset

Answer to RQ2: In order to verify the performance brought by considering the contribution of category module, the contribution of contextual attention mechanism and the contribution of temporal attention mechanism, we design three different variants of CPAM: (1) CPAM-C removes the category module, that is, users' preferences at the category level are no longer considered. (2) CPAM-CA removes contextual attention from the POI module. (3) CPAM-TA removes the temporal attention mechanism from the POI module. Figure 2 illustrates the performance of CPAM and its variants. It is found that CPAM has better performance than its variants. The three components are indispensable, and they together improve the next POI recommendation performance.

5 Conclusion

A next POI recommendation algorithm based on category preference and attention mechanism is put forward in this paper. The proposed method CPAM considers the user's category preference and POI preference respectively, mines the user's movement behavior patterns through multi-level attention mechanism. The experimental results show CPAM performs better than the other six comparative methods. In the future, we further study the influence of user comment information for next POI recommendation.

Acknowledgements. This work is supported by the National Nature Science Foundation of China (62172186) and the Fundamental Research Funds for the Central Universities, JLU under Grant No.93K172021Z02.

References

1. Chen, W., Yue, L., Li, B., Wang, C., Sheng, Q.Z.: DAMTRNN: a delta attention-based multi-task RNN for intention recognition. In: Li, J., Qin, S., Li, X., Wang, S., Wang, S. (eds.) ADMA 2019. LNCS (LNAI), vol. 11888, pp. 373–388. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35231-8_27
2. Dai, S., Yu, Y., Fan, H., Dong, J.: Spatio-temporal representation learning with social tie for personalized poi recommendation. Data Sci. Eng. **7**(1), 44–56 (2022)
3. Huang, L., Ma, Y., Wang, S., Liu, Y.: An attention-based spatiotemporal LSTM network for next POI recommendation. IEEE Trans. Serv. Comput. **14**(6), 1585–1597 (2021)
4. Jiao, W., Fan, H., Midtbø, T.: A grid-based approach for measuring similarities of taxi trajectories. Sensors **20**(11), 3118 (2020)
5. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: a recurrent model with spatial and temporal contexts. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA, pp. 194–200. AAAI Press (2016)
6. Lv, Q., Qiao, Y., Ansari, N.: Big data driven hidden markov model based individual mobility prediction at points of interest. IEEE Trans. Veh. Technol. **66**(6), 5204–5216 (2017)
7. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. CoRR abs/1205.2618 (2012)
8. Wu, Y., Li, K., Zhao, G., Qian, X.: Long- and short-term preference learning for next POI recommendation. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3–7, 2019, pp. 2301–2304. ACM (2019)
9. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. IEEE Trans. Syst. Man Cybern. Syst. **45**(1), 129–142 (2015)
10. Yu, D., Wanyan, W., Wang, D.: Leveraging contextual influence and user preferences for point-of-interest recommendation. Multimedia Tools Appl. **80**(1), 1487–1501 (2021)
11. Zhang, L., Sun, Z., Zhang, J., Lei, Y.: An interactive multi-task learning framework for next POI recommendation with uncertain check-ins. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 3551–3557. ijcai.org (2020)
12. Zhao, P., Zhu, H., Liu, Y., Xu, J., Li, Z.: Where to go next: A spatio-temporal gated network for next POI recommendation. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, pp. 5877–5884. AAAI Press (2019)
13. Zheng, C., Tao, D., Wang, J.: Memory augmented hierarchical attention network for next point-of-interest recommendation. IEEE Trans. Comput. Soc. Syst. **8**(2), 489–499 (2021)
14. Zhu, Y., Li, H., Liao, Y., Wang, B.: What to do next: Modeling user behaviors by time-lstm. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, pp. 3602–3608. ijcai.org (2017)