Personalized POI recommendation using deep reinforcement learning

Jing Huang, Tong Zhang

jing_h@whu.edu.cn, zhangt@whu.edu.cn

Wuhan University

Abstract. As an important location based service, next Point-Of-Interest (POI) recommendation has been widely utilized in helping people discover attractive and interesting locations. However, the sparsity of check-in data, the cold start issue and complicated contextual and semantic relationships between users and POIs bring severe challenges. To cope with these challenges, we develop a novel deep reinforcement learning based personalized POI recommendation framework. Within the proposed framework, a joint graph embedding model is proposed to compute users' dynamic preferences, accounting for inter-user relationships, historical check-in sequence, and category information of visited POIs. We are working on the experiments of applying the proposed POI recommendation framework on two typical real-world location-based social network datasets.

Keywords. POI recommendation, deep reinforcement learning, graph embedding

1. Introduction

With the rapid prevalence of smart mobile devices, users can post their locations and location-related contents on various social platforms, which underpin the building of online virtual society and generate massive user behavior data. Massive information of POIs has imposed serious information overload on mobile users. The selection dilemma faced by regular users has motivated the development of next POI recommendation algorithms in both computer science and location-based service communities.

Published in "Proceedings of the 16th International Conference on Location Based Services (LBS 2021)", edited by Anahid Basiri, Georg Gartner and Haosheng Huang, LBS 2021, 24-25 November 2021, Glasgow, UK/online.

https://doi.org/10.34726/1761 | © Authors 2021. CC BY 4.0 License.

Unlike traditional recommender systems, next POI recommendation faces several challenges. (1) **Data sparsity**. Due to high cost and privacy protection, users' check-in data generated in location based social networks (LBSNs) is much sparser than their rating data generated for music and movies. The performance of most existing collaborative filtering (CF) methods is significantly degraded by the sparse user-item matrix. (2) **Cold Start** is a common but critical problem in the field of recommender system. In POI recommendation tasks, there are broadly two main cold start issues: locations that have never been visited are called cold-start POIs, and users who have never visited any location are called cold-start users. Traditional research predicts a user's next POI based on the individual's sufficient historical data, which are difficult to be applied in cold start scenarios (Mazumdar et al. 2020). (3) **Dynamic User Preferences**. As time goes by or environments change, users' preferences for POIs and items may vary (Hu et al. 2021). We aim to deal with all above challenges and propose an effective data-driven POI recommendation method.

The primary contributions of this paper are summarized as follows:

- We develop a joint graph-based embedding model to learn the representations of POIs, timestamps, function zones, user comment and trajectory data in a shared low-dimension space. To track the changes of user preferences, we model the dynamic user preferences based on the learnt embedding of POIs.
- We propose a POI recommendation framework based on deep reinforcement learning techniques, which are used to produce a list of POIs by incorporating real-time feedbacks from users and unified representations of various factors such as temporal effect, geographical influence, semantic information and sequential features.

2. Methodology

2.1. Problem Statement

Definition 1. (**User check-in profile**) For each user u, we create a user profile vector D_u, which is a set of check-in activities associated with u and sorted by timestamp. A check-in activity is expressed as a tuple (u, v, l_v, t, M_v), denoting user u visits POI v at time t, l_v is v's geographical location including latitude and longitude coordinates and M_v is the set of words describing v, such as POI categories, reviews, and ratings. The dataset D used in our model includes the profiles of all users, $D = \{D_u : u \in U\}$.

Definition 2. (**POI-POI Graph**) A POI-POI graph, denoted as $G_{vv} = (V \cup V, \varepsilon_{vv})$, V is a collection of POIs and ε_{vv} is a collection of edges between POIs. In the given comment set C_{review}, the corresponding text set w_v is extracted for each POI v. w_v is then be used to calculate the topic feature $\overrightarrow{w_v}$ using the Latent Dirichlet Allocation (LDA) model (Chen et al. 2020). The cosine distance is used to calculate the similarity between the topic feature vectors of each POI. If the cosine similarity s_{ij} between the topic feature vectors of v_i and v_j is greater than the threshold α, then v_i and v_j are connected to an edge e_{ij} and the weight w_{ij} is set to s_{ij}.

Definition 3. (**POI-Zone Graph**) POI-Zone graph, denoted as $G_{vz} = (V \cup Z, \varepsilon_{vz})$, is a bipartite graph, where ε_{vz} is a set of edges between POIs and function zones. Considering the semantic connectivity among spatial entities, regions segmented based on their typical core functions are termed as function zones (Wang et al. 2020). If POI v_i is located in function zone z_j, there will be an edge e_{ij} connecting v_i and z_j, otherwise none. The weight w_{ij} is set to 1 if edge e_{ij} exists.

Definition 4. (**POI-Time Graph**) POI-Time graph, denoted as $G_{vt} = (V \cup T, \varepsilon_{vt})$, is a bipartite graph, where ε_{vt} is a set of edges between POIs and pre-defined time slots. If POI v_i is visited by users at time slot t_j, there will be an edge e_{ij} connecting v_i and t_j, otherwise none. The weight w_{ij} is set to frequency of POI v_i checked in at time slot t_j.

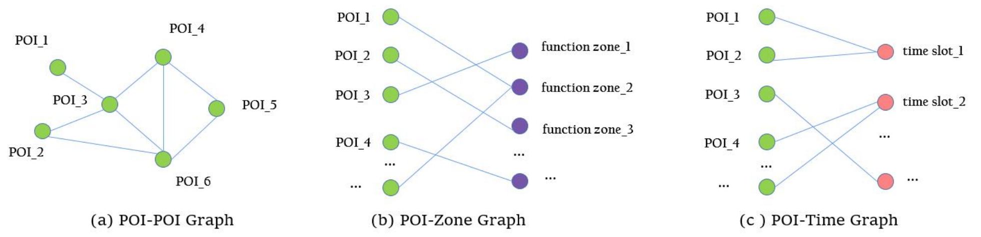

Figure 1. Bipartite graphs.

Problem Definition. (**Location-based Recommendation**) Given a user check-in sequence dataset D and a query user u with her current location l and time t (denoted as a query $q(u, l, t)$), our goal is to recommend top-k POIs that user u would be interested in and enhance her experience with these POIs.

2.2. Model description

We first propose a joint graph-based embedding approach using bipartite graph and produce the integrated representation of a user, spatial entities (e.g., POIs, activity types, function zones), sequence and semantic information in a latent space. Then, we present a POI recommendation framework based on graph embeddings using an actor-critic reinforcement learning framework (Zhao et al. 2018).

Joint Graph Embedding Learning. We adopt a bipartite graph embedding model with a joint training algorithm (Xie et al. 2016) to embed the POI-POI graph, POI-Zone graph and POI-Time graph into a low dimension latent space, in which visited POI, visit time slot and associated function zone of each user, are represented as a low dimensional vector, $\vec{v}$, $\vec{t}$ and $\vec{z}$, respectively. An intuitive way is to minimize the sum of all objective functions as following:

$$O = O_{vt} + O_{vv} + O_{vz} \tag{1}$$

To optimize the objective function Eqn. (1), we first merge all the edges in the three sets e_{vv}, e_{vt}, e_{vz} together, and then update the corresponding embedding model by alternatively sampling an edge from it at each step.

Deep Reinforcement Learning Based POI Recommendation. We model the POI recommendation task as a Markov Decision Process (MDP) and use Deep Reinforcement Learning (DRL) to learn and update the optimal recommendation strategies during the interactions between users and POIs. A recommender agent (RA) interacts with environment $\mathcal{E}$ over a sequence of time steps. We formally define the tuple of five elements $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ of **MDP Environment** $\mathcal{E}$, which informs the actor (i.e., user) about its states in the POI graph and possible actions to take. The environment also rewards the actor if the current policy fits the observed user interactions. (1) **State** $\mathcal{S}$. The state s is to describe the environment composed by users and spatial entities (e.g., POIs, activity types, function zones); (2) **Action** $\mathcal{A}$. An action $a = \{a^1, \cdots, a^M\} \in \mathcal{A}$ is to recommend a list of M POIs to a user based on the current state s; (3) **Transition** $\mathcal{P}$. Transition $p(s'|s, a)$ defines the state transition from s to s' when recommender agent (RA) takes action a; (4) **Reward** $\mathcal{R}$. After the RA takes an action a at the state s, i.e., recommending a list of POIs to a user, the user browses these POIs and provides her feedbacks. She can skip (not visit) or visit the recommended POIs. We consider the length of stay duration at POIs as an implicit feedback, and the agent receives immediate reward $r(s, a)$ according to the user's feedbacks; (5) **Discount factor** γ. $\gamma \in [0,1]$ defines the discount factor when we measure the present value of future reward.

At each time step, the RA takes an action $a \in \mathcal{A}$ according to $\mathcal{E}$'s state $s \in \mathcal{S}$, and receives a reward $r(s, a)$. According to action a, the environment $\mathcal{E}$ updates its state to s' with transition probability $p(s'|s, a)$. The recommendation task can then be solved via reinforcement learning (Afsar et al. 2021), which aims to find a recommendation policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which maximizes the cumulative reward for the POI recommender system. In this work, our recommending policy builds upon the Actor-Critic framework which is appropriate to handle enormous and dynamic action space and reduce redundant computation.

The Actor-Critic framework consists of two components: Actor and Critic. The Actor component inputs the current state s and outputs a deterministic action (or recommending a deterministic list of M points), i.e., $s \rightarrow a = \{a^1, \cdots, a^M\}$.

The Critic component is designed to leverage an approximator to learn an action-value function $Q\,(s, a)$, which determines whether the action a (or a recommendation list of POIs) generated by Actor matches the current state s. The approximation function can be defined as follows:

$$Q(s, a) = \mathbb{E}_{s'}[r+\gamma Q(s', a')|s, a]. \quad (2)$$

Then, according $Q\,(s, a)$, the Actor updates its' parameters in a direction of improving performance to generate proper actions (or recommendations) in the following iterations. In practice, the action-value function is highly nonlinear. Thus we choose deep Q-value function (DQN) (Volodymyr et al. 2015) as the approximator.

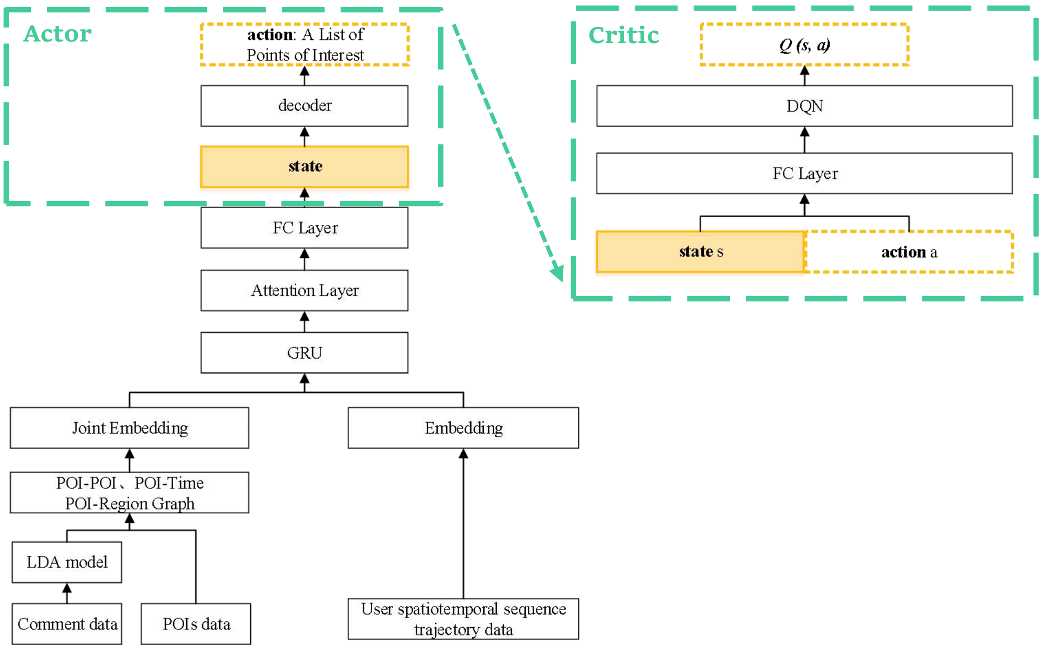

Figure 2. The proposed POI recommendation architecture.

We use Gated Recurrent Units (GRU) (Chung et al. 2014) to capture users' sequential behavior as user's initial preference. The input of GRU is a low-dimensional dense vector representing the information from the users' profile and spatial entities graphs, while the output vector is the representation of users' initial preference. To capture user's dynamic preference, we employ an attention mechanism (Sachdeva et al. 2018), which allows the *RA* to adaptively combine different parts of the input sequence in a linear manner. Given user's current preference (state) s, we aim to recommend a list of POIs to maximize the reward. It is the inverse process of what the joint embedding and GRU layer do. We use the decoder model (Cho et al. 2014) to restore one list from the low-dimensional representation s . The entire embedding and Actor-Critic based recommendation workflow is illustrated in Figure 2.

3. Current State and Future Work

Our experiments will be performed on two real large-scale LBSNs datasets: Foursquare (Chen et al. 2020) and Gowalla (Luo et al. 2019) to evaluate the effectiveness of the proposed framework. We will mainly focus on two questions: (1) how the proposed framework performs compared to the state-of-the-art baselines; and (2) how different components of the proposed model contribute to the performance.

We divide the dataset by sorting users' check-in records according to the check-in time, taking the earlier 80% as the training/validation set and the later 20% as the test set. To evaluate the recommendation performance and explainability, we adopt two widely-used metrics including *Precision@M* and *Recall@M*. The evaluation metrics are defined as follows:

$$Precision@M = \frac{|D_{test} \cap \mathrm{Top\_}M|}{|\mathrm{Top\_}M|} \tag{3}$$

$$Recall@M = \frac{|D_{test} \cap \mathrm{Top\_}M|}{|D_{test}|} \tag{4}$$

where $|D_{test}|$ denotes the test set and $|\mathrm{Top\_}M|$ denotes the list of recommendations of size M generated for the user. In the experiments, we evaluate the performance of the recommendation framework using the mean accuracy and mean recall of all users. Each time the *RA* recommends a list of $M = 5$ POIs to users. The rewards r of one skipped/visited POI is empirically set as 0 and 1 and the discounted factor $\gamma = 0.95$.

We will design experiments to perform parameter sensitivity analysis and explore the impacts of the number of topics, thresholds and the dimensions of the embedding vectors on the recommendation performance of the proposed framework.

References

Mazumdar P, Patra B. K, Babu K S (2020) Cold-start Point-of-interest Recommendation through Crowdsourcing. ACM Transactions on the Web, 14(4), 19:1-19:36.

Hu X, Xu J, Wang W, Li Z, Liu A (2021) A graph embedding based model for fine-grained poi recommendation. Neurocomputing, 428:376-384.

Chen J, Meng X, Ji W, Zhang Y (2020) POI Recommendation Based on Multidimensional Context-Aware Graph Embedding Model. Journal of Software, 31(12):3700-3715 (in Chinese).

Wang P, Liu K, Jiang L, Li X, Fu Y (2020) Incremental Mobile User Profiling: Reinforcement Learning with Spatial Knowledge Graph for Modeling Event Streams. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.853-861.

Xie M, Yin H, Wang H, Xu F, Chen W, Wang S (2016) Learning Graph-based POI Embedding for Location-based Recommendation. CIKM '16: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp.15-24.

Afsar M.M, Crump T, Far B (2021) Reinforcement learning based recommender systems: a survey.

Zhao X, Xia L, Zhang L, Ding Z, Yin D, Tang J (2018) Deep reinforcement learning for page-wise recommendations. Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18), pp.95-103.

Zhao X, Gu C, Zhang H, Yang X, Liu X, Tang J, Liu H (2019) DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems.

Volodymyr M, Koray K, David S, Andrei A. R, Joel V, et al. (2015) Human-level control through deep reinforcement learning. Nature, 518(7540):529-533

Chung J, Gulcehre C, Cho K. H, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp.1724-1734.

Sachdeva N, Cupta K, Pudi V (2018) Attention neural architecture incorporating song feature for music recommendation. Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18), pp.417-421.

Cho K, Merrienboer B. V, Gulcehre C, Ba Hdanau D, Bougares F, Schwenk H, et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. Computer Science.

Luo A, Zhao P, Liu Y, Xu J, Li Z, Zhao L, et al. (2019). Adaptive attention-aware gated recurrent unit for sequential recommendation. Database Systems for Advanced Applications – 24th International Conference, pp.317-322.