Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation

Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S. Sheng, *Senior Member, IEEE*, and Xiaofang Zhou, *Fellow, IEEE*

Abstract—Next Point-of-Interest (POI) recommendation which is of great value to both users and POI holders is a challenging task since complex sequential patterns and rich contexts are contained in extremely sparse user check-in data. Recently proposed embedding techniques have shown promising results in alleviating the data sparsity issue by modeling context information, and Recurrent Neural Network (RNN) has been proved effective in the sequential prediction. However, existing next POI recommendation approaches train the embedding and network model separately, which cannot fully leverage rich contexts. In this paper, we propose a novel unified neural network framework, named NeuNext, which leverages POI context prediction to assist next POI recommendation by joint learning. Specifically, the Spatio-Temporal Gated Network (STGN) is proposed to model personalized sequential patterns for users' long and short term preferences in the next POI recommendation. In the POI context prediction, rich contexts on POI sides are used to construct graph, and enforce the smoothness among neighboring POIs. Finally, we jointly train the POI context prediction and the next POI recommendation to fully leverage labeled and unlabeled data. Extensive experiments on real-world datasets show that our method outperforms other approaches for next POI recommendation in terms of Accuracy and MAP.

Index Terms—Next POI recommendation, POI context prediction, joint learning

1 Introduction

Recent years have witnessed a revolution in location-based social network (LBSN) services, such as Foursquare,[1] Facebook Places,[2] Gowalla,[3] Yelp[4] and so on. These LBSN services have attracted many users to check in and share their locations, tips, and experiences with their friends when they visit preferred Point-of-Interests (POIs), e.g., restaurants, hotels, and sightseeing sites. As a consequence, large amounts of geo-tagged data have been accumulated. For example, Foursquare has attracted 55 million users with more than 10 billion check-ins until December 2017.[5] These online footprints (or check-ins) provide an excellent opportunity to understand user preferences on POIs and can be used for POI recommendation [1], which is of a great value to help users explore surroundings and also benefits POI holders to launch advertisements to target customers for marketing.

However, there are three major challenges [2], [3], [4] to be tackled in the next POI recommendation.

Data Sparsity. Most recommendation tasks need to deal with the data sparsity problem, but the check-in data in the POI recommendation task is much sparser. As described in [5], [6], the density of the check-in data usually is around 0.1 percent, while that of Netflix data for movie recommendation is around 1.2 percent. The data sparsity problem has a greater impact on the next POI recommendation because sparse data leads to short sequence which makes it difficult to capture user's sequential pattern. Furthermore, data for POI recommendation used to be binary implicit rather than explicit, so the regularity of user behaviors is not conducive to being discovered and exploited.

Sequential Pattern. Compared to POI recommendation, next POI recommendation, which is to recommend POIs for a user to visit in the near future, considers the sequential information of users' check-ins in addition to users' preferences [7]. The sequential pattern of user behaviors usually has two meanings. On the one hand, users take specific actions at specific times, such as visiting a restaurant at lunchtime and going to the bar at night. On the other hand, there is a certain chronological order between the actions of users, resulting in a specific order of the POIs, which also reflects the personalized preferences of different users. As shown in Fig. 1, although both $user1$ and $user2$ like watching a movie after dinner, $user1$ chooses to go to the gym after watching the movie, while $user2$ goes to the cafe. At the same time, we can observe that $user1$ prefers to wash

1. http://foursquare.com.
2. https://www.facebook.com/directory/places
3. http://blog.gowalla.com
4. http://www.yelp.com.
5. https://foursquare.com/about

• *Pengpeng Zhao, Anjing Luo, Jiajie Xu, and Zhixu Li are with the Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou 215006, China. E-mail: {ppzhao, xujj, zhixuli}@suda.edu.cn, ajluo@stu.suda.edu.cn.*
• *Yanchi Liu is with Rutgers University, Piscataway, NJ 08854 USA. E-mail: yanchi@nec-labs.com.*
• *Fuzhen Zhuang is with the Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China, and also with the Henan Institutes of Advanced Technology, Zhengzhou University, Zhengzhou 450052, P.R. China. E-mail: zhuangfuzhen@ict.ac.cn.*
• *Victor S. Sheng is with the Department of Computer Science, Texas Tech University, Lubbock, TX 79409 USA. E-mail: victor.sheng@ttu.edu.*
• *Xiaofang Zhou is with the University of Queensland, Brisbane, QLD 4000, Australia. E-mail: zxf@itee.uq.edu.au.*

Manuscript received 16 Nov. 2019; revised 30 May 2020; accepted 25 June 2020. Date of publication 6 July 2020; date of current version 1 Apr. 2022.
(Corresponding author: Pengpeng Zhao.)
Recommended for acceptance by R. C-W. Wong.
Digital Object Identifier no. 10.1109/TKDE.2020.3007194

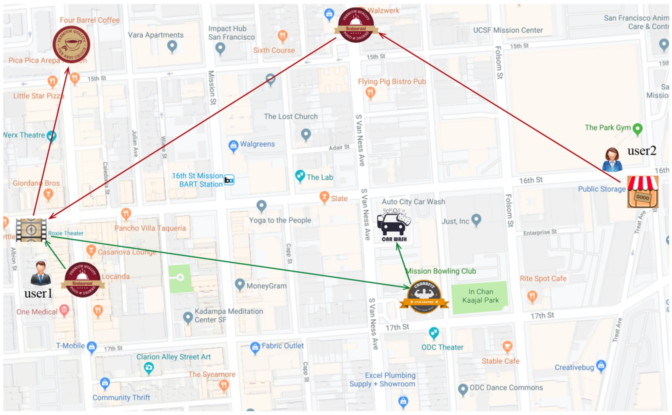


Fig. 1. An illustration of the sequential pattern in a real-world next POI recommendation.

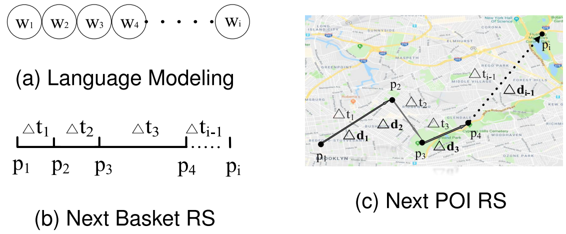


Fig. 2. w_i in (a) represents the ith word. In (b), p_i represents the ith item and $\triangle t$ is time interval between two neighbor items. In (c), $\triangle d$ further represents distance interval between two successive check-ins.

the car after finishing the exercise, and $user2$ goes shopping before dinner.

Various Contexts. POI recommendation has a wide range of contextual factors, including the temporal context, geographical influence, user social influence, auxiliary metadata information such as textual description, and so on. POI recommendation can effectively improve the performance by considering the available context since users tend to have different choices and needs at different time and places. In fact, the available contexts are often limited by the datasets. To this end, in this paper, we only use the time intervals, distance intervals, and the nearby POIs as POI contexts because datasets only contain user-POI interactions and locations. These contexts, which can model sequential and collaborative information, are naturally available.

In the literature, approaches like latent factor model and Markov chain have been widely applied for sequential data analysis and recommendation. Rendle *et al.* [8] proposed Factorizing Personalized Markov Chain (FPMC), which bridges matrix factorization and Markov chains together, for next-basket recommendation. [2] extended FPMC to embed personalized Markov chain and user movement constraint for next POI recommendation. [4] proposed a unified tensor-based latent model to capture the successive check-in behavior by exploring the latent pattern-level preference for each user. Recently, Recurrent Neural Networks (RNNs) have been successfully employed to model sequential data and become state-of-the-art methods. [9] focused on RNN solutions for session-based recommendation task, where no user id exists, and recommendations are made on short session data. [10] proposed a variant of Long-Short Term Memory network (LSTM), called Time-LSTM, to equip LSTM with time gates to model time intervals for next item recommendation.

However, none of the above recommendation methods considers both time intervals and geographical distances between neighbor items, which makes next POI recommendation different from other sequential tasks such as language modeling and next-basket recommender system (RS). As shown in Fig. 2, there is no spatio-temporal interval between neighbor words in language modeling, and there is no distance interval between neighbor items in next-basket RS, while there are time and distance intervals between neighbor check-ins in next POI recommendation. Some recent efforts have been made to extend RNNs for modeling dynamic time and distance intervals. For example, HST-LSTM [11] combines spatio-temporal influence into LSTM while it is for general location recommendation. A recent work ST-RNN [12] tried to extend RNN to model the temporal and spatial context for next location prediction. In order to model temporal context, ST-RNN models multi-check-ins in a time window in each RNN cell. Meanwhile, ST-RNN employs time-specific and distance-specific transition matrices to characterize dynamic time intervals and geographical distances, respectively. However, there exists some challenges preventing ST-RNN from becoming the best solution for next POI recommendation.

First of all, ST-RNN may fail to model spatial and temporal relations of neighbor check-ins properly. ST-RNN adopts time-specific and distance-specific transition matrices between cell hidden states within RNN. Due to data sparsity, ST-RNN employs linear interpolation to partition time interval and geographical distance into discrete bins respectively instead of learning every possible continuous time intervals and geographical distances. Second, ST-RNN is designed for short-term interests and not well designed for long-term interests. Jannach *et al.* [13] reported that users' short-term and long-term interests are both significant on achieving the best performance. The short-term interest here means that recommended POIs should depend on recently visited POIs, and the long-term interest means that recommended POIs should depend on all historical visited POIs. Third, it is hard to select the proper width of the time window for different applications in ST-RNN since it models multi-elements in a fixed time period.

More recently, semi-supervised learning (SSL), which aims at leveraging unlabeled data to boost the overall learning preference, has been introduced into POI recommendation. Yang *et al.* [14] developed Preference And Context Embedding (PACE), a deep neural architecture that jointly learned the embeddings of users and POIs to predict both user preference over POIs and various context associated with users and POIs. Nevertheless, PACE ignored the sequential pattern which made it not suitable for next POI recommendation.

To this end, in this paper, we propose a novel unified neural network framework, named NeuNext, to jointly learn POI embedding and network model from unlabeled context data and labeled check-in data to boost the next POI recommendation performance. The proposed NeuNext framework consists of two parts: next POI recommendation and POI context prediction. The NeuNext jointly optimizes the supervised loss over labeled check-in data and the unsupervised loss over labeled data and unlabeled context data to alleviate data sparsity and aims to leverage unlabeled data to improve next POI recommendation performance. Specifically, in the next POI recommendation, we propose a Spatio-Temporal Gated Network (STGN) by enhancing long short term memory, to model users' sequential visiting behaviors. Time intervals and distance intervals of neighbor check-ins are modeled by time gates and distance gates, respectively. Note that there are two time gates and two distance gates in the STGN model. One pair of time gate and distance gate are designed to exploit time and distance intervals to capture the short-term interest, and the other pair are introduced to memorize time and distance intervals to model the long-term interest. Furthermore, enlightened by [15], we use the coupled input and forget gates to reduce the number of parameters, making our model more efficient. In the POI context prediction, we use unlabeled information on POI sides to integrate rich context into the embedding vector. Then we jointly learn the next POI prediction part and the POI context prediction part through shared POI embedding. Different from previous work, the embedding of context is jointly trained with next POI recommendation in NeuNext. In this way, NeuNext is able to fully leverage labeled check-in data and unlabeled context data to improve the recommendation effectiveness and thus alleviate data sparsity issue.

To demonstrate the effectiveness of our proposed approach, we conducted experiments on four real-world datasets, and our results show that NeuNext improves the performance of next POI recommendation compared to the state-of-the-art methods. We summarize our contributions as follows:

- To the best of our knowledge, our model NeuNext, which models various contexts and users' sequential visiting patterns jointly, is the first work on employing hybrid embedding and recurrent neural network techniques for next POI recommendation.
- We propose an enhanced long-short term memory method, named STGN, to model users' short-term and long-term sequential preferences respectively.
- The proposed method has been evaluated on large-scale real-world data for next POI recommendation. Our experimental results show that our method outperforms state-of-the-art methods in terms of different metrics, such as accuracy and MAP.

The rest of this paper is organized as follows. We begin by briefly reviewing related work in Section 2. Then in Section 3, we introduce the preliminaries of this study. In Section 4, we introduce our proposed model NeuNext for next POI recommendation. Experimental results are shown in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

We survey the related work from three aspects, including general POI recommendation, next POI recommendation, and leveraging neural networks for recommendation.

2.1 General POI Recommendation

Collaborative filtering is widely used in POI recommendation. The state-of-the-art collaborative filtering (CF) is based on matrix factorization and its variants [16], [17], [18], [19]. Salakhutdinov & Mnih [16] proposed a PMF model in a Bayesian probabilistic framework to include Gaussian noise in observations. Under the Gaussian assumption, maximizing the posterior probability over latent features is equivalent to minimizing the square error.

Recently, more advanced models have been proposed to exploit additional information for POI recommendation [20], [21], such as check-in locations, social influence, temporal information, review information and transition between POIs. Ye *et al.* [22], [23] considered the social influence under the framework of a user-based CF model and modeled the geographical influence by a Bayesian CF model. Levandoski *et al.* [24] employed item-based CF to make POI recommendations with the consideration of travel penalty, which is proportional to the distance between a target user and a POI. Kurashima *et al.* [25] proposed a topic model, in which a POI is sampled based on the topics and the distance to historical POIs visited by a target user. Liu *et al.* [26] estimated the geographical correlations of check-in POIs by a power-law distribution. Zhang *et al.* [27] directly applied kernel density estimation to this distribution. And Lian *et al.* [28] incorporated the geographical information into weighted matrix factorization to improve the effectiveness of POI recommendation. Moreover, both Yuan *et al.* [29] and Gao *et al.* [30] introduced the temporal preference to enhance the efficiency and effectiveness of Ye *et al.*'s solution. Cheng *et al.* [31] considered more comprehensive information, such as the multi-center of user check-in patterns, and the skewed user check-in frequency. Moreover, Liu *et al.* [7] proposed a bi-weighted low-rank graph construction model, which integrates users' interests and their evolving sequential preferences with temporal interval assessment to provide POI recommendations for a specific time period.

2.2 Next POI Recommendation

The next POI recommendation is a newly emerging task, and is even more challenging than POI recommendation. In the literature, there exist only a few works on next POI recommendation and the sequential influence between successive check-ins is not yet well-studied in these works. As a commonly-used method for sequential prediction, Markov Chain (MC) based models aim to predict the next behavior of a user based on the past sequential behaviors. In these methods, an estimated transition matrix indicates the probability of a behavior based on the past behaviors. A tensor-based FPMC-LR [2] model was proposed by considering first-order Markov chain for POI transitions and distance constraints. It aimed to recommend POIs for next hours by merging consecutive check-ins in previous hours. It employed FPMC [8] to model the personalized POI transition. Based on the current POI, their method only considered the POIs in the defined region as candidates. Lian *et al.* [32] adopted FPMC to present the short-term and the long-term preference to predict the next check-in. S. Feng *et al.* [3] proposed a personalized ranking metric embedding method (PRME) to model personalized check-in sequences for next new POI recommendation. He *et al.* [4] proposed a tensor-based latent model for next POI recommendation under the influence of user's latent behavior patterns, which is determined by the contextual scenarios including temporal and categorical information. Zhao *et al.* [33] developed a ranking-based pairwise tensor factorization framework STELLAR to incorporate fine-grained temporal context (i.e., month, weekday/weekend and hour), which has a significant improvement in next POI recommendation. Xie *et al.* [34] 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. Four pairs of context factors, i.e., POI-POI, POI-Region, POI-Time, and POI-Word, were modeled in a unified optimization framework.

2.3 Neural Networks in Recommendation

Recently, deep neural networks have been widely used in various fields including recommender systems. First, neural networks are naturally used as feature learning to model various features of users or items, such as textual description of items [35], acoustic features of music [36], cross-domain behaviors of users [37] and the information in knowledge bases [38]. Second, neural networks have been explored as a core recommendation model to model non-linear complex interaction between users and items, and have more expressive than MF. The Restricted Boltzmann Machines [39] is the first work in collaborative filtering, which discretizes users' ratings and models the hidden features of users or items underlying the ratings. Zheng *et al.* [40] further improved it with an autoregressive method. Recently, autoencoders have also been successfully adopted for learning users' representations based on rated items [41]. Yang *et al.* [14] proposed a deep neural architecture named PACE for POI recommendation, which combines collaborative filtering and semi-supervised learning. Liu *et al.* [12] proposed a ST-RNN model for next POI recommendation, which extends a RNN model with Spatial and temporal constraints. Yang *et al.* [42] proposed a neural network model named JNTM, which jointly models a social network structure and users' trajectory behaviors.

Differences. The major differences between our work and related research described above are in several aspects. PACE [14] uses multi-layer perceptron (MLP) to model users' preference over POIs and jointly learns with contextual embeddings, but it is designed for POI recommendation which ignore users' visiting sequential patterns. ST-RNN [12] uses a RNN model to capture users' visiting sequential patterns. However, it only incorporates temporal and geographical influence for recommendation. JNTM [42] that uses both RNN and GRU to capture users' short-term and long-term visiting sequential patterns only jointly learns the social influence. To the best of our knowledge, our model NeuNext, which models various contexts and users' sequential visiting patterns jointly, is the first work on employing hybrid embedding and recurrent neural network techniques for next POI recommendation. Moreover, we jointly learn losses on preference and on context to improve the recommendation performance with fully leveraging unlabeled contextual data.

3 Problem Formulation and Framework Overview

In this section, we first give the problem statement, and then introduce LSTM which is the basic RNN-based method to lay the foundation for our approach. Finally, we show the framework overview consisting of next POI recommendation and POI context prediction.

3.1 Problem Statement

Let $\mathbb{U} = \{u_1, u_2, \ldots, u_M\}$ be the set of M users and $\mathbb{V} = \{v_1, v_2, \ldots, v_N\}$ be the set of N POIs. For user u, she has a sequence of historical POI visits up to time t_{i-1} represented as $H_i^u = \{v_{t_1}^u, v_{t_2}^u, \ldots, v_{t_{i-1}}^u\}$, where $v_{t_i}^u$ means user u visit POI v at time t_i. $\mathbb{X} = \{x_{t_1}^u, x_{t_2}^u, \ldots, x_{t_{i-1}}^u\}$ is the embedding of the sequence of POIs, each row of which represents a POI embedding. The goal of next POI recommendation is to recommend a list of unvisited POIs for a user to visit next at time point t_i. Specifically, a higher prediction score of a user u to an unvisited POI v_j indicates a higher probability that the user u would like to visit v_j at time t_i. According to prediction scores, we can recommend top-k POIs to user u.

3.2 LSTM

LSTM [43], a variant of RNN, is capable of learning short and long-term dependencies. LSTM has become an effective and scalable model for sequential prediction problems, and many improvements have been made to the original LSTM architecture. We use the basic LSTM model in our approach for the concise and general purpose, and it is easy to extend to other variants of LSTM. The basic update equations of LSTM are as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{1}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \tag{2}$$

$$\widetilde{c_t} = \tanh(W_c[h_{t-1}, x_t] + b_c), \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c_t}, \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{5}$$

$$h_t = o_t \odot \tanh(c_t), \tag{6}$$

where i_t, f_t, o_t represent the input, forget and output gates of the tth object, deciding what information to store, forget and output, respectively. c_t is the cell activation vector representing cell state, which is the key to LSTM. x_t and h_t represent the input feature vector and the hidden output vector, respectively. σ represents a sigmoid layer to map the values between 0 to 1, where 1 represents "complete keep this" while 0 represents "completely get rid of this". W_i, W_f, W_o and W_c are the weights of gates. b_i, b_f, b_o and b_c are corresponding biases. And $\odot$ represents for the element-wise (Hadamard) product. The update of cell state c_t has two parts. The former part is the previous cell state c_{t-1} that is controlled by forget gate f_t, and the latter part is the new candidate value scaled by how much to add state value.

3.3 Framework Overview

The overall representation of our NeuNext framework is shown in Fig. 3. Our NeuNext model makes predictions on both user preferences over POIs and context associated with POIs. Specifically, there are two parts in NeuNext:

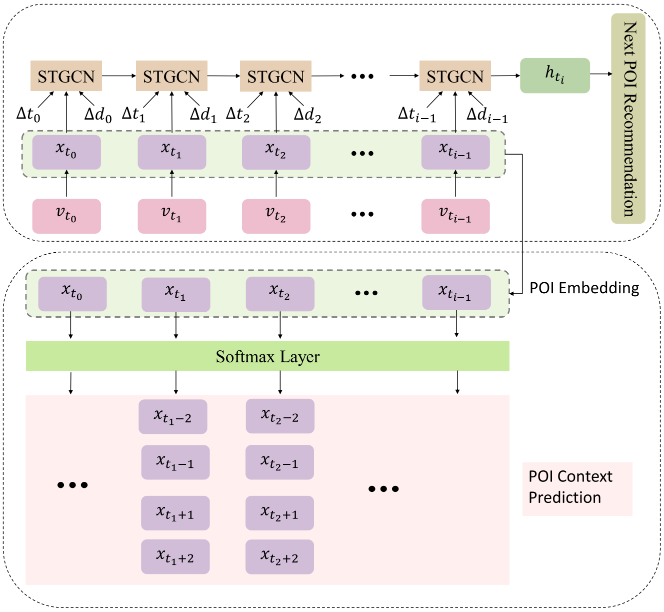


Fig. 3. Framework of NeuNext which makes predictions on both user preferences over POIs and context of POI. Both tasks optimize the POI embedding to improve the performance of next POI recommendation.

next POI recommendation captures users' visiting sequential pattern and POI context prediction enforces the smoothness among POIs and context. The basic input of the model is user-POI check-in sequences.

Next POI Recommendation. Recently, the recurrent neural network has been successfully employed for sequential recommendation. Therefore, we model the user's visiting sequence generation process with a sequential neural network method.We propose a new Spatio-Temporal Gated Network by enhancing long-short term memory, named STGN, with two pairs of time gates and distance gates to model users' long-term and short-term preference, respectively.

POI Context Prediction. For each POI in the user's check-in history, Skipgram model is used to predict the context of the POI on the POI context graph. The POI context graph contains most of the context related to POIs, such as nearby POIs and geographical information, which is often available as pairs of longitudes and latitudes. After getting the context of the POI, the objective of Skipgram is to optimize the log loss of predicting the context using the embedding of the POI. Note that the window size of the POI context prediction in Fig. 3 is 2.

Joint Learning. NeuNext is jointly trained according to two types of objective functions based on shared POI embedding to recommend next POI and predict the context of POIs. The objective functions consist of the supervised loss on labeled check-in data and the unsupervised loss or regularization penalty employed to enforce the smoothness among POIs and context. NeuNext is jointly trained by optimizing the sum of the above two loss which will learn a better POI embedding.

4 Our Approach

We will present the details of the proposed framework NeuNext in this section, including STGN and its variation which is used to learn sequential preference, POI context prediction, and joint learning.

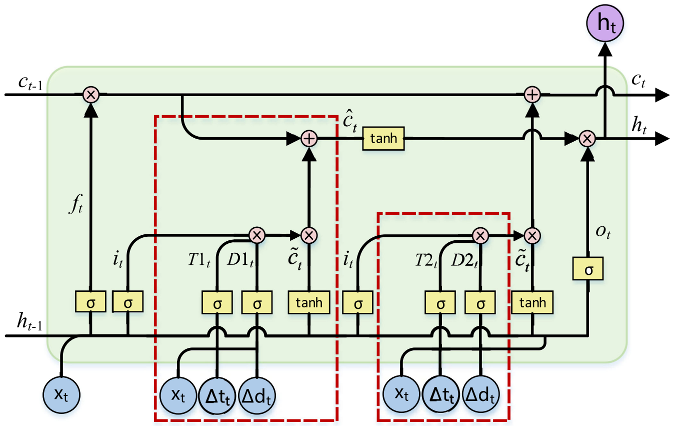


Fig. 4. STGN has two time gates and two distance gates, i.e., $T1_t$, $T2_t$, $D1_t$ and $D2_t$. $T1_t$ and $D1_t$ are designed to model time and distance intervals for short-term interests while $T2_t$ and $D2_t$ are to model time and distance intervals for long-term interest.

4.1 Next POI Recommendation

LSTM has proven its advantages in sequential preference learning in many areas. When using LSTM for next POI recommendation, x_t represents user's last visited POI, which can be exploited to learn user's short-term interest. While c_{t-1} contains the information of user's historical visited POIs, which reflect user's long-term interest. However, how much the short-term interest determines where to go next heavily depends on the time interval and the geographical distance between the last POI and the next POI. Intuitively, a POI visited long time ago and long distance away has little influence on next POI, and vice versa. In our proposed spatio-temporal gated network model, we use time gate and distance gate to control the influence of the last visited POI on next POI recommendation. Furthermore, the time gate and the distance gate can also help to store time and distance intervals in cell state c_t, which memorizes user's long-term interest. In this way, we utilize time and distance intervals to model user's short-term interest and long-term interest simultaneously.

As shown in two dotted red rectangles in Fig. 4, we add two time gates and two distance gates to LSTM, denoted as $T1_t$, $T2_t$, $D1_t$ and $D2_t$ respectively. $T1_t$ and $D1_t$ are used to control the influence of the latest visited POI on next POI, and $T2_t$ and $D2_t$ are used to capture time and distance intervals to model user's long-term interest. Based on LSTM, we add equations for time gates and distance gates as follows:

$$T1_t = \sigma(x_t W_{xt_1} + \sigma(\triangle t_t W_{t_1}) + b_{t_1}), \quad s.t. W_{xt_1} \leq 0 \tag{7}$$

$$T2_t = \sigma(x_t W_{xt_2} + \sigma(\triangle t_t W_{t_2}) + b_{t_2}), \tag{8}$$

$$D1_t = \sigma(x_t W_{xd_1} + \sigma(\triangle d_t W_{d_1}) + b_{d_1}), \quad s.t. W_{xd_1} \leq 0 \tag{9}$$

$$D2_t = \sigma(x_t W_{xd_2} + \sigma(\triangle d_t W_{d_2}) + b_{d_2}). \tag{10}$$

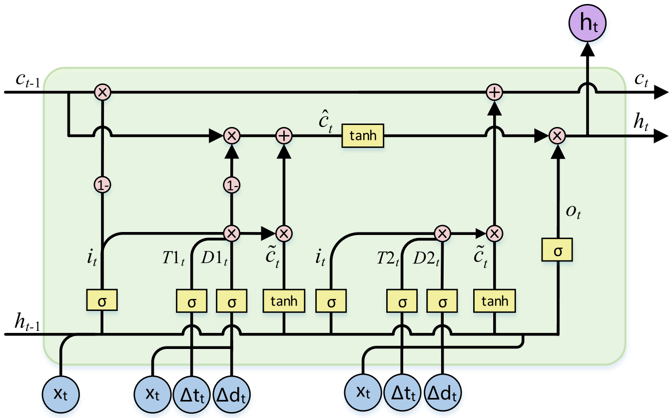


Fig. 5. A variant of STGN using coupled input and forget gates.

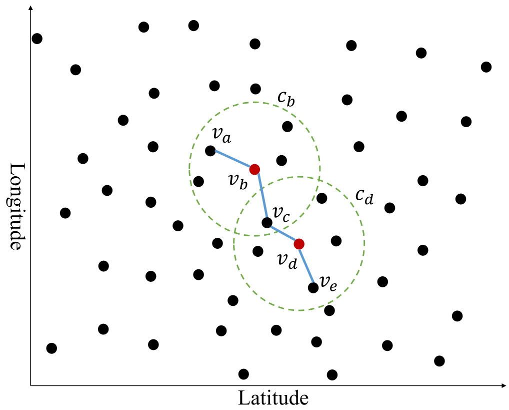


Fig. 6. Process of constructing POI context graph and sampling.

We then modify Eqs. (4), (5), and (6) to

$$\hat{c}_t = f_t \odot c_{t-1} + i_t \odot T1_t \odot D1_t \odot \tilde{c}_t, \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot T2_t \odot D2_t \odot \tilde{c}_t, \tag{12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + \triangle t_t W_{to} + \triangle d_t W_{do} + b_o), \tag{13}$$

$$h_t = o_t \odot \tanh(\hat{c}_t), \tag{14}$$

where $\triangle t_t$ is the time interval and $\triangle d_t$ is the distance interval. Besides input gate i_t, $T1_t$ can be regarded as an input information filter considering time interval, and $D1_t$ can be regarded as another input information filter considering distance interval. We add a new cell state $\hat{c}_t$ to store the result, then transfer to the hidden state h_t and finally influences next recommendation. Along this line, $\hat{c}_t$ is filtered by time gate $T1_t$ and distance gate $D1_t$ as well as input gate i_t on current recommendations.

Cell state c_t is used to memory users general interest, i.e., long-term interest. We designed a time gate and a distance gate to control the cell state c_t update. $T2_t$ first memorizes $\triangle t_t$ then transfers to c_t, further to $c_{t+1}, c_{t+2}, \ldots$. So $T2_t$ helps store $\triangle t_t$ to model user long-term interest. In the similar way, $D2_t$ memorizes $\triangle d_t$ and transfers to cell state c_t to help model user long-term interest. In this way, c_t captures user long-term interest by memorizing not only the order of user's historical visited POIs, but also the time and distance interval information between neighbor POIs. Modeling distance intervals can help capture user's general spatial interest, while modeling time intervals helps capture user's periodical visiting behavior.

Normally, a more recently visited POI with a shorter distance should have a larger influence on choosing next POI. To incorporate this knowledge in the designed gates, we add constraints $W_{x_{t1}} \leq 0$ and $W_{x_{d1}} \leq 0$ in Eqs. (7) and (9). Accordingly, if $\triangle t_t$ is smaller, $T1_t$ would be larger according to Eq. (7). In the similar way, if $\triangle d_t$ is shorter, $D1_t$ would be larger according to Eq. (9). For example, if time and distance intervals are smaller between x_t and next POI, then x_t better indicates the short-term interest, thus its influence should be increased. If $\triangle t_t$ or $\triangle d_t$ is larger, x_t would have a smaller influence on the new cell state $\hat{c}$. In this case, the short-term interest is uncertain, so we should depend more on the long-term interests. It is why we set two time gates and two distance gates to distinguish the short-term and long-term interests update.

4.2 Variation of Coupled Input and Forget Gates

Enlightened by [15], we propose another version of STGN, named STGCN, to reduce the number of parameters and improve recommendation performance. STGCN uses coupled input and forget gates instead of separately deciding what to forget and what new information to add, as shown in Fig. 5. Specifically, we remove the forget gate, and modify Eqs. (11) and (12) to

$$\hat{c}_t = (1 - i_t \odot T1_t \odot D1_t) \odot c_{t-1} \tag{15}$$

$$\begin{aligned} &+ i_t \odot T1_t \odot D1_t \odot \tilde{c}_t, \\ c_t &= (1 - i_t) \odot c_{t-1} + i_t \odot T2_t \odot D2_t \odot \tilde{c}_t. \end{aligned} \tag{16}$$

Since time gate $T1_t$ and distance gate $D1_t$ are regarded as input filters, we replace the forget gate with $(1 - i_t \odot T1_t \odot D1_t)$ in Eq. (15). $T2_t$ and $D2_t$ are used to store time intervals and distance intervals respectively, thus we use $(1 - i_t)$ in Eq. (16).

4.3 POI Context Prediction

Since graphs can represent various types of interactions and relationships, in the POI context prediction, we use the affinity graph which is widely used to encode distances among data to define POI context graph, encoding context information as affinity among instances. The recent work [44] shows promising results by leveraging unlabeled data from heterogeneous graphs. Specifically, the context information associated with the POI, i.e., the geographic information composed of the latitude and longitude pairs, is built into the POI context graphs. The POI context graph is represented by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of POIs and $\mathcal{E}$ is the set of edges between nearby POIs. Creating the edge for any two POIs is a huge project, so we set a super-parameter r in the POI context graph, drawing a circular area with the current POI as the center, and r as the radius. The POIs in the circular area are classified as the set of neighbors of the current POI, and then the current POI is connected to its neighbors to generate edges. As shown in Fig. 6, the two circles c_b and c_d are drawn according to the center POI

v_b and v_d, respectively. v_a and other POIs in the c_b are the nearby POIs of v_b while v_e and other POIs in the c_d are the nearby POIs of v_d. To make the description on the sampling process mentioned latter clear, here we just connect part of the nearby POIs. In this work, the edges between nearby POIs are unified. Furthermore, we can assign different weights to edges according to the distance between POIs to construct more complex POI graph as a future work.

In POI context prediction, our goal is to predict the context on graphs. In recent years, the Skipgram model [45] has been widely used in the embedding learning algorithms which are developed to predict context on graphs. Therefore, we apply it to establish the unsupervised loss function in Eq. (20) as a regularization. Given an instance and its context, the log loss of predicting the context is minimized by the embedding of the instance

$$\begin{aligned}\mathcal{L}_C &= -\sum_{(v_i,v_c)} \log p(v_c|v_i) \\ &= -\sum_{(v_i,v_c)} \log\left(\Psi_c^T x_i - \log\sum_{\acute{c}\in C}\exp(\Psi_{\acute{c}}^T x_i)\right),\end{aligned} \tag{17}$$

where C is the set of all possible POI contexts, and Ψ is the sets of parameters in the softmax layer on the POI context side. From the above equation we can see that when two POIs share the same context, i.e., the same neighbor POI, losses corresponding to the two POIs have exactly the same form and are minimized jointly. Accordingly, embeddings of the two must be similar in a certain way, and this suggests that minimizing the losses on all POI pairs ensures that POIs sharing more similar context will have closer embeddings. When training the Skipgram model, we use a popular negative sampling method [45] to sample (v_i, v_c, γ), where $\gamma = 1$ represents a positive pair and $\gamma = -1$ denotes negative, after we minimize the cross entropy loss of classifying the pair (v_i, v_c) to the binary label γ

$$\begin{aligned}\mathcal{L}_C = &-\mathbb{I}(\gamma = 1)\log\sigma(\Psi_c^T x_i) \\ &-\mathbb{I}(\gamma = -1)\log\sigma(-\Psi_c^T x_i),\end{aligned} \tag{18}$$

where σ is the sigmoid function, and $\mathbb{I}(\cdot)$ is an indicator function that outputs 1 when the argument is true and otherwise 0. Therefore, the unsupervised loss on the POI context side with negative sampling can be written as

$$\mathcal{L}_C = -\mathbb{E}_{(v_i,v_c,\gamma)}\log\sigma(\gamma\Psi_c^T x_i), \tag{19}$$

where the expectation is taken from the distribution $p(v_i, v_c, \gamma)$, which is conditioned on the POI context graph and encodes the distributional information in the graph structure.

The sampling process that defines the distribution $p(v_i, v_c, \gamma)$ follows [46] to deal with the real-valued edge weights in the POI context graph and combines the negative samples. Specifically, we first uniformly sample a random walk sequence in $\mathcal{G}$ which is constructed according to the radius, and then a POI is randomly sampled in all POIs and added to the path. If the sampled POI has no neighbors, another path will be created. Otherwise, the remaining neighbors of the sampled POI are randomly sampled until they form a path of length l, where l is the maximum path length. Then, we sample the positive pair with a probability p from the path with the length greater than 1, ensuring that the two POIs in the positive pair can appear simultaneously in a same window. The window size is represented as w. With probability $(1-p)$, we uniformly corrupt the context to sample a negative pair. According to above sampling process, a path whose length is 4 is formed as shown in Fig. 6. When $w = 3$, (v_a, v_d) is a positive pair.

4.4 Joint Learning

The complete loss function of NeuNext is as follows:

$$\mathcal{L} = \mathcal{L}_P + \lambda\mathcal{L}_C. \tag{20}$$

In the Eq. (20), $\mathcal{L}_P$ is the loss of the sequential preference on labeled check-in data and $\mathcal{L}_C$ is the loss of context embedding or regularization penalty applied to enforce smoothness among POIs and context. Our model is a linear combination of the objective functions for the two tasks, and the λ controls the trade-off among the two objectives.

Note that the way we jointly train our model w.r.t the two objectives is to perform SGD (stochastic gradient descent) with mini-batch Adam. First we transform H^u to $[(v_1^u, t_2^u - t_1^u, d(l_1,l_2)), (v_2^u, t_3^u - t_2^u, d(l_2,l_3)), \ldots, (v_n^u, t_q^u - t_n^u, d(l_n,l_q))]$. Then x_t in STGN is equivalent to v_t^u, $\triangle t_t$ is equivalent to $t_{t+1}^u - t_t^u$, and $\triangle d_t$ is equivalent to $d(l_{t+1}, l_t)$, where $d(\cdot,\cdot)$ is the function computing the distance between two geographical points. Moreover, we make use of all users' behavioral histories for learning and recommendation. We leverage the mini-batch learning method, and train the model on users existing histories until convergence. The model output is a probability distribution on all POIs calculated by h_t and v_t^u. And then we take a gradient step to optimize the loss based on the output and one-hot representations of v_{t+1}^u.

Then, we use Adam, a variant of Stochastic Gradient Descent(SGD), to optimize the parameters in STGN, which adapts the learning rate for each parameter by performing smaller updates for frequent parameters and larger updates for infrequent parameters. We use the projection operator described in [47] to meet the constraints $W_{t_1} \le 0$ in Eq. (7) and $W_{d_1} \le 0$ in Eq. (9). If we have $W_{t_1} > 0$ during the training process, we set $W_{t_1} = 0$. And parameter W_{d_1} is set in the same way.

Finally, we sample a batch of POI context (v_j, v_c, γ) to take another gradient step to optimize the unsupervised loss $\mathcal{L}_C$. We repeat the above process to approximate the weighting factors λ.

5 Experiment

In this section, we conduct experiments to demonstrate the effectiveness of our proposed model NeuNext for next POI recommendation on four real-world datasets. We first introduce the datasets, evaluation metrics, baseline methods and parameter settings in our experiments. Then we compare NeuNext with the state-of-the-art baseline methods, presenting the results of related analysis on our model. Finally, we study the effects of different parameters on the performance of NeuNext.

5.1 Experimental Settings

5.1.1 Datasets

We use four public LBSNs datasets that have user-POI interactions of users and locations of POIs. The statistics of the

TABLE 1
Statistics of the Four Datasets

Dataset	#user	#POI	#Check-in	Density
CA	49,005	206,097	425,691	0.004%
SIN	30,887	18,995	860,888	0.014%
Gowalla	18,737	32,510	1,278,274	0.209%
Brightkite	51,406	772,967	4,747,288	0.012%

four datasets are listed in Table 1. CA is a Foursquare dataset from users whose homes are in California, collected from January 2010 to February 2011 and used in [48]. SIN is a Singapore dataset crawled from Foursquare used by [29]. Gowalla[6] and Brightkite[7] are two widely used LBSN datasets, which have been used in many related research papers. We eliminate users with fewer than 10 check-ins and POIs visited by fewer than 10 users in the four datasets. Then, we sort each user's check-in records according to timestamp order, taking the first 70 percent as the training set, the remaining 30 percent as the testing set in next POI recommendation. Since the STGN method applied in modeling users' sequential visiting behaviors in the next POI recommendation does not involve any information about the user himself, this paper only predicts the context of POIs by utilizing the location information of POIs to establish an affinity graph and smooth the neighboring POIs.

5.1.2 Evaluation Metrics

To compare our model with baselines, we use two standard metrics Accuracy@K (Acc@K) [11] and Mean Average Precision (MAP), which are used in previous works [12], [34].

- *Accuracy@K* is the most basic and most common classification metric. It mainly measures the ratio of the number of correctly predicted samples to the total number of predicted samples. It does not consider whether the predicted samples are positive or negative. Specifically, for an instance in test data (a user will visit next POI for test), Acc@K is 1 if the visiting POI appears in top-K predicted POIs' set, 0 else. In this paper, we choose K = {1, 5, 10, 15, 20} to display different values of Acc@K.
- *MAP* is commonly used for global evaluation for ranking tasks. It's a standard metric for evaluating the quality of the whole ranked lists. The larger the MAP value, the better the performances are.

5.1.3 Baselines

We compare our model with several representative methods for next POI recommendation:

- *FPMC-LR*[2]: It combines the personalized Markov chains, which proposed in FPMC, with the user movement constraint, which specifically means moving around a localized region. It factorizes tensor of transition matrices of all users and predicts next location by computing the transition probability based on Markov chain assumption.
- *PRME-G*[3]: It utilizes the Metric Embedding method for the recommendation, which avoids drawbacks of the Matrix Factorization technique. Specifically, it embeds user and POI into the same latent space to capture the user transition patterns and then it integrates geographical influence through a simple weighing scheme. We use the settings with 60 dimensions and $\pi = 6$h as in their paper.
- *GE*[34]: It means graph-based embedding by embedding the four corresponding relational graphs (POI-POI, POI-Region, POI-Time, POI-Word) into a shared low dimensional space. The recommendation score is then calculated by a linear combination of the inner products for these contextual factors. In addition, GE dynamically computes the user's latest preferences based on the embedding of his/her checked-in POIs.
- *RNN*[49]: This is a state-of-the-art method for temporal prediction in click prediction by the use of standard RNN model. This method leverages the temporal dependency in user's behavior sequence through the recurrent structure.
- *LSTM* [43]: This is a variant of RNN model, which contains a memory cell and three multiplicative gates to allow long-term dependency learning.
- *GRU* [50]: This is a variant of RNN model, which is equipped with two gates to control the information flow.
- *PACE* [14]: It combines collaborative filtering and semi-supervised learning for POI recommendation, which jointly learns the embeddings of users and POIs to predict both user preference over POIs and various context associated with users and POIs. Only POI embedding can be jointly learned in our application scenario for having no user information.
- *ST-RNN*[12]: Based on the standard RNN model, ST-RNN replaces the single transition matrix in RNN with time-specific transition matrices and distance-specific transition matrices to model spatial and temporal contexts.
- *HST-LSTM* [11]: It introduces spatio-temporal factors into gate mechanism in LSTM to mitigate data sparsity problem. Since we do not have session information in our application scenario, we use its ST-LSTM version here.

5.1.4 Parameter Settings

For the parameters of baselines, we follow the best settings in their papers. In our model, we set its parameters to the following default values: the learning rate of the next POI recommendation task is 0.01 while that of the POI context prediction is 0.001. In addition, the radius is 0.5 and the window size is set to 3 when the path length is set to 10. The cell size and the batch size are set to 128 and 10, respectively. The important parameters, such as the cell size and the batch size in the next POI recommendation, and the window size and radius in the POI context prediction will be discussed in details in the Section 5.3.

6. http://snap.stanford.edu/data/loc-gowalla.html
7. http://snap.stanford.edu/data/loc-brightkite.html

TABLE 2
Evaluation of next POI recommendation in terms of Acc@K and MAP on four datasets

	CA				SIN			
	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP
FPMC-LR	0.0378	0.0493	0.0784	0.1791	0.0395	0.0625	0.0826	0.1724
PRME-G	0.0422	0.0650	0.0813	0.1868	0.0466	0.0723	0.0876	0.1715
GE	0.0294	0.0329	0.0714	0.1691	0.0062	0.0321	0.0607	0.1102
RNN	0.0475	0.0901	0.1138	0.1901	0.1321	0.1867	0.2043	0.2186
LSTM	0.0486	0.0937	0.1276	0.1975	0.1261	0.1881	0.2019	0.2123
GRU	0.0483	0.0915	0.1216	0.1934	0.1237	0.1921	0.1992	0.2101
PACE	0.0449	0.1010	0.1430	0.2183	0.1493	0.2311	0.2691	0.3388
ST-RNN	0.0505	0.0922	0.1232	0.2075	0.1379	0.1957	0.2091	0.2239
HST-LSTM	0.0594	0.1088	0.1372	0.2208	0.1920	0.2504	0.2794	0.3570
STGN	0.0716	0.1232	0.1508	0.2265	0.2157	0.2653	0.2954	0.3570
STGCN	0.0801	0.1308	0.1612	0.2556	0.2232	0.2737	0.3017	0.3608
NeuNext	**0.0890**	**0.1421**	**0.1748**	**0.2736**	**0.2284**	**0.2768**	**0.3022**	**0.3704**

	Gowalla				**Brightkite**			
	Acc@1	**Acc@5**	**Acc@10**	**MAP**	**Acc@1**	**Acc@5**	**Acc@10**	**MAP**
FPMC-LR	0.0293	0.0524	0.0849	0.1745	0.1634	0.2475	0.3164	0.3300
PRME-G	0.0334	0.0652	0.0869	0.1916	0.1976	0.2993	0.3495	0.3115
GE	0.0174	0.0600	0.0947	0.1973	0.0521	0.1376	0.2118	0.2602
RNN	0.0473	0.0892	0.1207	0.1998	0.3401	0.4087	0.4320	0.4130
LSTM	0.0503	0.0967	0.1241	0.2004	0.3575	0.4146	0.4489	0.4303
GRU	0.0498	0.0931	0.1289	0.2045	0.3310	0.4007	0.4377	0.4042
PACE	0.0364	0.1070	0.1568	0.1891	0.3060	0.4244	0.4684	0.5483
ST-RNN	0.0519	0.0953	0.1304	0.2187	0.3672	0.4231	0.4477	0.4369
HST-LSTM	0.0702	0.1366	0.1676	0.2414	0.4336	0.4783	0.4999	0.5476
STGN	0.0835	0.1522	0.1879	0.2443	0.4389	0.4807	0.5035	0.5266
STGCN	**0.0933**	0.1644	0.2020	0.2557	0.4443	0.4953	0.5231	0.5626
NeuNext	0.0930	**0.1689**	**0.2034**	**0.2685**	**0.4567**	**0.5109**	**0.5422**	**0.5683**

5.2 Results and Discussions

5.2.1 Comparison With Baselines

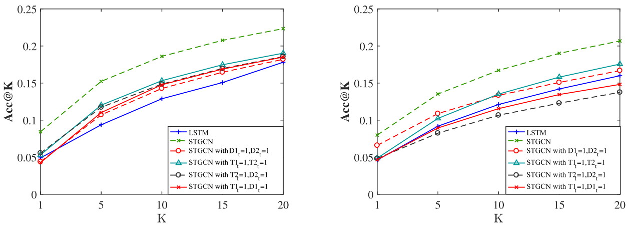
The performance of all methods on four datasets evaluated by Acc@K and MAP is shown in Table 2. From the experimental results, we have the following observations.

RNN performs better than Markov chain method FPMC-LR and embedding method PRME-G, due to the use of deep neural networks, which shows the ability of neural networks on modeling users general tastes and their sequential behaviors. In addition, the result of GE is not good for missing social and textual information in our datasets.

LSTM and GRU, the variants of RNN model, slightly improve the performance compared with RNN because of their advantages in modeling long-term interests. ST-RNN which models spatial and temporal contexts with time and distance performs better than LSTM and GRU. This indicates that it is necessary to consider spatial and temporal information when predicting the next POI. However, the performance of ST-RNN is still close to the standard RNN method, which may be caused by the difficulty of manually setting the windows of time and distance intervals. HST-LSTM performs better than ST-RNN because it introduces spatio-temporal factors into the gate mechanism. It proves the effectiveness of the idea of combining spatial-temporal factors with the gate mechanism. PACE performs competitive, especially on relatively sparse datasets, which shows the feasibility of combining collaborative filtering with semi-supervised learning and the effectiveness of jointly learning POI embeddings.

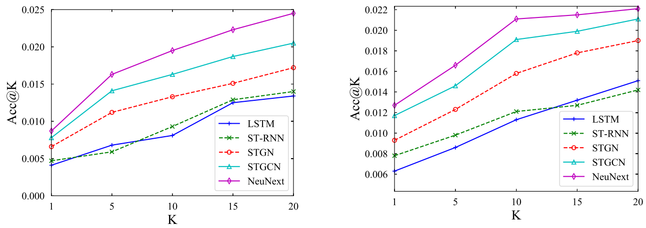
STGN and STGCN all perform significantly better than other baselines evaluated here on the four datasets in all metrics. Specifically, STGCN outperforms the Markov chain based methods considerably by a large margin. In addition, STGCN outperforms PACE. This may be because PACE ignores the sequential pattern and time intervals between user behaviors, which shows the importance of sequence information, and also reflects that time intervals cannot be learned through context prediction. Moreover, STGCN consistently outperforms five RNN-based methods: RNN, LSTM, GRU, ST-RNN, and HST-LSTM. The performance gains provided by STGCN over these five counterparts are about 34.8 - 68.6, 16.3 - 80.0, 32.9 - 97.3 and 2.5 - 34.2 percent in terms of Acc@1 metric on CA, SIN, Gowalla, and Brightkite respectively. The significant improvement indicates that the mechanism to model temporal and spatial contexts in STGCN can better catch the user's sequential behaviors and is effective for the task of next POI recommendation. This is because we add time and distance gates to integrate time and distance intervals into the model. Moreover, STGCN always performs better than STGN. This may be because STGCN has fewer parameters than STGN, which alleviates the complexity and over-fitting problems of STGN caused by too many parameters.

In general, our NeuNext performs the best among all methods on the four datasets. Specifically, NeuNext achieves MAP gains of 7.1, 2.7, 5.0 and 1.0 percent in CA, SIN, Gowalla, Brightkite respectively, which demonstrates the efficacy of the joint learning framework in NeuNext. Note



(a) Gowalla - Acc@K (b) CA - Acc@K

Fig. 7. Performances with different time and distance gates in STGCN.



(a) Gowalla - Acc@K (b) BrightKite - Acc@K

Fig. 8. Performances for cold start users on two datasets.

that, NeuNext performs better on CA than on the other three datasets, comparing with STGCN, probably because CA is relatively sparse. As we introduced before, the SSL framework can effectively alleviate the problem of data sparsity by utilizing the labeled and unlabeled data together. The task of next POI recommendation with labeled data accurately models the user's sequential sequential preference, while the SSL framework effectively incorporates the POI context knowledge from the vast unlabeled data through joint learning. Therefore, it is reasonable that the improvement of NeuNext is larger on sparser datasets.

5.2.2 Effectiveness of Time and Distance Gates

There are two time gates and two distance gates in STGCN model. We first investigate the effectiveness of time and distance gates on modeling time and distance intervals. Specifically, we set $D1_t = 1$ and $D2_t = 1$, in Eqs. (9) and (10), respectively. That is, we close two distance gates and only consider the time intervals. Similarly, we set $T1_t = 1$ and $T2_t = 1$, in Eqs. (7) and (8), respectively. That is, we close two time gates and only consider distance information. From Fig. 7, we can see that the time gates and distance gates have similar importance on both datasets (i.e., Gowalla and CA). Moreover, they both are critical for improving the recommendation performances.

We also investigate the effectiveness of time and distance gates on modeling short-term and long-term interests. We set $T2_t = 1$ and $D2_t = 1$, in Eqs. (8) and (10), which means we close time and distance gates on long-term interests and only activate time and distance gates on short-term interest. Similarly, we set $T1_t = 1$ and $D1_t = 1$, in Eqs. (7) and (9), which means we close time and distance gates for short-term interest. As shown in Fig. 7, we can observe that they all perform worse than original STGCN, which means that time and distance intervals are not only critical to short-term interests but also important to long-term interests. Distance intervals may help model user general spatial preference and time intervals may help to model user long-term periodical behavior.

5.2.3 Performance of Cold Start

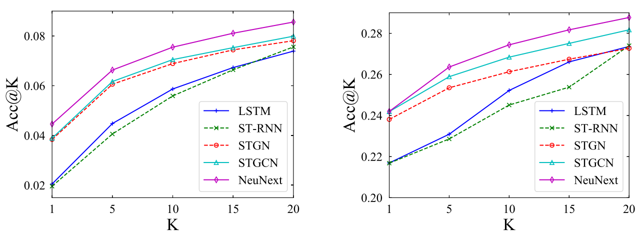
Taking Acc@K as the measure metric, we also evaluate the performance of NeuNext on two datasets (i.e., Gowalla and BrightKite) by comparing with other next POI recommendation competitors for cold-start users and cold-start POIs. On the one hand, if a user just visits a few POIs in the datasets, which means we can hardly learn user preference on POIs, we think the user is a cold case. Specifically, we take users with less than 5 check-ins as a cold user in our experiments. As shown in Fig. 8, we can observe that STGCN and STGN perform much better than the other two RNN-based methods under the cold start scenario, probably because they can model long-term interests as well as short-term interests with considering time and distance intervals. Furthermore, NeuNext performs the best among all methods, which proves that joint training using both unlabeled data and labeled data can effectively alleviate data sparsity.

On the other hand, we select POIs that have been interacted by more than 40 users in the training data and by at least 1 user in the test data as cold-start POIs [51]. Then, we filter out the interactions related to cold-start POIs in the training data, and ensure each user's interaction records in the training data are ordered chronologically. Fig. 9 shows that NeuNext performs better than other next POI recommendation models under cold-start POIs setting. This is because the cold-start POIs that appeared for the first time in the test is a new POI for the user. The other next POI recommendation models do not have any collaborative information to use, while NeuNext can use geographic information to find the nearby POIs of the cold-start POIs through the POI context prediction and mitigate the lack of collaborative information.

5.3 Impact of Parameters

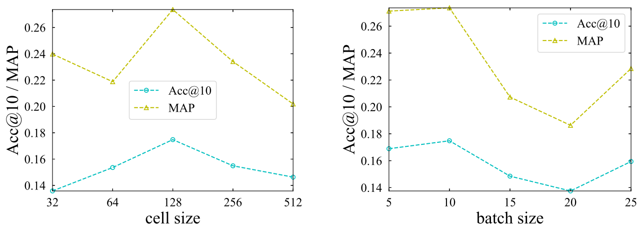
5.3.1 Impact of Cell Size and Batch Size

In the standard RNN, different cell sizes and batch sizes may lead to different performances. We investigate the impact of these two parameters for NeuNext. We vary cell sizes and batch sizes to observe the performance of our proposed model in terms of Acc@10 and MAP. We only show the impact of the two parameters on CA dataset due to space constraint. As shown in Fig. 10, the proper cell size can help achieve the best performance, and the smaller batch size has better results than the larger ones. The cell size determines the model complexity, a small cell size is



(a) Gowalla - Acc@K (b) BrightKite - Acc@K

Fig. 9. Performances for cold start POIs on two datasets.



(a) Different Cell Size (b) Different Batch Size

Fig. 10. Performances with different cell sizes and batch sizes on CA.

TABLE 3
The Performance of NeuNext With Varying r and w on CA

Acc@10 \ r / w	$r = 0.1$	$r = 0.5$	$r = 1.0$	$r = 1.5$
w =2	0.1658	0.1720	0.1589	0.1422
w =3	0.1536	0.1748	0.1662	0.1516
w =4	0.1644	0.1559	0.1544	0.1584
w =5	0.1437	0.1576	0.1631	0.1631

not sufficient to model user preference while a large one may make it too complicated to cause over-fitting. Moreover, a big batch size may lead to insufficient updating of parameters in our model.

5.3.2 *Impact of Radius and Window Size*

We apply a grid search over combination of varying $r \in \{0.1, 0.5, 1.0, 1.5\}$ and $w \in \{2, 3, 4, 5\}$ for best results. The results are shown in Table 3. When setting $r = 0.5$ and $w = 3$, the proposed model NeuNext achieves the best performance in terms of Acc@10. The radius determines not only the neighbors of the POI but also the creation of the edges in the process of constructing the POI context graph. We can observe that a smaller or larger radius can not get the best performance, which may be because two POIs that are close in distance are not related, and the number of neighbors of the POI increases as the radius increases, which may cause more unrelated POIs to affect the context prediction of the graph. Furthermore, the window size is related to the context sampling on graphs, and determines the sampling range of the positive sample pair. A larger window size allows two POIs that are farther away to form a positive pair, while a smaller window selects two close POIs to make up a positive pair. So the appropriate window size is also important.

6 Conclusion

This paper proposed a joint learning approach NeuNext for next POI recommendation. NeuNext is a SSL framework that consists of two parts: next POI recommendation and POI context prediction. In the next POI recommendation, a new enhanced long-short term memory method was proposed to model users' short-term and long-term sequential preferences through two pairs of time gates and distance gates, respectively. In the POI context prediction, each POI was predicted with a set of context POIs through affinity graph with Skipgram corresponding to all the POIs. The two tasks were jointly learned through shared POI embedding which makes the proposed model able to fully leverage labeled check-in data and unlabeled context data to improve the recommendation effectiveness and thus alleviates the data sparsity issue. We conducted extensive experiments on real-world datasets from various location-based social networks and our experimental results demonstrated the effectiveness of NeuNext.

Acknowledgments

This research was supported by NSFC (No. 61876117, 6187-6217, 61772356, 61728205, 61972069, 61872258) and PAPD.

References

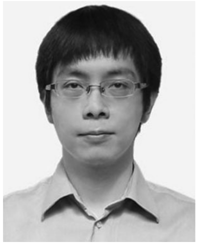
[1] S. Feng, G. Cong, B. An, and Y. M. Chee, "POI2Vec: Geographical latent representation for predicting future visitors," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 102–108.
[2] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, vol. 13, pp. 2605–2611.
[3] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 2069–2075.
[4] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung, "Inferring a personalized next point-of-interest recommendation model with latent behavior patterns," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 137–143.
[5] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *Proc. VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.
[6] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newslett.*, vol. 9, no. 2, pp. 75–79, 2007.
[7] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong, "Unified point-of-interest recommendation with temporal interval assessment," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1015–1024.
[8] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 811–820.
[9] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016.
[10] Y. Zhu *et al.*, "What to do next: Modeling user behaviors by time-LSTM," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3602–3608.
[11] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2341–2347.
[12] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 194–200.
[13] D. Jannach, L. Lerche, and M. Jugovac, "Adaptation and evaluation of recommendations for short-term shopping goals," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 211–218.
[14] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 1245–1254.
[15] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[16] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, vol. 1, pp. 1257–1264.
[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[18] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 426–434.
[19] L. Wu, E. Chen, Q. Liu, L. Xu, T. Bao, and L. Zhang, "Leveraging tagging for neighborhood-aware probabilistic matrix factorization," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1854–1858.

[20] R. P. Adams, G. E. Dahl, and I. Murray, "Incorporating side information in probabilistic matrix factorization with Gaussian processes," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 1–9.
[21] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proc. SIAM Int. Conf. Data Mining*, 2010, pp. 199–210.
[22] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proc. 18th SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2010, pp. 458–461.
[23] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 325–334.
[24] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel, "LARS: A location-aware recommender system," in *Proc. IEEE 28th Int. Conf. Data Eng.*, 2012, pp. 450–461.
[25] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura, "Geo topic model: Joint modeling of user's activity area and interests for location recommendation," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 375–384.
[26] B. Liu, Y. Fu, Z. Yao, and H. Xiong, "Learning geographical preferences for point-of-interest recommendation," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1043–1051.
[27] J.-D. Zhang and C.-Y. Chow, "iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2013, pp. 334–343.
[28] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 831–840.
[29] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2013, pp. 363–372.
[30] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 93–100.
[31] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, vol. 12, pp. 17–23.
[32] D. Lian, V. W. Zheng, and X. Xie, "Collaborative filtering meets next check-in location prediction," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 231–232.
[33] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 315–322.
[34] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 15–24.
[35] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1235–1244.
[36] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 627–636.
[37] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 278–288.
[38] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 353–362.
[39] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 791–798.
[40] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A neural autoregressive approach to collaborative filtering," *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 764–773.
[41] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 111–112.
[42] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, 2017, Art. no. 36.
[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[44] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in *Proc. Int. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2010, pp. 570–586.
[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[46] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 40–48.
[47] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1571–1578.
[48] H. Gao, J. Tang, and H. Liu, "gSCorr: Modeling geo-social correlations for new check-ins on location-based social networks," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1582–1586.
[49] Y. Zhang *et al.*, "Sequential click prediction for sponsored search with recurrent neural networks," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1369–1375.
[50] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.
[51] S.-T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naïve filterbots for robust cold-start recommendations," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 699–705.

Pengpeng Zhao received the PhD degree in computer science from Soochow University, China, in 2008. He is a professor with the School of Computer Science and Technology, Soochow University, China. From 2016 to 2017, he was a visiting scholar, working with the Data Mining and Business Analysis Laboratory, Rutgers University, Piscataway, New Jersey. He has published more than 60 papers in prestigious international conferences and journals, including ACM MM, AAAI, IJCAI, ICDM, CIKM, DASFAA, and ICME. He was a program committee member of international conferences, such as AAAI, IJCAI, CIKM, and PAKDD. His current research interests include data mining, deep learning, big data analysis, and recommender systems.

Anjing Luo is currently working toward the MS degree in the School of Computer Science and Technology, Soochow University, Suzhou, China. Her main research interests are spatial data processing, recommender systems, and data mining.

Yanchi Liu received the PhD degree in information technology from Rutgers, the State University of New Jersey and the PhD degree in management science and engineering from the University of Science and Technology Beijing, China. He is a researcher in NEC Labs America. His research interests include data mining, business intelligence, urban computing, and recommender systems. He has published in the *IEEE Transactions on Cybernetics*, the *ACM Transactions on Intelligent Systems and Technology*, KDD, IJCAI, ICDM etc.



Jiajie Xu received the master's degree from the University of Queensland, Australia, in 2006, and the PhD degree from the Swinburne University of Technology, Australia, in 2011, and then worked in the Institute of Software, Chinese Academy of Sciences as assistant professor before joining Soochow University, China. He is an associate professor of Soochow University, China, and a member of the advanced data analytics (ADA) research center.

Zhixu Li received the bachelor's and master's degrees from the Renmin University of China, China, in 2006 and 2009 respectively, and the PhD degree on computer science from the University of Queensland, Australia, in 2013. He is an associate professor with the School of Computer Science and Technology, Soochow University, China, and a member of the Advanced Data Analytics (ADA) research center. Before joining Soochow University, China, he has worked at King Abdullah University of Science and Technology, Saudi Arabia as a postdoc fellow from 2013 to 2014.

Fuzhen Zhuang is an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences, China. His research interests include transfer learning, machine learning, data mining, multi-task learning, and recommendation systems. He has published more than 80 papers in some prestigious refereed journals and conference proceedings, such as the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Cybernetics*, the *IEEE Transactions on Neural Networks and Learning Systems*, the *ACM Transactions on Intelligent Systems and Technology*, the *Information Sciences*, the *Neural Networks*, SIGKDD, IJCAI, AAAI, WWW, ICDE, ACM CIKM, ACM WSDM, SIAM SDM, and IEEE ICDM.

Victor S. Sheng (Senior Member, IEEE) received the master's degree in computer science from the University of New Brunswick, Canada, in 2003, and the PhD degree in computer science from Western University, Ontario, Canada, in 2007. He is an associate professor of computer science at Texas Tech University, Lubbock, Texas, and the founding director of Data Analytics Lab (DAL). His research interests include data mining, machine learning, crowdsourcing, and related applications in business, industry, medical informatics, and software engineering. He is a lifetime member of ACM, a SPC and PC member for many international conferences.

Xiaofang Zhou (Fellow, IEEE) received the bachelor's and master's degrees in computer science from Nanjing University, China, in 1984 and 1987, respectively, and the PhD degree in computer science from the University of Queensland, Australia, in 1994. He is a professor of computer science with the University of Queensland, Australia. He is the head of the Data and Knowledge Engineering Research Division, School of Information Technology and Electrical Engineering. He is also a specially appointed adjunct professor with Soochow University, China. His research is focused on finding effective and efficient solutions to managing integrating, and analyzing very large amounts of complex data for business and scientific applications. His research interests include spatial and multimedia databases, high performance query processing, Web information systems, data mining, and data quality management.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.