Front. Comput. Sci., 2020, 14(2): 314–333
https://doi.org/10.1007/s11704-018-8011-2

RESEARCH ARTICLE

NEXT: a neural network framework for next POI recommendation

Zhiqian ZHANG[1], Chenliang LI (✉)[1], Zhiyong WU[2], Aixin SUN[3], Dengpan YE[1], Xiangyang LUO[4]

1 Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
2 Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong 999077, China
3 School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore
4 State Key Lab of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Abstract The task of *next POI recommendations* has been studied extensively in recent years. However, developing a unified recommendation framework to incorporate multiple factors associated with both POIs and users remains challenging, because of the heterogeneity nature of these information. Further, effective mechanisms to smoothly handle cold-start cases are also a difficult topic. Inspired by the recent success of neural networks in many areas, in this paper, we propose a simple yet effective neural network framework, named NEXT, for next POI recommendations. NEXT is a unified framework to learn the hidden intent regarding user's next move, by incorporating different factors in a unified manner. Specifically, in NEXT, we incorporate meta-data information, e.g., user friendship and textual descriptions of POIs, and two kinds of temporal contexts (i.e., time interval and visit time). To leverage sequential relations and geographical influence, we propose to adopt DeepWalk, a network representation learning technique, to encode such knowledge. We evaluate the effectiveness of NEXT against other state-of-the-art alternatives and neural networks based solutions. Experimental results on three publicly available datasets demonstrate that NEXT significantly outperforms baselines in real-time next POI recommendations. Further experiments show inherent ability of NEXT in handling cold-start.

Keywords POI, neural networks, POI recommendation

Received January 7, 2018; accepted November 9, 2018

E-mail: cllee@whu.edu.cn

1 Introduction

The huge volume of check-in data from various location-based social networks (LBSNs) enables studies on human mobility behavior in a large scale. Next POI recommendations is the task to predict the next POI a user will visit at a specific time point given her historical check-in data. This task has been studied extensively in recent years.

Next POI recommendations is different from typical recommendation tasks (e.g., movies, songs, books) because a wide range of contextual factors are related to the user spatial behaviors. These auxiliary factors include the temporal context, sequential relations, geographical influence, and auxiliary meta-data information (such as textual description, user friendship). However, these factors are heterogeneous in nature. While some relevant aspects are continuous values (e.g., geographical distance, the time interval), others are in the form of discrete values (e.g., friendship, textual words, day of the week). Harnessing useful signals from all these heterogeneous factors to predict user's next move is not an easy task. Existing solutions based on matrix factorization and embedding learning techniques have delivered encouraging per-

formances. These solutions project users and POIs and the associated context factors into a shared hidden space with dense vector representations (i.e., embeddings). The preference score is then calculated directly based on these vectors through the inner product operation.

However, shallow factor/embedding learning is too limited to express the complex knowledge underlying user spatial behaviors with multiple context factors [1]. In existing methods, different context factors are often modeled separately. Then a simple combination is applied to derive the final recommendation score [2–5]. That is, we need to devise an individual model for each context factor. This modeling methodology is complicated and the resultant solution would be inferior, since the different factors carry varying degrees of useful knowledge and their interactions could be much more complex. Some other methods incorporated multiple context factors as additional constraints to guide the learning process [6–8]. For example, check-ins made at a specific time period are grouped together for dynamic feature learning. However, these constraints may not always be useful to match user-POI interactions. A single factor/embedding learning could inevitably incur information loss through a joint optimization of both preserving the constraints and matching user-POI interactions. One plausible solution is to enlarge the dimension number. However, given the sparsity nature of user-POI interaction data, it would easily result in data overfitting. The neural network (NN) with dense vector representation based techniques provide a new way of modeling these factors in a unified manner. This offers two benefits:

- By adding nonlinear transformations on top of the embeddings of users, POIs and their associated factors, we can separate the embedding learning and high-level spatial intent learning to better understand user spatial behavior, leading to a better recommendation accuracy. Specifically, we encode the semantic relatedness or constraints among POIs/users into the corresponding embeddings. For example, users at *Golden Gate Overlook* are likely to visit *Baker Beach* in San Francisco, and vice versa. Therefore, *Golden Gate Overlook* and *Baker Beach* are projected closer in the embedding space. Without the need to match user-POI interactions, the embedding learning would capture the latent features for users, POIs and the associated constraints to its fullness. Also, both new users and POIs may be covered partially by the associated auxiliary meta-data (e.g., textual description, friendship). With dense vector representations, we can easily estimate the spatial intent from the associated meta-data for these cold-start cases.
- Since not all constraint information or latent features are useful for all user-POI interactions, the nonlinear transformation operation is adopted to learn how to extract high-level spatial intent for next POI recommendations. We can also devise factor-based nonlinear extractions to accommodate some specific context factors that are strongly relevant to spatial intent extraction. Through a unified framework with neural treatment and dense representations, the complex interactions among the context factors, users and POIs can be learnt smoothly without handcrafted modelling for each factor alone.

Although the outlook is encouraging, the challenge is how to jointly utilize these context factors effectively in NN. In this paper, we take a special interest on developing a unified neural network based framework to address the above challenge. We propose a simple yet effective neural network framework for next POI recommendation task, named NEXT. With a single layer of feed-forward neural network supercharged by ReLU (i.e., rectified linear unit), NEXT is able to incorporate temporal context, sequential relations, geographical influence and auxiliary meta-data information, in an integrated architecture.

Specifically, NEXT utilizes one-layer of nonlinearity to learn high-level spatial intent for a user from both the user and her latest POI visit information. In other words, NEXT does not calculate an inner product directly on the embeddings of users and POIs in a common hidden space as many existing embedding learning approaches did [3,5,9]. Instead, NEXT utilizes two parallel non-linear transformations to extract the user-based and POI-based spatial intents separately. Empowered by this separation and nonlinearity extraction, we can easily incorporate temporal context, auxiliary meta-data information into the user-based and POI-based intent learning process, in an integrated manner. To further leverage the sequential relations and geographical influence in the context of POI recommendation, we devise a strategy to pre-train POI embeddings. The resultant POI embeddings could encode both the sequential relations and geographical influence.

Based on three real-world datasets, the proposed NEXT achieves significantly better recommendation accuracy than existing state-of-the-art approaches and neural network based alternatives. In summary, the main contributions of this paper are as follows:

- We present a novel neural network based solution for the task of next POI recommendations. The proposed

NEXT is a unified framework such that temporal context, sequential relations, geographical influence and auxiliary meta-data information can be exploited naturally as a single model. By injecting auxiliary meta-data information into the intent learning process, we endow NEXT with the inherent ability to handle cold-start recommendations.

- We adopt the network representation learning technique to pre-train POI embeddings. This pre-training strategy enables us to retain the sequential relations and geographical influence for better model learning. This is a flexible strategy such that other constraints besides these two context factors can also be captured.

The rest of this paper is organized as follows. We start with a literature review about POI recommendation and neural networks in Section 2. In Section 3, we present the proposed framework in detail. In Section 4, we conduct experimental evaluation of the proposed NEXT framework against state-of-the-art alternatives, followed by detailed analysis about NEXT. We conclude this paper in Section 5.

2 Related work

Our work is related to two lines of literatures, POI recommendation and neural networks. We review the recent advances in both areas.

2.1 POI recommendation

The conventional collaborative filtering (CF) techniques have been widely studied for POI recommendation [4, 6, 8]. Ye et al. proposed a friendship-based collaborative filtering (FCF) approach for POI recommendation based on common visited POIs of friends [6]. Temporal context information and geographical constraints were then proven to be effective for POI recommendation [4, 8, 10, 11].

Recently, recommendation models based on matrix factorization and embedding learning have been intensively studied. Cheng et al. proposed a multi-center Gaussian model to capture user geographical influence and combined it with matrix factorization model to recommend POIs [12]. In [2], a tensor-based model called FPMC-LR is proposed by considering first-order Markov chain for POI transitions and distance constraints. Xiong et al. proposed a tensor factorization based framework which takes temporal context into account to derive the latent features in a dynamic manner [7]. Specifically, they take the user-POI visit records within a period of time to construct a particular tensor. The latent factors of users, items and monthes are then learnt based on the corresponding tensors through a Gibbs sampling procedure. However, the temporal context modeled in their work is too coarse (i.e., one month period). Li et al. proposed a ranking based factorization method for POI recommendation which performs factorization by fitting user's preference over POIs, where the preference was measured in terms of POI visit frequency [5]. Feng et al. integrated sequential information, individual preference and geographical influence into a personalized ranking metric embedding model to improve recommendation performance [3]. Liu et al. proposed a general latent factor framework to learn personalized preferences for POI recommendation [13, 14]. It incorporates both the user mobility and geographical influence into a unified factor model. Gao et al. introduced matrix factorization based POI recommendation algorithm with temporal influence based on two temporal properties: non-uniforms and consecutiveness [15]. He et al. proposed a tensor-based latent model which incorporates the date information, geographical distance and personal POI transition patterns into a unified framework [16]. Zhao et al. developed a ranking-based pairwise tensor factorization framework, named STELLAR [17]. STELLAR incorporates fine-grained temporal contexts (i.e., month, weekday/weekend and hour) and brings significant improvement. These works tried to fit the model by maximizing the interaction between users and POIs, where the recommendation decision is made based on the last POI visit alone. Recently, Xie et al. proposed an embedding learning approach that utilizes a bipartite graph to model a pair of context factors in the context of POI recommendation, named GE model [9]. Four pairs of context factors: POI-POI, POI-Region, POI-Time, POI-Word were modeled in a unified optimization framework. Experimental results showed that GE significantly outperforms alternative algorithms for next POI recommendations.

2.2 Neural networks

Neural networks techniques have experienced great success in natural language processing area such as language modeling [18, 19], machine translation [20, 21], question answering [22], summarization [23], etc. Conventional neural networks such as artificial neural network (ANN) [24] and multilayer perceptron (MLP) architectures [24–26] are among the first invented networks. Although relatively simple, it has been proven that a MLP with a single hidden layer containing a sufficient number of nonlinear units can approximate any

continuous function on a compact input domain to arbitrary precision [27]. Recently, several works have been proposed for various recommendation tasks by utilizing deep neural network models [1, 28–31]. Covington et al. introduced a deep MLP network for video recommendation in YouTube platform [28]. In their approach, the heterogenous features (e.g., video categories, user search tokens, video descriptions, users' geographic regions) are represented as individual embeddings. They then combine the concatenation of all related embeddings and hand-crafted user based demographic features as the input to a deep MLP network for candidate ranking. Kim et al. integrated convolutional neural network (CNN) into probabilistic matrix factorization for item recommendation [29]. They utilized an one-layer CNN model to learn the item feature vector based on the associated textual description. Similarly, Zheng et al. [31] proposed a joint neural network model that utilizes CNN model to learn the user and item feature vectors respectively. Then, they introduced factorization machine (FM) [32] as the second layer to derive the final recommendation score. Recently, He et al. developed a deep neural network based matrix factorization approach for collaborative filtering with implicit feedback data [1]. Based on the embeddings of items and users, they applied multiple layers of MLP to extract the high-level hidden features by maximizing user-item interactions.

Among the various neural network structures, recurrent neural networks (RNN) have been widely used to model sequential data of arbitrary length with its recurrent calculation of hidden representation [18,33]. For example, RNN has been successfully adopted in the tasks like poem generation [34] and sequential click prediction [35]. However, RNN suffers from the *exploding or vanishing gradients* problem [36]. That is the distant dependencies within a longer sequence could not be learnt appropriately. Two RNN variants: long short-term memory (LSTM) and gated recurrent unit (GRU), were proposed to tackle this problem to enable long-term dependency learning. LSTM utilizes three gates and a memory cell to control the information flow [36]. It forgets the irrelevant signals by turning off the corresponding three gates and updating memory content. LSTM has been widely used in different tasks involving sequence modeling [37, 38]. GRU is a recent variant of RNN with two gates and no memory cell [20]. The two gates control the expose of the previous hidden output and the update of the new hidden output respectively. GRU has been proven to capture the long-term dependencies just like LSTM [39]. There is very limited studies on using neural network for the task of next POI recommendations. Liu et al. proposed a RNN-based neural network solution by modeling user's historical POI visits in a sequential manner [30], named STRNN. STRNN adopts time-specific transition matrices and distance-specific transition matrices in a recurrent manner under the framework of RNN model. Recently, Manotumruksa et al. proposed a deep recurrent collaborative filtering framework for POI recommendation [40]. Similarly, Feng et al. proposed an attentional recurrent network for mobility prediction from lengthy and sparse trajectories [41]. The proposed NEXT here differs significantly from STRNN in several aspects. First, NEXT is a single-layer feed-forward neural network based model where only the latest POI visit is taken as input. On the contrary, STRNN (and also other RNN variants) has to take all historical POI visits as input and processes in a sequential (or recurrent) manner, which increases the complexity of the model. Second, while STRNN only incorporates temporal context and geographical influence for recommendation, NEXT is able to incorporate multiple context factors (i.e., temporal context, geographical influence, auxiliary meta-data) in a unified framework. Third, instead of applying distance-specific latent feature extraction in STRNN, NEXT encodes sequential relations (transition behaviors and geographical information) within the pre-trained POI embeddings by adopting DeepWalk technique [42]. Our experimental results show that NEXT delivers superior performance than existing matrix factorization and embedding learning based models as well as the neural network based techniques.

3 Our approach

In this section, we first formally define the research problem and then present the proposed **n**eural n**e**twork framework for ne**xt** POI recommendation task, named NEXT. We first introduce the basic neural architecture of NEXT to extract the hidden intent regarding the user's next move. We then describe the mechanism to accommodate NEXT with the temporal context modeling. Next, a pre-training strategy based on the network representation technique (i.e., DeepWalk) is introduced to integrate the sequential relations and geographical influence. We also discuss the traits of NEXT to interpret the hidden intent features and to handle the cold-start issue.

3.1 Next POI recommendations

We first define the problem of next POI recommendations. Given a user with a sequence of historical POI visits $L_i^u = \{q_{t_1}^u, q_{t_2}^u, \ldots, q_{t_{i-1}}^u\}$ up to time t_{i-1}, the task is to calculate a score for each POI based on L_i^u and a time point t_i. Higher

score indicates higher probability that the user will like to visit that POI at time t_i. The POI with the highest score will then be recommended.

In most LBSNs, in addition to the sequence of historical POI visits, users and POIs are associated with auxiliary meta-data information. For example, a user could build connections with other users (e.g., friends) to share their activities and opinions. A POI could contain textual description or category labels. Here, we denote the auxiliary meta-data associated with user u and POI q as $\mathcal{A}_u$ and $\mathcal{A}_q$, respectively.

3.2 Neural architecture

Basic model Different from existing works that directly take the shallow embeddings of users and POIs for score calculation (i.e., an inner product), in NEXT, we introduce an additional feed-forward neural network layer to model user's spatial intent, on top of the embedding.

Let $\mathbf{u}^u \in R^d$ be the embedding of user u, $\mathbf{q}^\ell \in R^d$ be the embedding of a candidate POI q_ℓ to be recommended, and $\mathbf{q}^u_{t_{i-1}} \in R^d$ be the embedding of POI $q^u_{t_{i-1}}$, the last visited POI by user u at time t_{i-1}.[1)] We model the hidden intent of next visit by a nonlinear activation function, *rectified linear unit*: $ReLU(x) = \max(x, 0)$.

$$\mathbf{h}^q_{t_i} = ReLU(\mathbf{W}_1\mathbf{q}^u_{t_{i-1}} + \mathbf{b}_1), \tag{1}$$

$$\mathbf{h}^u = ReLU(\mathbf{W}_2\mathbf{u}^u + \mathbf{b}_2), \tag{2}$$

$$\mathbf{c}^\ell = ReLU(\mathbf{W}_3\mathbf{q}^\ell + \mathbf{b}_3). \tag{3}$$

In the above modeling, the hidden intent vector $\mathbf{h}^q_{t_i}$ is expected to capture semantics regarding user's next move at time t_i based on her last POI visit. Intent vector $\mathbf{h}^u$ captures user specific knowledge on spatial preference of a particular user. $\mathbf{c}^\ell$ is the intent representation of candidate POI ℓ. $\mathbf{W}_1, \mathbf{W}_3 \in R^{d\times d}$ and $\mathbf{W}_2 \in R^{d\times d}$ are transition matrices from POI embeddings and user embeddings respectively, to the hidden intent space. $\mathbf{b}_1, \mathbf{b}_2$ and $\mathbf{b}_3$ are all d-dimensional bias vectors.

With the hidden intent vectors $\mathbf{h}^q_{t_i}$, $\mathbf{h}^u$, and $\mathbf{c}^\ell$, the recommendation score $y_{u,t_i,\ell}$ of POI q_ℓ for user u at time t_i is computed as follows:

$$y_{u,t_i,\ell} = (\mathbf{h}^u + \mathbf{h}^q_{t_i})^{\mathrm{T}}\mathbf{c}^\ell. \tag{4}$$

In simple words, in NEXT, instead of directly using embedding vectors of users and POIs, a feed-forward network layer is used to transform the embeddings to intent vectors. Recommendations are made based on the intent vectors. The transition matrices and bias vectors make it possible to identify the most useful information from the embeddings. By separating the intent vectors and embedding vectors, NEXT framework also makes it simple and straightforward to be extended by incorporating information from different context factors. Figure 1 illustrates the basic model of NEXT.

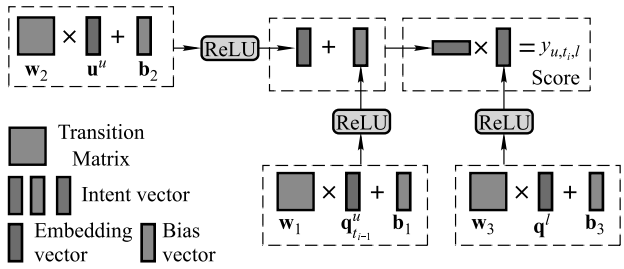

Fig. 1 Basic model of NEXT, where $\mathbf{u}^u$, $\mathbf{q}^\ell$ and $\mathbf{q}^{t_{i-1}}$ are the embedding vectors of the user, candidate poi, and the last visited POI

Incorporating meta-data information Since the associated meta-data information could offer complementary knowledge about users and POIs respectively, it is expected to enhance the understanding of user movement by further considering these auxiliary semantics. For example, a user could hold a similar trajectory and preference over POIs with her friends. Also, a POI could contain textual descriptions or user reviews. These textual information could enable us to better learn the user intent from the last visited POI. Hence, we further enrich NEXT framework by taking these auxiliary meta-data information into the intent calculations. We assume that the meta-data information associated with users and POIs are discrete data. That is, $\mathcal{A}_u$ and $\mathcal{A}_q$ are the sets of meta-data items associated with user u and POI q respectively. First, we calculate the embedding $\mathbf{m}^q$ to represent auxiliary meta-data information of $\mathcal{A}_q$ as follows:

$$\mathbf{m}^q = \frac{1}{|\mathcal{A}_q|}\sum_{m\in\mathcal{A}_q}\mathbf{m}^m, \tag{5}$$

where $\mathbf{m}^m$ is the embedding of item m in meta-data set $\mathcal{A}_q$. When a POI's textual description is available, $\mathcal{A}_q$ works as the set of the words mentioned in the description. In this case, $\mathbf{m}^m$ refers to the embedding of word m. Based on $\mathbf{m}^q$ from Eq. (5), we rewrite Eqs. (1) and (3) as follows:

$$\mathbf{h}^q_{t_i} = ReLU\left(\mathbf{W}_1\left(\alpha\mathbf{q}^u_{t_{i-1}} + (1-\alpha)\mathbf{m}^{q^u_{t_{i-1}}}\right) + \mathbf{b}_1\right), \tag{6}$$

$$\mathbf{c}^\ell = ReLU\left(\mathbf{W}_3\left(\alpha\mathbf{q}^\ell + (1-\alpha)\mathbf{m}^{q_\ell}\right) + \mathbf{b}_3\right), \tag{7}$$

where α works as a tuning parameter, controlling the importance of meta-data information. Similar to Eqs. (5) and (6),

[1)] For model simplicity, we set intent vectors, POI embeddings, user embeddings, word embeddings to be of the same dimension

we rewrite Eq. (2) with auxiliary meta-data set $\mathcal{A}_u$ as follows:

$$\mathbf{m}^u = \frac{1}{|\mathcal{A}_u|} \sum_{m \in \mathcal{A}_u} \mathbf{m}^m, \tag{8}$$

$$\mathbf{h}^u = ReLU(\mathbf{W}_2 (\beta \mathbf{u}^u + (1-\beta)\mathbf{m}^u) + \mathbf{b}_2), \tag{9}$$

where $\mathbf{m}^m$ is the embedding of item m in the meta-data $\mathcal{A}_u$, and β is a tuning parameter like α in Eq. (6). When $\mathcal{A}_u$ is the set of friends of user u, $\mathbf{m}^m$ refers to the embedding of friend m. In this case, $\mathbf{m}^u$ and $\mathbf{u}^u$ are two distinct embedding vectors. Both Eqs. (5) and (8) are taking the average of the embedding vectors and this is a standard approach in neural networks.

Note that the embeddings of users (i.e., $\mathbf{u}^u$) and the embeddings of POIs (i.e., $\mathbf{q}^q$) are not fixed to be within the same hidden space. In this sense, given the types of meta-data information are homogenous for $\mathcal{A}_u$ and $\mathcal{A}_q$, NEXT is flexible to associate two sets of embeddings for the meta-data information. This is reasonable because these two kinds of meta-data may convey very different semantics. For example, both users and POIs can be associated with textual labels. While the users use labels to indicate their tastes and preferred locations, the labels of POIs may cover the related services instead.

3.3 Incorporating temporal context

Temporal context has been widely used in existing POI recommendation studies and proven to be effective. Here, we accommodate NEXT with temporal context by influencing the computation of the hidden intent.

There are two kinds of temporal context available: (i) the time interval between two successive POI visits (i.e., $t_i - t_{i-1}$), and (ii) the particular time point of next POI visit (i.e., t_i). For example, a POI visit happened 12 hours ago could contain less guidance about the user's current spatial intent. Similarly, users could express different spatial intents at different time slots, e.g., lunch hours, or at different days of a week, e.g., weekend. That is, temporal context for next POI recommendations involves both the continuous and discrete information. Here, we design a mechanism to incorporate several kinds of temporal context into the POI based intent calculation (Eq. (6)).

The time interval from the last POI visit is critical to decide the user's next move. It is intuitive that the historical POI visits with different time intervals could provide with varying spatial intents. And the interplay between the intent and time interval could be complicated and subtle. Here, we replace $\mathbf{W}_1$ in Eq. (1) with a time interval t dependent transition matrix $\mathbf{W}_\pi(t)$ as follows:

$$\mathbf{W}_\pi(t) = \begin{cases} \frac{\pi - t}{\pi}\mathbf{W}_0 + \frac{t}{\pi}\mathbf{W}_\pi, & \text{for } t < \pi, \\ \mathbf{W}_\pi, & \text{for } t \geqslant \pi, \end{cases} \tag{10}$$

where $\mathbf{W}_0, \mathbf{W}_\pi \in R^{d \times d}$ are two transition matrices, π is an interval threshold. Equation (10) adopts a linear interpolation between $\mathbf{W}_0$ and $\mathbf{W}_\pi$ to derive the interval dependent transition matrix. When time interval t is close to 0, $\mathbf{W}_0$ is mainly in charge of intent calculation, otherwise, $\mathbf{W}_\pi$ leads the computation when t approaches π. π works as a window, and $\mathbf{W}_\pi$ is only used when the time interval is larger than π.

As to the visit time information, there exist several aspects in discrete forms. We can split a day into 24 time slots, each of which spans one hour (e.g., 17:00 - 18:00). Each time slot is associated with a specific bias vector $\mathbf{b}$. Assigning each time slot with a specific bias vector is reasonable, because users generally express different POI preferences in different time slots [8]. For example, users at the time slots of 20:00 - 22:00 prefer entertainment. The bias vector for each time slot is expected to store such preference information and correct the mistake incurred by considering the last visited POI alone. For example, a user goes from office to a restaurant. If this transition happens in the midnoon, she probably will come back to the office again. However, it is likely for her to go home when this transition takes place during the time period 18:00 - 20:00. Similarly, we can introduce a specific bias vector for each day of the week, or each month. Let $\mathcal{A}_t$ be the aspects associated with visit time t, we calculate the bias vector $\mathbf{b}_t$ for t as follows:

$$\mathbf{b}_t = \sum_{a \in \mathcal{A}_t} \mathbf{b}_a, \tag{11}$$

where $\mathbf{b}_a$ is the bias vector associated with aspect a. Finally, NEXT calculates the hidden intent $\mathbf{h}^q_{t_i}$ as follows:

$$\mathbf{h}^q_{t_i} = ReLU\left(\mathbf{W}_\pi(t_i - t_{i-1})\left(\alpha \mathbf{q}^u_{t_{i-1}} + (1-\alpha)\mathbf{m}^{q^u_{t_{i-1}}}\right) + \mathbf{b}_{t_i}\right). \tag{12}$$

Here, the interval dependent transition in Eq. (10) is similar to the work in STRNN [30]. However, STRNN takes all historical POIs within the interval window for consideration in a recurrent manner, which is computational expensive. Further, STRNN does not consider time-specific bias vector $\mathbf{b}_{t_i}$ (i.e., discrete aspects).

3.4 POI embeddings pre-training

The sequential relations refer to the transition probability that a user visits POI q_b after visiting POI q_a (i.e., $q_a \rightarrow q_b$).

Hence, the transition probabilities convey the general transition patterns, (e.g., from an airport to a hotel). Also, since users like to visit the nearby POIs and their activities are often constrained within a few regions, the visiting behaviors are affected a lot by geographical influence. Sequential relations and geographical influence are validated to be effective for the POI recommendation in many studies [3–5, 43, 44].

In NEXT, we propose a POI embedding pre-training strategy to encode the sequential relations and geographical influence among POIs. Because the non-convexity of the objective function in NEXT, there does not exist a global optimal solution. In such case, current optimization strategy is to find a local optimum. It is widely accepted that a good embedding initialization scheme could result in a faster convergence and superior performance of neural network models [1]. In this sense, POI embedding pre-training can also benefit the model learning.

We adopt DeepWalk [42], a network representation learning technique, to learn the embedding of each POI. DeepWalk builds short sequences of nodes based on random walk over the network structure. Then a neural language model SkipGram [19] is adopted to learn the embeddings of the nodes by maximizing the probability of seeing a node's neighbor in the sequences.

In order to retain these two kinds of information in the latent embedding space, we build a network structure by taking each POI as a distinct node in the network. Specifically, we create the random walk sequences over POIs by using a mixture of both the POI transition patterns and the geographical influence. The random walk transition probability from POI q_i to POI q_j over the network is calculated as follows:

$$p(q_j|q_i) = \rho \frac{\kappa(q_i, q_j)}{\sum_k \kappa(q_i, q_k)} + (1-\rho)\frac{f_{q_i,q_j}}{\sum_k f_{q_i,q_k}}, \tag{13}$$

$$\kappa(q_i, q_j) = 1/(1 + e^{5\frac{d(q_i,q_j)-\bar{d}}{\sigma(d)}}), \tag{14}$$

where $d(q_i, q_j)$ denotes the Euclidean distance between POIs q_i and q_j by using their coordinates, $\bar{d}$ and $\sigma(d)$ are the mean and standard deviation of $d(q_i, q_j)$ respectively, f_{q_i,q_j} is the transition frequency from q_i to q_j in the training dataset.

In Eq. (13), the first term in the right part captures the inherent geographical influence between POIs, while the second term captures the transition behaviors of massive users. Here, the transition behaviors of massive users refer to the frequent POI transition pairs that have been made by a significant number of users. For example, users at *Golden Gate Overlook* are likely to visit *Baker Beach* in San Francisco, and vice versa. ρ is used here to balance the two components. For each POI, we generate τ random walks of length r according to Eq. (13) as in [42]. Then SkipGram language model with hierarchical softmax is applied over these random walk sequences. A POI's embedding is learnt to maximize the probability of seeing its neighbors in the sequences. Based on Eq. (13), the POIs that are close in geographical distance and likely to be visited successively by users will be closer in the embedding space than two random POIs. After finishing the embedding learning by SkipGram, we use the pre-trained POI embeddings as the initialization in model training. In the evaluation part (Section 4), we find that this pre-training strategy delivers better recommendation accuracy.

Furthermore, we use the pre-trained POI embeddings to initialize user embedding $\mathbf{u}^u$. This is reasonable since $\mathbf{u}^u$ is expected to carry the personalized preference for user u. And this preference is strongly relevant to her historical POI visits. We first count the frequency of the POI a user u has visited in the training dataset, and then use the normalized frequency as the weight to initialize user embedding:

$$\mathbf{u}^u = \frac{1}{|\mathbf{L}_u|}\sum_j f_j^u \cdot \mathbf{q}^j, \tag{15}$$

where $|\mathbf{L}_u|$ is the number of POI visits of user u in the training set, f_j^u is the frequency of POI q_j being visited by user u. Although we observe trivial performance improvement by using this initialization strategy, we do obtain the faster convergence for the model training.

3.5 Model discussion

Cold-start The proposed NEXT can inherently handle POI recommendation for both cold-start users and cold-start POIs. In Eq. (4), the final intent calculation is the sum of $\mathbf{h}_{l_i}^q$ and $\mathbf{h}^u$. This additive mechanism has a potential merit for cold-start problems. Given a new user with very few historical visits (i.e., user embedding $\mathbf{u}^u$ is not available), we can directly recommend the POIs based on Eq. (4) by using $\mathbf{h}_{l_i}^q$ alone. Further, with Eq. (9), we can calculate $\mathbf{h}^u$ by using her meta-data information $\mathcal{A}_u$ (i.e., by setting $\beta = 0$). This is particularly helpful for freshers that have no historical visit records. We will investigate the effectiveness of NEXT for cold-start users in Section 4.4. For a cold-start POI q that has not been visited by any user. It is possible to calculate $\mathbf{h}_{l_i}^q$ in Eq. (6) based on its nearby POIs and meta-data information $\mathcal{A}_q$.

Overview Figure 2 summarizes the overall network architecture of NEXT. In comparison with the basic model illustrated in Fig. 1, we jointly utilize the user/POI and their associated meta-data information for hidden spatial intent learning respectively, in a linear interpolation manner

(Eqs. (5)–(9)). The temporal context information is then encoded within a nonlinear calculation of the spatial intent learning (Eqs. (10)–(12)). The sequential relations and geographical influence are retained in the pre-training strategy based on a network representation learning technique (i.e., DeepWalk [42]). Further possible constraints among POIs can be seamlessly integrated into NEXT by using this pre-training process. For example, other existing network representation learning techniques (e.g., TADW [45] and GENE [46]) can be easily adopted in NEXT to encode the POI categorical information or text-based relevance. That is, the proposed NEXT provides a unified framework to incorporate various heterogeneous information, i.e., users, POIs, the auxiliary meta-data information and contextual factors, under the neural network paradigm.

3.6 Training

The parameters of our model are: $\Theta = \{\mathbf{W}_*, \mathbf{M}, \mathbf{B}, \mathbf{U}, \mathbf{Q}, \mathbf{b}_2, \mathbf{b}_3\}$, where $\mathbf{W}_*$ refers to all transition matrices $\mathbf{W}_0$, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{W}_\pi$; and $\mathbf{M}$ contains all item embeddings for the associated meta-data of both users and POIs; $\mathbf{B}$ contains all bias vectors for the discrete aspects associated with the visit times. $\mathbf{U}$ contains all user embeddings, and $\mathbf{Q}$ contains all POI embeddings.

The model training aims to optimize above parameters such that each POI visit in the sequence of a user's POI visits in the training set can be predicted successfully. We adopt a softmax function to calculate the predicted POI probability vector $\mathbf{p}_{t_i}^u$ for user u at time t_i:

$$\mathbf{p}_{t_i}^u(k) = \frac{e^{y_{u,t_i,k}}}{\sum_j e^{y_{u,t_i,j}}}. \tag{16}$$

Then, we use the cross-entropy error between the ground truth POI distribution (i.e., in a one-hot form) and predicted POI distribution by Eq. (16) as the cost objective:

$$J = \frac{1}{U}\sum_u^U \sum_{t\in\mathbf{L}_u} \sum_k^Q \hat{\mathbf{q}}_t^u(k) \cdot \log \mathbf{p}_t^u(k) + \lambda\|\Theta\|_2, \tag{17}$$

where $\mathbf{L}_u$ is the set of historical POI visits in the training set for user u, Q is the number of all POIs under consideration, $\hat{\mathbf{q}}_t^u$ is the ground truth POI distribution at time t with *1-of-Q* coding scheme, λ controls the importance of the regularization term, and U is the number of users under consideration.

To minimize the objective, we use stochastic gradient descent (SGD) and back propagation to update the parameters. Although POI embeddings $\mathbf{Q}$ is pre-trained based on the sequential relations and geographical influence, the embeddings are fine tuned based on the cost objective. Several tunable parameters are used to control the impact of the corresponding contextual factors (e.g., α, β, π, ρ). Here, we choose to optimize these parameters in an incremental manner based on the validation set, similar to incremental testing. For example, starting with the basic model illustrated in Fig. 1 (Eqs. (1)–(4)), we add the auxiliary meta-data associated with users and choose the optimal α value. After α is fixed, we add the auxiliary meta-data associated with POIs and choose the optimal β value. Then β is fixed. This incremental

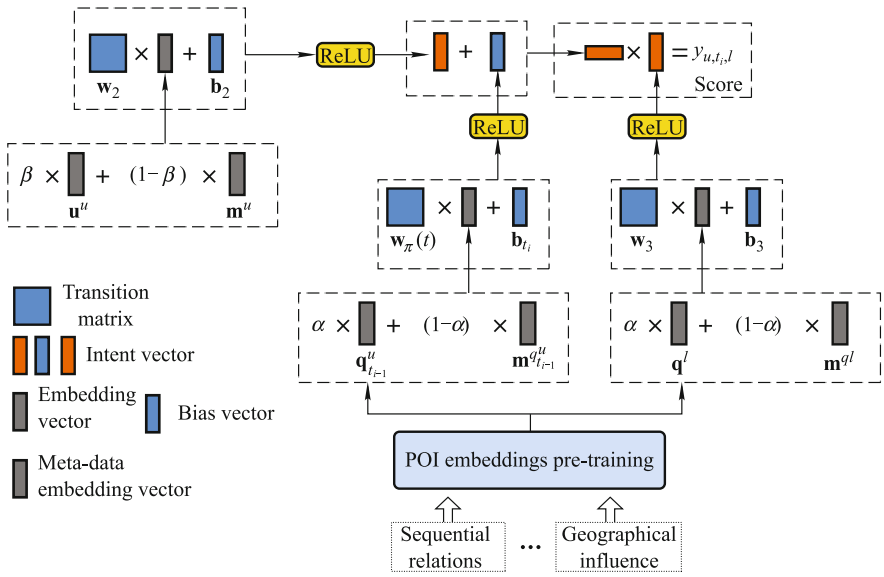

Fig. 2 Overall architecture of NEXT, $\mathbf{m}^u$, $\mathbf{m}^\ell$ and $\mathbf{m}^{q_{t_{i-1}}}$ are the embedding vectors of the meta-data associated with the user, candidate POI and the last visited POI respectively

optimization process continues until all the parameters are optimized.

4 Experiments

In this section, we first conduct experiments to evaluate the proposed NEXT against the state-of-the-art alternatives over three real-world datasets in Section 4.3. Then, we evaluate the performance of NEXT in the scenario of recommendation for cold-start users in Section 4.4. A detailed analysis about NEXT is provided in Section 4.5. The experimental results show that our proposed framework delivers promising performance for next POI recommendations, including cold-start users. At last, we discuss the architecture settings with a deeper network structure for NEXT in Section 4.6.

4.1 Datasets

Foursquare Singapore (SIN) dataset is a collection of 194, 108 check-ins made within Singapore from 2, 321 users at 5, 596 POIs between August 2010 and July 2011 in Foursquare [8]. This dataset has previously been used in other studies [3, 5, 8].

Gowalla dataset contains 736, 148 check-ins made within California and Nevada between February 2009 and October 2010 in Gowalla [43]. The *Gowalla* dataset has previously been used in [3, 5, 8, 30].

CA dataset is a collection of 483, 813 check-ins made in Foursquare by 4, 163 users living in California. Each distinct POI is provided with a text description indicating its content. There are total 50 distinct words in all descriptions. Moreover, each user is connected to a number of other users (i.e., friendship). This dataset has previously been used in [11]. Note that, this is the only dataset that contains auxiliary metadata for both users and POIs, and the date information (i.e., day of the week) for each POI check-in record.

In all three datasets, each check-in is associated with a timestamp indicating when the user made this check-in and a latitude-longitude coordinate pair indicating its physical location. Following the work of PRME-G in [3], we remove the less frequent users and POIs from each dataset, such that each user has at least 10 check-ins, and each POI has been visited by at least 10 users. The data statistics on these three datasets after preprocessing is reported in Table 1. In *CA* dataset, there are on average 2.67 descriptive words for a POI and 4.36 friends for a user. We also calculate the distance between two random POIs for each dataset based on the Haversine formula[2)]. The average distance for *SIN*, *Gowalla* and *CA* datasets is 6.9KM, 306.5KM and 795.5KM respectively. We can see that the distance between two POIs is positively correlated with the size of the area covered by the dataset. For performance evaluation, we use the last 20% POI visits of each user as test set, the earliest 70% POI visits as training set, and the remaining 10% data as validation set to tune parameters.

Table 1 Statistics on the three datasets

Dataset	#User	#POI	#Check-in	#AvgC	Meta-data	
					#Avg($\mathcal{A}_u$)	#Avg($\mathcal{A}_q$)
SIN	1,918	2,678	155,514	81.08	-	-
Gowalla	5,073	7,020	252,945	49.86	-	-
CA	2,031	3,112	105,836	52.1	4.36	2.67

Note: #User: the number of users; #POI: the total number of POIs; #Check-in: the total number of check-ins; #AvgC: average number of check-ins per user; #Avg($\mathcal{A}_u$): average number of items in $\mathcal{A}_u$; #Avg($\mathcal{A}_q$): average number of items in $\mathcal{A}_q$

4.2 Experimental setup

Methods and parameter settings We compare our model against the following recent state-of-the-art POI recommendation approaches.

- **PMF** is a method based on conventional probabilistic matrix factorization over the user-POI matrix [47].
- **WMF** is a method with implicit feedbacks based on the weighted matrix factorization [48]. The user-POI interactions are weighted to reflect the hidden perference.
- **BPTF** incorporates temporal information into a tensor factorization algorithm [7]. It also utilizes a Bayesian treatment to automatically tune the parameters. We use the released code and recommended settings by the authors.
- **FMFMGM** fuses matrix factorization with geographical and social influence for POI recommendation [12]. The geographical influence is modeled by a multi-center Gaussian model (MGM). The recommendation score is calculated as the product of MGM and MF parts. The optimal parameters are tuned on the validation set.
- **GeoPFM** [14] is an extended version of GT-BNMF [13]. It is a general geographical probabilistic framework which takes personal preferences, geographical factors and user mobility behaviors into a unified factor model. The optimal parameters are tuned on the valida-

[2)] see Wikipedia

tion set.

- **PRME-G** embeds user and POI into the same latent space to capture the user transition patterns [3]. The geographical and temporal influence are incorporated in PRME-G through a simple weighting scheme. We use the recommended settings with 60 dimensions and $\pi = 6h$ as in their paper.
- **Rank-GeoFM** is a ranking based geographical factorization approach [5]. Rank-GeoFM learns the embeddings of users, POIs by fitting the user's POI frequency. Both temporal context and geographical influence are incorporated in a weighting scheme. We use the recommended settings with $K = 100, k = 300$ as in their paper and fine-tune the parameters α and β on the validation set.
- **Graph based embedding (GE)** jointly learns the embeddings of POIs, regions, time slots, and auxiliary meta-data (i.e., descriptive words of POIs) in one common hidden space [9]. The recommendation score is then calculated by a linear combination of the inner products for these contextual factors. We tune hyperparameters N and $\triangle T$ on the validation set.
- **Neural matrix factorization (NeuMF)** is a recent state-of-the-art deep neural network based algorithm over implicit feedback [1]. NeuMF combines both generalized matrix factorization and MLP under one framework to learn latent features. Like PMF, we apply NeuMF over the user-POI matrix for the recommendation. The best performance is reported by tuning hyperparameters.
- **STRNN** is a RNN-based model for next POI recommendations [30]. It incorporates both the temporal context and geographical information within recurrent architecture.
- **RNN** is a standard RNN model for sequence modeling, upon which the above STRNN model was built [18]. In the context of POI recommendation, the hidden feature vector $\mathbf{h}^u_{t_i}$ of user u at time t_i is calculated recurrently based on the whole historical POI visits:

$$\mathbf{h}^u_{t_i} = \sigma(\mathbf{W_4}\mathbf{q}^u_{t_{i-1}} + \mathbf{C}\mathbf{h}^u_{t_{i-1}}), \tag{18}$$

where $\mathbf{W_4}$ is the transition matrix from the input embedding to the hidden state, $\mathbf{C}$ is the state-to-state recurrent weight matrix, σ is chosen to be the sigmoid function. Following the work in [30], we calculate the recommendation score $y_{u,t_i,\ell}$ of POI ℓ for user u at time t_i as follows:

$$y_{u,t_i,\ell} = (\mathbf{h}^u_{t_i} + \mathbf{u}^u)^{\mathrm{T}}\mathbf{q}^\ell. \tag{19}$$

- **LSTM** is a variant of RNN model which contains a memory cell and three multiplicative gates to allow long-term dependency learning [36]. We calculate the recommendation score by using Eq. (19).
- **GRU** is a variant of RNN model which is equipped with two gates to control the information flow [20]. We calculate the recommendation score by using Eq. (19).

Other reported alternatives are empirically found to be inferior to STRNN, PRME-G, and Rank-GeoFM, in their works respectively[3)]. Hence, we do not repeat the comparison. Note that, the proposed TRM model in [11] can be evaluated based on *CA* dataset. However, due to the shortness of POI description and the smaller number of POIs after preprocessing, TRM only achieves a slightly better performance than PMF. Therefore, we exclude TRM from further comparison. The first eight baseline methods listed above are conventional factorization or embedding learning based techniques. The next five baseline are neural networks based methods, which apply the nonlinearity for high-level transformation. Note that GRU and LSTM have not been evaluated in previous work on the task of next POI recommendations. Moreover, we need to highlight that the task of next POI recommendations is different from general POI recommendation problems. As defined in Section 3.1, we need to predict the POI that a user will visit at a specific time point for next POI recommendations. In this sense, at a specific time point, there is only one POI that a user will visit. On the contrary, the task of general POI recommendations is to predict the POIs that a user will visit in the future. Among the methods in comparison, PMF, WMF, FMFMGM, GeoPFM and NeuMF are the methods for general POI recommendations. Here, we evaluate these general POI recommendation methods as a performance reference.

Metrics Following the existing works [9, 16, 30], two standard metrics are used for performance evaluation: Acc@K and Mean Average Precision (MAP). For a specific test instance (i.e., a user visited a POI in the test set), Acc@K is 1 if the visited POI appears in the top-K ranking; otherwise 0 is taken. The overall Acc@K is the average value over all test instances. Here, we choose to report Acc@K with $K = \{1, 5, 10\}$. MAP is widely used to evaluate the quality

[3)] Some recent works (e.g., [16, 17]) that incorporate POI categories, are excluded for comparison, because our datasets do not contain these meta-data

of ranking. The higher the ground truth POI is ranked, the larger the MAP value, which indicates a better recommendation accuracy.

Hyperparameters and training The interval threshold π in Eq. (10) is empirically set to be 6/6/72 hours for *SIN*, *Gowalla* and *CA* datasets respectively. The dimensionality for the embeddings and the hidden intent are fixed to be 60 for neural network based methods for fair comparison (i.e., $d = 60$ in NEXT). The regularization parameter λ is 0.01 and the learning rate γ is 0.005. As to incorporating auxiliary meta-data information, we set $\alpha = 0.3, \beta = 0.2$ in NEXT. We apply the early stop based on the validation set, or a maximum of 50 epochs are run for neural network based methods.

As to POI embeddings pre-training, we set $\tau = 50$ and $r = 20$ as in the original work of DeepWalk [42]. In Eq. (13), $\rho = 0$ is used in generating random walks for performance comparison. The impact of ρ will be studied in Section 4.5.

4.3 Performance comparison

For performance comparison, we report the recommendation accuracy of different methods over the three datasets in Table 2, where significance test is by Wilcoxon signed-rank test. We make the following observations:

First, the proposed NEXT model performs significantly better than all existing state-of-the-art baselines evaluated here on the three datasets on all the metrics. NEXT outperforms the conventional matrix/tensor factorization method PMF, BPTF, FMFMGM and GeoPFM significantly by a large margin. Specifically, NEXT outperforms the four methods by around 468.1%–5581.5%, 541.2%–8486.9%, and 243.4%–5629.0% in terms of MAP metric on *SIN*, *Gowalla* and *CA* datasets respectively. As to the three embedding learning based solutions (i.e., PRME-G, Rank-GeoFM, GE), NEXT outperforms them by around 62.0%–552.5%, 46.5%–602.8% and 50.9%–67.0% in terms of MAP metric on *SIN*, *Gowalla* and *CA* datasets respectively. Note that both PRME-G and Rank-GeoFM incorporate information from temporal context and geographical influence within their models on *SIN* and *Gowalla*. The large improvement suggests that high-level intent features extracted through a nonlinearity in NEXT better catch user's spatial behaviors. Moreover, NEXT consistently outperforms four RNN-based methods: RNN, LSTM, GRU, and STRNN. The performance gain provided by NEXT over these four counterparts is about 22.1%–48.6% and 35.8%–83.0% in terms of MAP metric on *SIN* and *Gowalla* respectively. Note that no temporal context information can be incorporated by the vanilla RNN, LSTM and GRU models. This indicates that the mechanism to absorb two kinds of temporal context in NEXT is effective for the task of next POI recommendations.

Second, both PMF and BPTF obtain much worse performance on three datasets in all metrics, because the user-POI matrix is very sparse on these datasets, and no fine-grained temporal context or geographical influence is leveraged at all. On the other hand, both FMFMGM and GeoPFM perform better than PMF and BPTF, which indicates the significance of geographical influence in POI recommendations. Similar results are observed on NeuMF, a neural network based collaborative filtering technique based on implicit feedback information. Since both PRME-G and Rank-GeoFM utilize ranking based optimization strategy, the data sparsity issue

Table 2 Performance comparison over three datasets by Acc@K and MAP

Method	SIN				Gowalla				CA			
	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP
PMF	0.0013†	0.0311†	0.0731†	0.0235†	0.0002†	0.0149†	0.0418†	0.0125†	0.0006†	0.0050†	0.0109†	0.0106†
WMF	0.0374†	0.1366†	0.2331†	0.0985†	0.0355†	0.1281†	0.1956†	0.0888†	0.0532†	0.1605†	0.2262†	0.1104†
BPTF	0.0055†	0.0129†	0.0196†	0.0038†	0.0018†	0.0048†	0.0083†	0.0023†	0.0035†	0.0061†	0.0077†	0.0031†
FMFMGM	0.0114†	0.0501†	0.0843†	0.038†	0.0158†	0.0401†	0.0511†	0.0308†	0.0270†	0.0676†	0.0905†	0.0516†
GeoPFM	0.0102†	0.0364†	0.0693†	0.0325†	0.0158†	0.0314†	0.0508†	0.027†	0.0213†	0.0612†	0.0669†	0.0413†
PRME-G	0.0751†	0.1156†	0.1357†	0.0991†	0.1088†	0.1600†	0.1783†	0.1348†	0.0888†	0.1287†	0.1520†	0.1130†
Rank-GeoFM	0.0705†	0.1870†	0.2575†	0.1313†	0.0488†	0.1428†	0.1997†	0.1000†	0.0540†	0.1505†	0.2085†	0.1061†
GE	0.0123†	0.0486†	0.0735†	0.0326†	0.0100†	0.0158†	0.0488†	0.0281†	0.0894†	0.1402†	0.1651†	0.1174†
NeuMF	0.025†	0.0854†	0.1341†	0.0654†	0.0230†	0.0682†	0.1082†	0.0549†	0.0437†	0.0944†	0.1361†	0.0781†
RNN	0.1063†	0.2397†	0.3072†	0.1742†	0.084†	0.1859†	0.2364†	0.1376†	0.0865†	0.1877†	0.2370†	0.1397†
LSTM	0.1032†	0.2344†	0.3015†	0.1701†	0.0868†	0.1979†	0.2535†	0.1443†	0.0931†	0.2028†	0.2583†	0.1511†
GRU	0.0999†	0.2211†	0.2864†	0.1626†	0.0838†	0.2015†	0.2644†	0.1454†	0.0924†	0.1974†	0.2505†	0.1482†
STRNN	0.0826†	0.1948†	0.2636†	0.1431†	0.0557†	0.1539†	0.2081†	0.1079†	0.0713†	0.1637†	0.2181†	0.1221†
NEXT	**0.1405**	**0.2917**	**0.3649**	**0.2159**	**0.1282**	**0.2644**	**0.3339**	**0.1975**	**0.1134**	**0.2403**	**0.3097**	**0.1789**

Note: The best results are highlighted in boldface on each dataset. † indicates that the difference to the best result is statistically significant at 0.05 level

is alleviated by making use of unobserved data to learn the parameters. Moreover, temporal information and geographical influence are incorporated in these two methods. Therefore, a large performance improvement is obtained by PRME-G and Rank-GeoFM over PMF, BPTF, FMFMGM, GeoPFM and NeuMF. The same phenomenon was also observed in the related works [3, 5, 30]. It is interesting to note that WMF obtains superior performance against the conventional matrix factorization methods like PMF and BPTF across the three datasets. The results validate that explicitly discriminating the user-POI interactions could enable a better understanding of the users' spatial behaviors.

Third, NeuMF significantly outperforms conventional latent factor based methods. This suggests the superiority of nonlinearity for extracting hidden high-level features. As being an embedding learning technique, GE performs much worse than PRME-G and Rank-GeoFM on both *SIN* and *Gowalla* datasets. The probable reason is that no region information is available on these two datasets. The region information works as the geographical influence for GE model. The region information is provided in *CA* dataset, and we observe that GE achieves very close performance to PRME-G and Rank-GeoFM.

Fourth, the three RNN-based methods (i.e., RNN, LSTM, GRU) perform much better than PMF, BPTF, GeoPFM, FMFMGM, PRME-G and Rank-GeoFM on most metrics. This is consistent with our above discussion that non-linear transformation operation as provided by the neural network models enables better high-level spatial intent learning. Although LSTM and GRU were designed to alleviate the *exploding or vanishing gradients* problem, no superiority is observed for them over the vanilla RNN model on the *SIN* dataset. Reported in Table 1, the users in *SIN* have more POI visits on average. Because RNN-based models accumulate all historical information in the last hidden feature vector [22], the longer POI sequence could introduce much irrelevant information that hurts the performance. This result indicates that the visiting behaviors performed a long time ago are irrelevant to next POI recommendations. We observe that STRNN only achieves similar performance with Rank-GeoFM and PRME-G. STRNN even performs much worse than RNN, LSTM and GRU. Desipte our best efforts (including contacting the authors), due to various restrictions we could not get the source code of STRNN. Therefore, we implement STRNN based on the original paper [30]. Also, the datasets used here are completely different ones. Note that *Gowalla* used in our work excludes inactive POIs and users (see Section 4.1). However, no filtering was applied in the work of STRNN [30]. We argue that these factors could contribute to the inconsistent performance, compared with the ones reported in the original paper.

As described in Section 4.2, two main categories of baseline methods are evaluated here for performance comparison: 1) factorization or embedding learning based methods, i.e., PMF, WMF, BPTF, FMFMGM, PRME-G, Rank-GeoFM, and GE; 2) neural network based methods, i.e., NeuMF, RNN, LSTM, GRU, STRNN, and NEXT. These neural network based methods can be further classified into two subcategories: 1) sequence modeling based methods by taking higher-order sequential relations into account, i.e., RNN, LSTM, GRU, and STRNN; 2) pointwise based methods by taking each interaction or latest visit record into account, i.e., NeuMF and NEXT. That is, the underlying methodologies of these methods are quite different.

Although the proposed NEXT delivers superior recommendation accuracy over other existing alternatives significantly, it is interesting to examine to what extend NEXT has complemented the weakness of the methods with a different methodology. Hence, we conduct a detailed performance analysis by comparing NEXT with two representative methods of different methodologies: PRME-G and LSTM. As mentioned in Section 4.2, PRME-G is an embedding learning based method by incorporating both temporal context and geographical influence for next POI recommendations. And LSTM is a recurrent neural network based method that models the historical POI visits as a sequential sequence. We like to check the number of test instances that NEXT performs correctly on, but the existing methods fail on, and vice versa.

Table 3 lists the performance comparison of NEXT against PRME-G and LSTM in three distinct cases on metric Acc@K with $K = \{1, 5, 10\}$ across three datasets. Symbol ► refers to the number of test instances that NEXT successfully recommends in terms of Acc@K, but PRME-G/LSTM fails on; ♦ refers to the number of test instances that both NEXT and PRME-G/LSTM success in; and ◄ refers to the number of test instances that PRME-G/LSTM successes in but NEXT fails on. We make the following observations:

First, all the instance numbers for ►, ♦ and ◄ cases increase as K becomes larger in both comparisons. Specifically, the increasement for ► case is much larger than that of ♦ and ◄ cases when NEXT is compared against PRME-G. We also observe that the corresponding instance number for ► is always larger than that of ♦ case with different K values across the datasets. Similar preformance comparison is also observed by comparing NEXT against Rank-GeoFM (results not shown). This indicates that the nonlinearity operation

Table 3 Performance comparison between NEXT and PRME-G/LSTM on three datasets

Methods	Acc@K	SIN			Gowalla			CA		
		►	♦	◄	►	♦	◄	►	♦	◄
NEXT - PRME-G	1	3283	1244	1199	4695	2125	3660	1635	785	673
	5	6499	2780	1118	9097	4797	3846	3526	1613	673
	10	8097	3493	1101	11753	5970	3660	4473	2017	729
NEXT - LSTM	1	2136	2391	885	3725	3095	1474	908	1512	535
	5	2956	6323	1301	5908	7986	2429	1554	3585	874
	10	3240	8350	1495	7153	10570	2772	1867	4623	1057

utilized by NEXT is more appropriate to capture the complex interactions between users and POIs. On the contrary, the increasement for ► case is much smaller than ♦ case when NEXT is compared against LSTM. Moreover, the instance number for ♦ case by LSTM is much larger than that of PRME-G in all K settings and datasets. Note that neither temporal or geographical information is incorporated in LSTM. One possible reason for this observation is that LSTM also utilizes a nonlinearity to extract the hidden spatial intent for each user, leading to its superiority over PRME-G.

Second, both PRME-G and LSTM perform better than NEXT in varying number of test instances across the three datasets. In terms of Acc@10, PRME-G performs better than NEXT in 1,101, 3,660 and 729 test instances over the three datasets respectively. These numbers for LSTM are 1,495, 2,772 and 1,057. This observation sheds light on the potential direction of further enhancement for better next POI recommendations. That is, the methods with different underlying methodologies complement each other to some extent. Hence, we plan to explore the possible fusion strategy to combine the merits of the different methodologies in our future work.

In summary, the experimental results show that the proposed NEXT can successfully learn user's spatial intent, leading to superior performance of next POI recommendations.

4.4 Experiments on cold-start

Here, we evaluate the performance of NEXT and other competitors for cold-start users. Specifically, since each dataset is preprocessed to retain only active users and POIs (see Section 4.1), we therefore take 200 inactive users that were excluded from the training for evaluation. We conduct the experiments on *CA* dataset, because it is the only dataset containing auxiliary meta-data information.

For each cold-start user u, we randomly pick a POI transition record (q_i, q_j) such that the user visited q_j after her latest visit at q_i. For evaluation purpose, we restrict to the record of both q_i and q_j being included in the training set. Here, we test to recommend q_j by utilizing both her latest POI visit and meta-data. Among the baseline methods, only PRME-G, STRNN, RNN, LSTM and GRU can be adapted here by utilizing only the POI information. STRNN, RNN, LSTM and GRU are all RNN-based models. Since LSTM achieves the best performance on *CA* dataset among these RNN variants (see Table 2), we choose LSTM as the baseline method, and report its performance for cold-start user recommendation. Other variants are found to be inferior than LSTM for this experiment. Table 4 reports the performance of different methods. We observe that NEXT outperforms PRME-G and LSTM in most metrics. This confirms that incorporating meta-data information is positive for addressing the recommendation for cold-start users. Based on Eqs. (5)–(9), different kinds of auxiliary meta-data can be incorporated by using dense vector representations. That is, we can exploit the auxiliary meta-data information in NEXT to smoothly derive user intent in a unified way.

Table 4 Performance comparison for cold-start users

Method	Acc@1	Acc@5	Acc@10	MAP
PRME-G	0.0550	0.0650	0.0800	0.0631
LSTM	0.0300	0.1200	**0.1900**	0.0765
NEXT	**0.0600**	**0.1400**	0.1850	**0.1045**

Note: The best results are highlighted in boldface

4.5 Analysis of NEXT

We now investigate the impact of different parameter settings in NEXT. Note that when studying a specific parameter, we set the other parameters to the values used in Section 4.2. Here, we choose to report the performance of NEXT under different settings on the test set directly. The similar performance patterns are also observed on the validation set.

Temporal context We first investigate the effect of the two kinds of temporal contexts in NEXT. Table 5 lists the performance comparison over three datasets, where ✓ refers to the model with the corresponding temporal information. Note that only *CA* dataset can provide with the date information for

each POI visit (i.e., day of the week). Observe that incorporating either time interval or visit time information leads to better performance. More performance gain is obtained by introducing the time interval dependent transition, compared to using visit time specific aspects alone. This validates that the time interval since the latest POI visit plays a critical role in learning spatial intent from historical spatial behavior. Further improvement is obtained by incorporating both time interval and visit time information together. This indicates that these two kinds of temporal context provide complementary benefits for next POI recommendations. We also observe that marginal improvement is obtained by incorporating the day of the week information. This is reasonable since the scale of a day is too big to hold fine-grained spatial preference.

Table 5 Impact of the temporal contexts

Dataset	TI	TS	DW	Acc@1	Acc@5	Acc@10	MAP
SIN	-	-	-	0.1161	0.2576	0.3250	0.1869
	✓	-	-	0.1322	0.2833	0.3569	0.2077
	-	✓	-	0.1272	0.2690	0.3414	0.1986
	✓	✓	-	**0.1405**	**0.2917**	**0.3649**	**0.2159**
Gowalla	-	-	-	0.0986	0.2254	0.2861	0.1630
	✓	-	-	0.1172	0.2535	0.3250	0.1868
	-	✓	-	0.1058	0.2310	0.2919	0.1691
	✓	✓	-	**0.1282**	**0.2644**	**0.3339**	**0.1975**
CA	-	-	-	0.0942	0.2104	0.2661	0.1553
	✓	-	-	0.0994	0.2185	0.2782	0.1607
	-	✓	-	0.1058	0.2281	0.2898	0.1691
	-	-	✓	0.0978	0.2146	02715	0.1558
	✓	✓	-	0.1115	0.2396	0.3038	0.1772
	✓	✓	✓	**0.1134**	**0.2403**	**0.3097**	**0.1789**

Note: TI: time interval; TS: time slot; DW: day of the week. The best results are highlighted in boldface on each dataset

We further study the impact of π value in NEXT. Recall in Eq. (10), a larger π indicates that the user's spatial intent changes temporally slower, while a smaller π indicates that the user's spatial intent is mainly determined by the nearby movement. Table 6 reports the performance of different π values over the three datasets. The optimal π values are $12h$, $6h$ and $72h$ on *SIN*, *Gowalla* and *CA* datasets respectively. This finding is consistent with the statistic properties of the time interval between two successive POI visits on the three datasets. The median time intervals are $19.3h$, $13.1h$ and $47.3h$ on *SIN*, *Gowalla* and *CA* datasets respectively. These numbers correlate well with the optimal π values for the three datasets. We note that the performance starts to degrade on *SIN* and *Gowalla* datasets when π is larger than $72h$. However, little performance fluctuation is observed for a wide range of π values across the three datasets. Based on the results, we set π to be 6/6/72 hours for *SIN*, *Gowalla* and *CA* datasets respectively in our experiments.

Table 6 Effect of different π values

Dataset	π	Acc@1	Acc@5	Acc@10	MAP
SIN	6h	0.1405	0.2917	0.3649	0.2159
	12h	**0.1418**	0.2912	**0.3658**	**0.2168**
	24h	0.1396	**0.2929**	0.3657	0.2155
	48h	0.1364	0.2873	0.3585	0.2113
	72h	0.1321	0.2798	0.3548	0.2068
Gowalla	6h	**0.1282**	**0.2644**	**0.3339**	**0.1975**
	12h	0.1200	0.2542	0.3251	0.1890
	24h	0.1244	0.2574	0.3250	0.1924
	48h	0.1211	0.2520	0.3204	0.1880
	72h	0.1155	0.2483	0.3153	0.1829
CA	24h	0.1112	0.2354	0.2980	0.1750
	48h	0.1094	0.2343	0.2979	0.1738
	72h	**0.1134**	**0.2403**	**0.3097**	**0.1789**
	96h	0.1097	0.2338	0.2965	0.1738
	120h	0.1100	0.2337	0.2976	0.1744

Note: The best results are highlighted in boldface on each dataset

Number of dimensions We study the effect of the number of dimensions of hidden vectors and POI embeddings. Here, we vary the number of dimensions from 10 to 100. Figure 3 shows the MAP and Acc@10 values for varying dimension numbers on the three datasets. NEXT achieves stable performance in the range of [50, 100]. We observe that NEXT outperforms RNN, LSTM and GRU even when the number of dimensions is as small as 20. The results further confirm the superiority of the proposed NEXT for next POI recommendations.

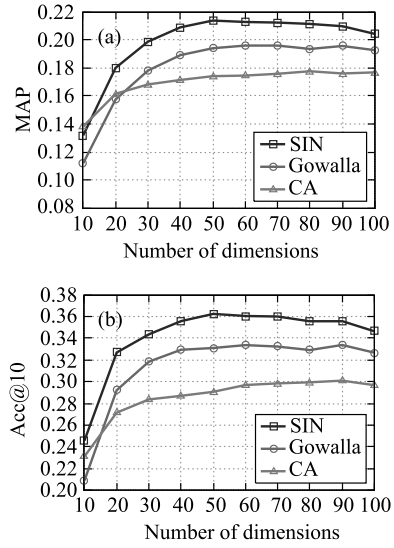

Fig. 3 Effect of the number of dimensions in NEXT. (a) MAP; (b) Acc@10

Auxiliary meta-data We further study the impact of incorporating auxiliary meta-data information to the recommendation accuracy in NEXT. Table 7 reports the performance

with/without incorporating the associated friendship and textual description on *CA*. We observe that NEXT achieves significant better performance by incorporating auxiliary meta-data information. Following the parameter tuning strategy mentioned in Section 3.6, we set $\alpha = 0.3$ and $\beta = 0.2$ for *CA* dataset. Figure 4 plots the performance of NEXT by varying β and α values after fixing $\alpha = 0.3$ and $\beta = 0.2$ respectively. Observe that the performance of NEXT starts decrease as either α or β increases towards 1. The optimal range of β is $[0.1, 0.3]$. Also, the optimal range of α is $[0.3, 0.6]$. We argue that the meta-data information associated with the users could be more useful on *CA* dataset. Overall, the experimental results demonstrate that the proposed NEXT is competent to exploit the auxiliary meta-data for better recommendation accuracy.

Table 7 Impact of incorporating auxiliary meta-data in NEXT

Meta-data	Acc@1	Acc@5	Acc@10	MAP
-	0.1007	0.2173	0.2793	0.1615
✓	**0.1134**	**0.2403**	**0.3097**	**0.1789**

Note: The best results are highlighted in boldface on each dataset

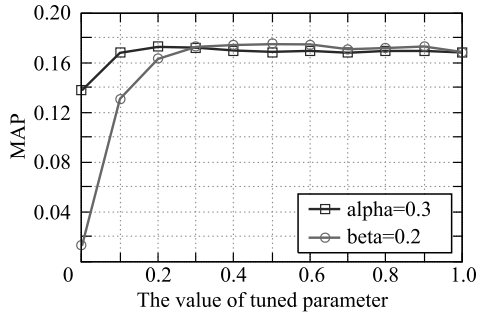

Fig. 4 Performance of NEXT with different β and α values by fixing $\alpha = 0.3$ and $\beta = 0.2$ respectively

Though the optimal values for α and β can be determined based on the validation set, it is interestng to derive a network to adjust these two values automatically based on the representations of the users and meta-data information. The calculation of α and β can be implemented through an attention mechanism[4]:

$$\alpha = \sigma(\mathbf{v}_1 \mathbf{q}^u_{l_{i-1}} + \mathbf{v}_2 \mathbf{m}^{q^u_{l_{i-1}}}), \tag{20}$$

$$\beta = \sigma(\mathbf{v}_3 \mathbf{u}^u + \mathbf{v}_4 \mathbf{m}^u), \tag{21}$$

where $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$ are the attentive parameters, and σ is the sigmoid function. With this attention mechanism, we allow the model to automatically balance the influence of different meta-data information. The experiments on *CA* dataset suggest that the attention mechanism achieves a MAP of 0.1656. Also, inferior performance is also observed in terms of Acc@1, Acc@5 and Acc@10 respectively (0.1040, 0.2233 and 0.2866). It suggests that setting α and β values based on a validation set is still a better option when the training instances are adequate. However, since the performance deterioration is just minor, the attention mechanism would be a desired solution when the training instances are relatively scarce.

POI embeddings pre-training DeepWalk is used to generate POI sequences in NEXT to encode the sequential relations and geographical influence among POIs. Table 8 reports the performance of different ρ values over the three datasets, where symbol – refers to the model without using the pre-trained POI embeddings for initialization. First, we observe that the models initialized with pre-trained POI embeddings outperform the model without this initialization by a large margin. The performance gain by using pre-training strategy is at least 28.6%, 61.4% and 131.5% in terms of MAP on *SIN*, *Gowalla* and *CA* datasets respectively. This validates the effectiveness of utilizing geographical distance and transition pattern between two POIs to pre-train POI embeddings. Second, all the settings with varying ρ values achieve similar performance. On *SIN* and *Gowalla* datasets, the optimal ρ values are 0.7 and 0.5 respectively. However, on *CA* dataset, the optimal ρ value is 0, indicating that transition patterns carry enough discriminative signal to help understand user's spatial intent. The close performance obtained with varying ρ values suggests that the geographical distance and transition patterns do not contain much complementary information. Based on Tobler's first law of geography, "Everything is related to everything else, but near things are more related than distant things." This indicates that when a user visits the next place, she will likely to visit a place near the place she visited last time. In this sense, the geographical influence could be encoded within the transition patterns, as being validated by the results. Accordingly, we set $\rho = 0$ in our experiments.

Efficiency On a workstation with a NVIDIA GTX 1080 GPU, we implemented the proposed NEXT framework based on Theano. NEXT takes 16, 30 and 14 minutes to finish one epoch of model training on *SIN*, *Gowalla* and *CA* datasets respectively. For recommendation score calculation, NEXT takes 1.75, 2.67 and 2.0 milliseconds per test instance (i.e., 2.06M/Hr, 1.35M/Hr and 1.8M/Hr) on the three datasets respectively. That is, after the model training, NEXT can be easily adopted in real-time applications with parallel

[4] We have investigated the calcaultion in different forms. Equations (20) and (21) produce the best performance

Table 8 Performance of NEXT with varying ρ values

ρ	SIN				Gowalla				CA			
	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP
-	0.1019	0.2378	0.3119	0.1722	0.0662	0.1728	0.2404	0.1245	0.0383	0.101	0.1406	0.0744
0	0.1405	0.2917	0.3649	0.2159	0.1282	0.2644	0.3339	0.1975	**0.1134**	**0.2403**	**0.3097**	**0.1789**
0.3	0.1428	0.2958	0.367	0.2189	0.1306	0.2666	0.3361	0.2000	0.1023	0.2145	0.2747	0.1626
0.5	0.1447	0.2973	0.3684	0.2202	**0.1316**	**0.2660**	**0.3386**	**0.2009**	0.1015	0.2077	0.2713	0.1605
0.7	**0.1456**	**0.2964**	**0.3697**	**0.2214**	0.1174	0.2504	0.3228	0.1860	0.1026	0.2148	0.2779	0.1628
1	0.1408	0.2919	0.2673	0.2163	0.1265	0.2605	0.3307	0.1949	0.106	0.221	0.2813	0.1666

Note: The best results are highlighted in boldface on each dataset

computing technique.

4.6 A single layer vs. multiple layers

In NEXT, we utilize an additional feed-forward neural network layer to deduce the users' spatial intent. Many existing works have validated the superiority of using a deep network structure for the recommendation tasks [1, 28, 31]. Does a deeper network structure deliver better recommendation performance under the framework of NEXT? To answer this question, here, we investigate the potential of adding further hidden layers within NEXT. Motivated by the existing work [1, 31], we study the following two variants with a deeper network structure on the basis of NEXT:

- **NEXT-k** A straightforward strategy is to stack more nonlinear layers to extract the higher-level intent features. Specifically, with $\mathbf{c}^\ell$, $\mathbf{h}^u$ and $\mathbf{h}^q_{t_i}$ from Eqs. (7), (9) and (12) respectively, we calculate the hidden intent vectors of kth layer as follows:

$$\begin{aligned}\mathbf{h}_2 &= ReLU(\mathbf{W}^h_2(\mathbf{h}^u + \mathbf{h}^q_{t_i}) + \mathbf{b}^h_2),\\ \mathbf{c}^\ell_2 &= ReLU(\mathbf{W}^c_2\mathbf{c}^\ell + \mathbf{b}^c_2),\\ &\cdots\\ \mathbf{h}_k &= ReLU(\mathbf{W}^h_k\mathbf{h}_{k-1} + \mathbf{b}^h_k),\\ \mathbf{c}^\ell_k &= ReLU(\mathbf{W}^c_k\mathbf{c}^\ell_{k-1} + \mathbf{b}^c_k),\end{aligned}$$

  where $\mathbf{W}^c_k$, $\mathbf{W}^h_k$, $\mathbf{b}^h_k$ and $\mathbf{b}^c_k$ are the transition matrices and bias vectors for kth layer. The recommendation score $y_{u,t_i,\ell}$ is then calculated as follows:

$$y_{u,t_i,\ell} = \mathbf{h}^T_k\mathbf{c}^\ell_k.$$

  Hence, NEXT can be considered as being equivalent to NEXT-1. Figure 5 illustrates the network architecture of Next-k.

- **NEXT-C** Concatenating hidden features of different components has been widely adopted in multimodal deep learning work [49]. Following the approach proposed in NeuMF [1], we apply a vector concatenation on the hidden vectors $\mathbf{h}^u$, $\mathbf{h}^q_{t_i}$ and $\mathbf{c}^\ell$, and learn their interactions via a standard MLP. Formally, the recommendation score $y_{u,t_i,\ell}$ is calculated as follows:

$$\begin{aligned}\mathbf{h}_1 &= (\mathbf{h}^u + \mathbf{h}^q_{t_i}) \oplus \mathbf{c}^\ell,\\ \mathbf{h}_2 &= ReLU(\mathbf{W}^c_2\mathbf{h}_1 + \mathbf{b}^c_2),\\ &\cdots\\ \mathbf{h}_k &= ReLU(\mathbf{W}^c_k\mathbf{h}_{k-1} + \mathbf{b}^c_k),\\ y_{u,t_i,\ell} &= \sigma(\mathbf{w}^T\mathbf{h}_k + b),\end{aligned} \tag{22}$$

  where $\oplus$ denotes the concatenation operation, $\mathbf{W}^c_k$ and $\mathbf{b}^c_k$ are the transition matrix and bias vector for the kth layer, σ is chosen to be sigmoid function. In Eq. (22), we derive the recommendation score $y_{u,t_i,\ell}$ with a logistic regression based on the hidden features extracted by the kth layer, where $\mathbf{w}$ works as the weight vector and σ is choosen to be the sigmoid function. We denote this model with k hidden layers as NEXT-C-k. Figure 6 illustrates the network architecture of NEXT-C.

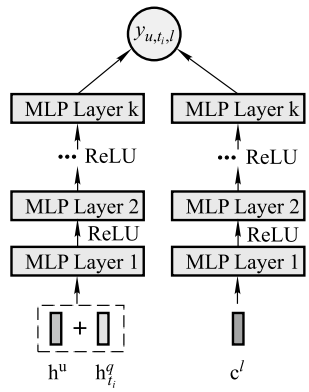

Fig. 5 Overall architecture of NEXT-k

Note that we set the nonlinear activation function to be ReLU for both NEXT-k and NEXT-C-k. This setting has also been used in NeuMF for its deep network structure [1]. ReLU has the advantage of alleviating vanishing gradient problems when the network is deep [50]. Also, this setting leads to a

fair comparison with NEXT, since we adopt ReLU as the activation function in NEXT to faciliate interpretation ability. Moreover, since NEXT-*k* and NEXT-C-*k* share the same architecture with NEXT for the first layer, we set the same setting as used by NEXT in Section 4.3 for a fair comparison.

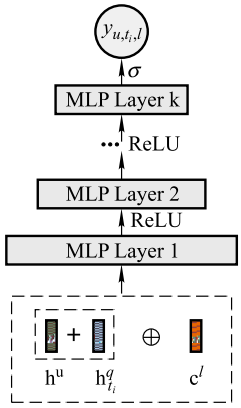

Fig. 6 Overall architecture of NEXT-C

Figures 7(a) and 7(b) plot the performance of NEXT-2 and NEXT-C-2 under different dimension settings for the second layer, where the performance of NEXT are also plotted as a reference (in dotted lines). First, we observe that both models with a deeper network structure result in a much poorer recommendation performance, compared to NEXT. Both NEXT-2 and NEXT-C-2 achieve the optimal performance with the dimension number of the second layer being 60/60/50 on *SIN*/*Gowalla*/*CA* datasets respectively. Also, the performance deteriorates significantly when the dimension number of the second layer becomes smaller.

Second, we observe that NEXT-2 obtains much better recommendation performance than NEXT-C-2 across the three datasets under different dimension numbers. Under the optimal settings, NEXT-2 delivers a relative performance reduction of 12.9%, 9.3% and 18.3% in terms of MAP for *SIN*, *Gowalla* and *CA* respectively, compared to NEXT. Similarly, the relative performance reduction obtained by NEXT-C-2 over NEXT is 29.8%, 31.1% and 33.5% respectively. Note that neural networks can approximate any continuous function to arbitrary precision [27]. Stacked nonlinear transformations and vector concatenation operations enable us to better learn complex interactions between user and item features [1]. The inferior performance obtained by NEXT-2 and NEXT-C-2, however, validate that the context factors are more critical in understanding the user's spatial behavior. Recall the workflow of NEXT demonstrated in Fig. 2: a) we encode the sequential relations in POI embedding pre-training phase; b) then, we combine the embedding vectors from the users, POIs and their associated meta-data information; c) at last, we apply nonlinearity transformations (with temporal context) to derive the latent intent vectors for users and POIs. The experimental results suggest that a single nonlinearity layer (with temporal context) applied over the embeddings encoded with sequential relations leads to a compact intent feature learning process. More complicated modeling attached to this architecture degenerates the effective feature learning. That is, the mechanisms proposed in NEXT to incorporate temporal context, sequential relations, geographical influence and auxiliary meta-data information work together as an integrated architecture, leading to superior performance for next POI recommendations.

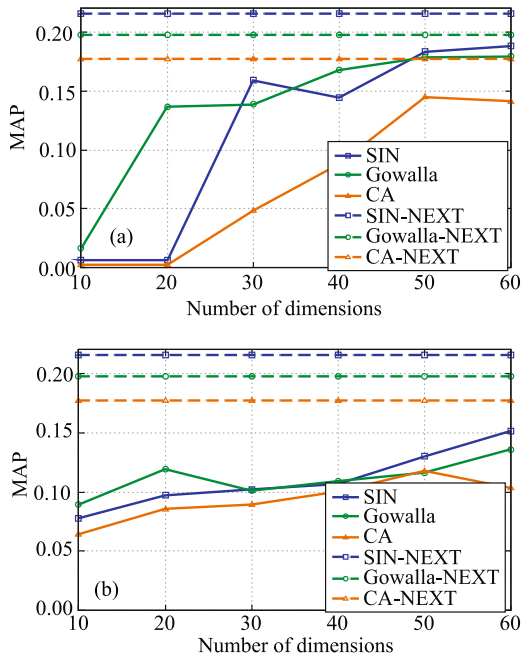

Fig. 7 Performance of NEXT-2 (a) and NEXT-C-2 (b) with varying dimension numbers for the second layer. (a) NEXT-2; (b) NEXT-C-2

5 Conclusion

In this paper, we propose a simple neural network framework for next POI recommendations, named NEXT. NEXT derives the spatial intent for a user by calculating POI-based intent and user-based intent separately based on two individual ReLU nonlinearities. Under this framework, we incorporate different contextual factors to enhance next POI recommendations in a unified architecture. Specifically, we incorporate two kinds of temporal context to enhance the intent calculation process. Furthermore, we adopt DeepWalk to encode the spatial constraints such as geographical information and sequential relations pattern into POI embeddings through a

pre-training scheme. Comprehensive experiments are conducted on three real-world datasets. The experimental results show that the proposed NEXT outperforms existing state-of-the-art alternatives in terms of MAP and Acc@K. We further show that NEXT achieves promising performance in the task of cold-start user recommendations. This uniqueness makes NEXT a preferable choice in real-world applications. As a future work, we plan to devise some effective fusion strategies to combine the different modeling methodologies together for better next POI recommendation accuracy. Also, the proposed NEXT involves several tunable parameters. The guideline to ease optimal parameter setting would be a desirable feature for real-world applications. We will further investigate this possibiity in future work. We observed that WMF achieves much better performance than PMF and other factorization based models. It is interesting to devise an attention mechanism to discriminate the user-POI interactions for better performance. Moreover, we like to devise mechanisms based on the semantic context factors to enable recommendation explanation for next POI recommendations. Note that we can also fuse the POI embedding learning and recommendation modeling as a unified model. This joint learning strategy will be investigated in the future.

Acknowledgements This research was supported by the National Natural Science Foundation of China (Grant Nos. 61872278, 61502344, 1636219, U1636101), Natural Science Foundation of Hubei Province (2017CFB502), Academic Team Building Plan for Young Scholars from Wuhan University (Whu2016012) and Singapore Ministry of Education Academic Research Fund Tier 2 (MOE2014-T2-2-066). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

References

1. He X, Liao L, Zhang H, Nie L, Hu X, Chua T S. Neural collaborative filtering. In: Proceedings of International Conference on World Wide Web. 2017, 173–182
2. Cheng C, Yang H, Lyu M R, King I. Where you like to go next: successive point-of-interest recommendation. In: Proceedings of International Joint Conference on Artificial Intelligence. 2013, 2605–2611
3. Feng S, Li X, Zeng Y, Cong G, Chee Y M, Yuan Q. Personalized ranking metric embedding for next new POI recommendation. In: Proceedings of International Joint Conference on Artificial Intelligence. 2015, 2069–2075
4. Ye M, Yin P, Lee W C, Lee D L. Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval. 2011, 325–334
5. Li X, Cong G, Li X L, Pham T A N, Krishnaswamy S. Rank-GeoFM: a ranking based geographical factorization method for point of interest recommendation. In: Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval. 2015, 433–442
6. Ye M, Yin P, Lee W C. Location recommendation for location-based social networks. In: Proceedings of SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2010, 458–461
7. Xiong L, Chen X, Huang T K, Schneider J G, Carbonell J G. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In: Proceedings of SIAM International Conference on Data Mining. 2010, 211–222
8. Yuan Q, Cong G, Ma Z, Sun A, Thalmann N M. Time-aware point-of-interest recommendation. In: Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval. 2013, 363–372
9. Xie M, Yin H, Wang H, Xu F, Chen W, Wang S. Learning graph-based POI embedding for location-based recommendation. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2016, 15–24
10. Zhang W, Wang J. Location and time aware social collaborative retrieval for new successive point-of-interest recommendation. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2015, 1221–1230
11. Yin H, Cui B, Zhou X, Wang W, Huang Z, Sadiq S. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. ACM Transaction on Information Systems, 2016, 35(2): 11
12. Cheng C, Yang H, King I, Lyu M R. Fused matrix factorization with geographical and social influence in location-based social networks. In: Proceedings of AAAI Conference on Artificial Intelligence. 2012, 17–23
13. Liu B, Fu Y, Yao Z, Xiong H. Learning geographical preferences for point-of-interest recommendation. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2013, 1043–1051
14. Liu B, Xiong H, Papadimitriou S, Fu Y, Yao Z. A general geographical probabilistic factor model for point of interest recommendation. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5): 1167–1179
15. Gao H, Tang J, Liu H. gSCorr: modeling geo-social correlations for new check-ins on location-based social networks. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2012, 1582–1586
16. 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: Proceedings of AAAI Conference on Artificial Intelligence. 2016, 137–143
17. Zhao S, Zhao T, Yang H, Lyu M R, King I. STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation. In: Proceedings of AAAI Conference on Artificial Intelligence. 2016, 315–322
18. Mikolov T, Karafiát M, Burget L, Černocký J, Khudanpur S. Recurrent neural network based language model. In: Proceedings of Annual Conference of the International Speech Communication Association. 2010, 1045–1048
19. Mikolov T, Chen K, Corrada G, Dean J. Efficient estimation of word

representations in vector space. 2013, arXiv preprint arXiv: 1301.3781

20. Cho K, Merrienboer B V, Gülçehre Ç, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 2014, 1724–1734
21. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proceedings of International Conference on Learning Representations. 2015
22. Wang B, Liu K, Zhao J. Inner attention based recurrent neural networks for answer selection. In: Proceedings of Annual Meeting of the Association for Computational Linguistics. 2016, 1288–1297
23. Allamanis M, Peng H, Sutton C. A convolutional attention network for extreme summarization of source code. In: Proceedings of International Conference on Machine Learning. 2016, 2091–2100
24. Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation. Technical Report, DTIC Document, 1985
25. Werbos P J. Generalization of backpropagation with application to a recurrent gas market model. Neural Networks, 1988, 1(4): 339–356
26. Bishop C M. Neural Networks for Pattern Recognition. Oxford: Oxford University Press, 1995
27. Hornik K, Stinchcombe M B, White H. Multilayer feedforward networks are universal approximators. Neural Networks, 1989, 2(5): 359–366
28. Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations. In: Proceedings of ACM Conference on Recommender Systems. 2016, 191–198
29. Kim D H, Park C, Oh J, Lee S, Yu H. Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of ACM Conference on Recommender Systems. 2016, 233–240
30. Liu Q, Wu S, Wang L, Tan T. Predicting the next location: a recurrent model with spatial and temporal contexts. In: Proceedings of AAAI Conference on Artificial Intelligence. 2016, 194–200
31. Zheng L, Noroozi V, Philip S Y. Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of ACM International Conference on Web Search and Data Mining. 2017, 425–434
32. Rendle S. Factorization machines with libFM. ACM Transactions Intelligent Systems and Technology, 2012, 3(3): 57
33. Elman J L. Finding structure in time. Cognitive Science, 1990, 14(2): 179–211
34. Yan R. i, poet: automatic poetry composition through recurrent neural networks with iterative polishing schema. In: Proceedings of International Joint Conference on Artificial Intelligence. 2016, 2238–2244
35. Zhang Y, Dai H, Xu C, Feng J, Wang T, Bian J, Wang B, Liu T Y. Sequential click prediction for sponsored search with recurrent neural networks. In: Proceedings of AAAI Conference on Artificial Intelligence. 2014, 1369–1375
36. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780
37. Chen X, Qiu X, Zhu C, Liu P, Huang X. Long short-term memory neural networks for Chinese word segmentation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing. 2015, 1197–1206
38. Rocktäschel T, Grefenstette E, Hermann K M, Kociský T, Blunsom P. Reasoning about entailment with neural attention. In: Proceedings of International Conference on Learning Representations. 2016
39. Chung J, Gülçehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In: Proceedings of 2014 Workshop on Deep Learning. 2014
40. Manotumruksa J, Macdonald C, Ounis I. A deep recurrent collaborative filtering framework for venue recommendation. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2017, 1429–1438
41. Feng J, Li Y, Zhang C, Sun F, Meng F, Guo A, Jin D. Deepmove: predicting human mobility with attentional recurrent networks. In: Proceedings of International Conference on World Wide Web. 2018, 1459–1468
42. Perozzi B, Rami A R, Skiena S. Deepwalk: online learning of social representations. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014, 701–710
43. Cho E, Myers S A, Leskovec J. Friendship and mobility: user movement in location-based social networks. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2011, 1082–1090
44. Zhang J D, Chow C Y. Geosoca: exploiting geographical, social and categorical correlations for point-of-interest recommendations. In: Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval. 2015, 443–452
45. Yang C, Liu Z, Zhao D, Sun M, Chang E Y. Network representation learning with rich text information. In: Proceedings of International Joint Conference on Artificial Intelligence. 2015, 2111–2117
46. Chen J, Zhang Q, Huang X. Incorporate group information to enhance network embedding. In: Proceedings of ACM International Conference on Information and Knowledge Management. 2016, 1901–1904
47. Salakhutdinov R, Mnih A. Probabilistic matrix factorization. In: Proceedings of International Conference on Neural Information Processing Systems. 2007, 1257–1264
48. Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In: Proceedings of IEEE International Conference on Data Mining. 2008, 263–272
49. Srivastava N, Salakhutdinov R. Multimodal learning with deep boltzmann machines. In: Proceedings of International Conference on Neural Information Processing Systems. 2012, 2231–2239
50. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of International Conference on Artificial Intelligence and Statistics. 2011, 315–323

Zhiqian Zhang is currently a Master student at Wuhan University, under the supervision of Dr. Chenliang Li. She received Bachelor degree from Wuhan University, China in 2015. Her research interests include natural language processing, information retrieval, data mining, and social media analysis and mining.

Chenliang Li received PhD from Nanyang Technological University, Singapore in 2013. Currently, he is an associate professor at School of Cyber Science and Engineering, Wuhan University, China. His research interests include information retrieval, text/Web mining, data mining, and natural language processing. His papers appear in SIGIR, CIKM, ACL, AAAI, TOIS, TKDE, and JASIST.

Zhiyong Wu is a currently PhD student at Department of Computer Science, the University of Hong Kong, China. He received Bachelor degree from Wuhan University, China in 2017. His research interests include data mining, natural language processing, and database.

Aixin Sun is an associate professor with School of Computer Engineering, Nanyang Technological University, Singapore. He received PhD from the same school in 2004. His research interests include information retrieval, text mining, social computing, and multimedia. His papers appear in major international conferences like SIGIR, KDD, WSDM, ACM Multimedia, and journals including TOIS, TKDE, and JASIST.

Dengpan Ye is currently a professor in School of Cyber Science and Engineering, Wuhan University, China. He received the BASc in automatic control from SCUT in 1996 and PhD degree at NJUST in 2005 respectively. He worked as a Post-Doctoral Fellow in Information System School of Singapore Management University, Singapore. His research interests include machine learning and multimedia security. He is the author or co-author of more than 30 refereed journal and conference papers.

Xiangyang Luo is currently a professor at Zhengzhou Science and Technology Institute and the State Key Laboratory of Mathematical Engineering and Advanced Computing, China. His research interests lie in multimedia security and cyberspace surveying and mapping. He is the author or co-author of more than 100 refereed international journal and conference papers. He has obtained the support of the National Natural Science Foundation of China and the National Key R&D Program of China.