

Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts

Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan

Center for Research on Intelligent Perception and Computing
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{qiang.liu, shu.wu, wangliang, tnt}@nlpr.ia.ac.cn

Abstract

Spatial and temporal contextual information plays a key role for analyzing user behaviors, and is helpful for predicting where he or she will go next. With the growing ability of collecting information, more and more temporal and spatial contextual information is collected in systems, and the location prediction problem becomes crucial and feasible. Some works have been proposed to address this problem, but they all have their limitations. Factorizing Personalized Markov Chain (FPMC) is constructed based on a strong independence assumption among different factors, which limits its performance. Tensor Factorization (TF) faces the cold start problem in predicting future actions. Recurrent Neural Networks (RNN) model shows promising performance comparing with FPMC and TF, but all these methods have problem in modeling continuous time interval and geographical distance. In this paper, we extend RNN and propose a novel method called Spatial Temporal Recurrent Neural Networks (ST-RNN). ST-RNN can model local temporal and spatial contexts in each layer with time-specific transition matrices for different time intervals and distance-specific transition matrices for different geographical distances. Experimental results show that the proposed ST-RNN model yields significant improvements over the competitive compared methods on two typical datasets, i.e., Global Terrorism Database (GTD) and Gowalla dataset.

Introduction

With the rapid growth of available information on the internet and the enhancing ability of systems in collecting information, more and more temporal and spatial contexts have been collected. Spatial and temporal contexts describe the essential factors for an event, i.e., where and when. These factors are fundamental for modeling behavior in practical applications. It is challenging and crucial to predict where a man will be at a give time point with complex temporal and spatial information. For example, based on user historical check-in data, we can analyze and predict where a user will go next. Moreover, such analysis can also be used for social good, such as predicting where traffic jams will happen or which city terrorist organizations will attack.

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Nowadays, the spatial temporal predicting problem has been extensively studied. Factorizing Personalized Markov Chain (FPMC) (Rendle, Freudenthaler, and Schmidt-Thieme 2010) is a personalized extension of common markov chain models, and has become one of the most popular methods for sequential prediction. FPMC has also been applied for next location prediction (Cheng et al. 2013; Chen, Liu, and Yu 2014). The main concern about FPMC is that it is based on a strong independence assumption among different factors. As another popular method, Tensor Factorization (TF) has been successfully applied for time-aware recommendation (Xiong et al. 2010) as well as modeling spatial temporal information (Zheng et al. 2010a; Bahadori, Yu, and Liu 2014). In TF, both time bins and locations are regarded as additional dimensions in the factorized tensor, which leads the cold start problem in behavior prediction with new time bins, e.g., behaviors in the future. Recently, Recurrent Neural Networks (RNN) has been successfully employed for word embedding (Mikolov et al. 2010; 2011a; 2011b) and sequential click prediction (Zhang et al. 2014). RNN shows promising performance comparing with conventional methods.

Although the above methods have achieved satisfactory results in some applications, they are unable to handle continuous geographical distances between locations and time intervals between nearby behaviors in modeling sequential data. *At first*, these continuous values of spatial and temporal contexts are significant in behavior modeling. For instance, a person may tend to go to a restaurant nearby, but he or she maybe hesitate to go to a restaurant far away even if it is delicious and popular. Meanwhile, suppose a person went to an opera house last night and a parking lot last month, where he or she will go to today has a higher probability to be influenced by the opera house because of the similar interests and demands in a short period. *Secondly*, these local temporal contexts have fundamental effects in revealing characteristics of the user and are helpful for the behavior modeling. If the person went to an opera house last night and an art museum this morning, it is probable that both the opera house and the museum have higher important than other contexts, e.g., going to shopping mall last month, in the next location prediction. *Finally*, as some behaviors are periodical such as going to church every Sunday, the effect of time interval becomes important for temporal prediction in such situations.

In this paper, to better model spatial and temporal information, we propose a novel method called Spatial Temporal Recurrent Neural Networks (ST-RNN). Rather than considering only one element in each layer of RNN, taking local temporal contexts into consideration, ST-RNN models sequential elements in an almost fixed time period in each layer. Besides, ST-RNN utilizes the recurrent structure to capture the periodical temporal contexts. Therefore, ST-RNN can well model not only the local temporal contexts but also the periodical ones. On the other hand, ST-RNN employs the time-specific and distance-specific transition matrices to characterize dynamic properties of continuous time intervals and geographical properties of distances, respectively. Since it is difficult to estimate matrices for continuous time intervals and geographical distances, we divide the spatial and temporal values into discrete bins. For a specific temporal value in one time bin, we can calculate the corresponding transition matrix via a linear interpolation of transition matrices of the upper bound and lower bound. Similarly, for a specific spatial value, we can generate the transition matrix. Incorporating the recurrent architecture with continuous time interval and location distance, ST-RNN can better model spatial temporal contexts and give more accurate location prediction.

The main contributions of this work are listed as follows:

- We model time intervals in a recurrent architecture with time-specific transition matrices, which presents a novel perspective on temporal analysis.
- We incorporate distance-specific transition matrices for modeling geographical distances, which promotes the performance of spatial temporal prediction in a recurrent architecture.
- Experiments conducted on real-world datasets show that ST-RNN is effective and clearly outperforms the state-of-the-art methods.

Related Work

In this section, we review several types of methods for spatial temporal prediction including factorization methods, neighborhood based methods, markov chain based methods and recurrent neural networks.

Matrix Factorization (MF) based methods (Mnih and Salakhutdinov 2007; Koren, Bell, and Volinsky 2009) have become the state-of-the-art approaches to collaborative filtering. The basic objective of MF is to factorize a user-item rating matrix into two low rank matrices, each of which represents the latent factors of users or items. The original matrix can be approximated via the multiplying calculation. MF has been extended to be time-aware and location-aware nowadays. Tensor Factorization (TF) (Xiong et al. 2010) treats time bins as another dimension and generate latent vectors of users, items and time bins via factorization. And timeSVD++ (Koren 2010; Koenigstein, Dror, and Koren 2011) extends SVD++ (Koren 2008) in the same way. Rather than temporal information, spatial information can also be modeled via factorization models, such as tensor factorization (Zheng et al. 2010a) and collective matrix factorization (Zheng et al. 2010b). Moreover, temporal

and spatial information can be included in TF simultaneously as two separated dimensions and make location prediction (Bahadori, Yu, and Liu 2014; Bhargava et al. 2015; Zhong et al. 2015). However, it is hard for factorization based models to generate latent representations of time bins that have never or seldom appeared in the training data. Thus, we can say that, it is hard to predict future behaviors with factorization based models.

Neighborhood based models might be the most natural methods for prediction with both temporal and spatial contexts. Time-aware neighborhood based methods (Ding and Li 2005; Nasraoui et al. 2007; Lathia, Hailes, and Capra 2009; Liu et al. 2010) adapt the ordinary neighborhood based algorithms to temporal effects via giving more relevance to recent observations and less to past observations. And for spatial information, the distance between locations is calculated and the prediction is made based on power law distribution (Ye et al. 2011) or the multi-center gaussian model (Cheng et al. 2012). Recently, some work considers users' interest to the neighborhood of the destination location (Liu et al. 2014; Li et al. 2015). And Personalized Ranking Metric Embedding (PRME) method (Feng et al. 2015) learns embeddings as well as calculating the distance between destination location and recent visited ones. However, neighborhood based methods are unable to model the underlying properties in users' sequential behavior history.

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, a estimated transition matrix indicates the probability of a behavior based on the past behaviors. Extending MC via factorization of the probability transition matrix, Factorizing Personalized Markov Chain (FPMC) (Rendle, Freudenthaler, and Schmidt-Thieme 2010) has become a state-of-the-art method. It has also been extended by generating user groups (Natarajan, Shin, and Dhillon 2013), modeling interest-forgetting curve (Chen, Wang, and Wang 2015) and capturing dynamic of boredom (Kapoor et al. 2015). Recently, rather than merely modeling temporal information, FPMC is successfully applied in the spatial temporal prediction by using location Constraint (Cheng et al. 2013) or combining with general MC methods (Chen, Liu, and Yu 2014). However, FPMC assumes that all the component are linearly combined, indicating that it makes strong independent assumption among factors (Wang et al. 2015).

Recently, Recurrent Neural Networks (RNN) not only has been successfully applied in word embedding for sentence modeling (Mikolov et al. 2010; 2011a; 2011b), but also shows promising performance for sequential click prediction (Zhang et al. 2014). RNN consists of an input layer, an output unit and multiple hidden layers. Hidden representation of RNN can change dynamically along with a behavioral history. It is a suitable tool for modeling temporal information. However, in modeling sequential data, RNN assumes that temporal dependency changes monotonously along with the position in a sequence. This does not confirm to some real situations, especially for the most recent elements in a historical sequence, which means that RNN can not well model local temporal contexts. Moreover,

though RNN shows promising performance comparing with the conventional methods in sequential prediction, it is not capable to model the continuous geographical distance between locations and time interval between behaviors.

Proposed Model

In this section, we first formulate our problem and introduce the general RNN model, and then detail our proposed ST-RNN model. Finally we present the learning procedure of the proposed model.

Problem Formulation

Let P be a set of users and Q be a set of locations, $\mathbf{p}_u \in \mathbb{R}^d$ and $\mathbf{q}_v \in \mathbb{R}^d$ indicate the latent vectors of user u and location v . Each location v is associated with its coordinate $\{x_v, y_v\}$. For each user u , the history of where he has been is given as $Q^u = \{q_{t_1}^u, q_{t_2}^u, \dots\}$, where $q_{t_i}^u$ denotes where user u is at time t_i . And the history of all users is denoted as $Q^U = \{Q^{u_1}, Q^{u_2}, \dots\}$. Given historical records of a users, the task is to predict where a user will go next at a specific time t .

Recurrent Neural Networks

The architecture of RNN consists of an input layer, an output unit, hidden layers, as well as inner weight matrices (Zhang et al. 2014). The vector representation of the hidden layers are computed as:

$$\mathbf{h}_{t_k}^u = f \left(\mathbf{M}\mathbf{q}_{t_k}^u + \mathbf{C}\mathbf{h}_{t_{k-1}}^u \right), \quad (1)$$

where $\mathbf{h}_{t_k}^u$ is the representation of user u at time t_k , $\mathbf{q}_{t_k}^u$ denotes the latent vector of the location the user visits at time t_k , \mathbf{C} is the recurrent connection of the previous status propagating sequential signals and \mathbf{M} denotes the transition matrix for input elements to capture the current behavior of the user. The activation function $f(x)$ is chosen as a *sigmoid* function $f(x) = \exp(1/(1 + e^{-x}))$.

RNN With Temporal Context

Since long time intervals have different impacts comparing with short ones, the length of time interval is essential for predicting future behaviors. But continuous time intervals can not be modeled by the current RNN model. Meanwhile, since RNN can not well model local temporal contexts in user behavioral history, we need more subtle processing for the most recent elements in a behavioral history. Accordingly, it will be reasonable and plausible to model more elements of local temporal contexts in each layer of the recurrent structure and take continuous time intervals into consideration. Thus, we replace the transition matrix \mathbf{M} in RNN with time-specific transition matrices. Mathematically, given a user u , his or her representation at time t can be calculated as:

$$\mathbf{h}_t^u = f \left(\sum_{q_{t_i}^u \in Q^u, t-w < t_i < t} \mathbf{T}_{t-t_i} \mathbf{q}_{t_i}^u + \mathbf{C}\mathbf{h}_{t-w}^u \right), \quad (2)$$

where w is the width of time window and the elements in this window are modeled by each layer of the model, \mathbf{T}_{t-t_i}

denotes the time-specific transition matrix for the time interval $t - t_i$ before current time t . The matrix \mathbf{T}_{t-t_i} captures the impact of elements in the most recent history and takes continuous time interval into consideration.

Spatial Temporal Recurrent Neural Networks

Conventional RNN have difficulty in modeling not only the time interval information, but also the geographical distance between locations. Considering distance information is an essential factor for location prediction, it is necessary to involve it into our model. Similar to time-specific transition matrices, we incorporate distance-specific transition matrices for different geographical distances between locations. Distance-specific transition matrices capture geographical properties that affect human behavior. In ST-RNN, as shown in Figure 1, given a user u , his or her representation at time t can be calculated as:

$$\mathbf{h}_{t, q_t^u}^u = f \left(\sum_{q_{t_i}^u \in Q^u, t-\hat{w} < t_i < t} \mathbf{S}_{q_t^u - q_{t_i}^u} \mathbf{T}_{t-t_i} \mathbf{q}_{t_i}^u + \mathbf{C}\mathbf{h}_{t-\hat{w}, q_{t-\hat{w}}^u}^u \right), \quad (3)$$

where $\mathbf{S}_{q_t^u - q_{t_i}^u}$ is the distance-specific transition matrix for the geographical distance $q_t^u - q_{t_i}^u$ according to the current coordinate, and $q_{t_i}^u$ denotes the coordinate of user u at time t . The geographical distance can be calculated as an Euclidean distance:

$$q_t^u - q_{t_i}^u := \|x_t^u - x_{t_i}^u, y_t^u - y_{t_i}^u\|_2. \quad (4)$$

Usually, the location q_{t-w}^u , i.e., the location user u visits at time $t - w$, does not exist in the visiting history Q^u . We can utilize the approximate value \hat{w} as the local window width. Based on the visiting list Q^u and the time point $t - w$, we set the value \hat{w} to make sure that \hat{w} is the most closed value to w and $q_{t-\hat{w}}^u \in Q^u$. Thus, \hat{w} is usually a slightly larger or small than w .

Moreover, when the history is not long enough or the predicted position is at the very first part of the history, we have $t < w$. Then, Equation 3 should be rewritten as:

$$\mathbf{h}_{t, q_t^u}^u = f \left(\sum_{q_{t_i}^u \in Q^u, 0 < t_i < t} \mathbf{S}_{q_t^u - q_{t_i}^u} \mathbf{T}_{t-t_i} \mathbf{q}_{t_i}^u + \mathbf{C}\mathbf{h}_0^u \right), \quad (5)$$

where $\mathbf{h}_0^u = \mathbf{h}_0$ denotes the initial status. The initial status of all the users should be the same because there does not exist any behavioral information for personalised prediction in such situations.

Finally, the prediction of ST-RNN can be yielded via calculating inner product of user and item representations. The prediction of whether user u would go to location v at time t can be computed as:

$$o_{u,t,v} = (\mathbf{h}_{t, q_t^u}^u + \mathbf{p}_u)^T \mathbf{q}_v, \quad (6)$$

where \mathbf{p}_u is the permanent representation of user u , indicating his or her interest and activity range, and $\mathbf{h}_{t, q_t^u}^u$ captures his or her dynamic interests under the specific spatial and temporal contexts.

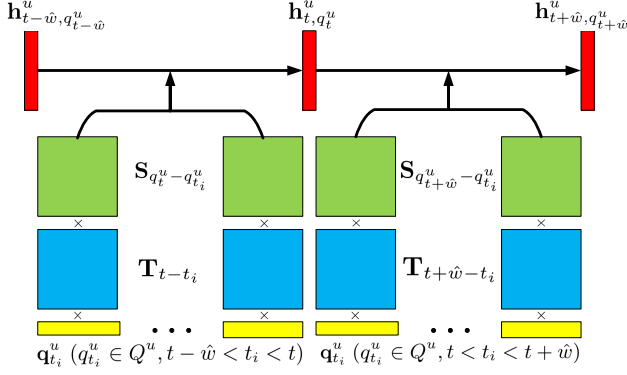


Figure 1: Overview of proposed ST-RNN model.

Linear Interpolation for Transition Matrices

If we learn a distinct matrix for each possible continuous time intervals and geographical distances, the ST-RNN model will face the data sparsity problem. Therefore, we partition time interval and geographical distance into discrete bins respectively. Only the transition matrices for the upper and lower bound of the corresponding bins are learned in our model. For the time interval in a time bin or geographical distance in a distance bin, their transition matrices can be calculated via a linear interpolation. Mathematically, the time-specific transition matrix T_{t_d} for time interval t_d and the distance-specific transition matrix S_{l_d} for geographical distance l_d can be calculated as:

$$\mathbf{T}_{t_d} = \frac{[\mathbf{T}_{L(t_d)}(U(t_d) - t_d) + \mathbf{T}_{U(t_d)}(t_d - L(t_d))]}{[(U(t_d) - t_d) + (t_d - L(t_d))]}, \quad (7)$$

$$\mathbf{S}_{l_d} = \frac{[\mathbf{S}_{L(l_d)}(U(l_d) - l_d) + \mathbf{S}_{U(l_d)}(l_d - L(l_d))]}{[(U(l_d) - l_d) + (l_d - L(l_d))]}, \quad (8)$$

where $U(t_d)$ and $L(t_d)$ denote the upper bound and lower bound of time interval t_d , and $U(l_d)$ and $L(l_d)$ denote the upper bound and lower bound of geographical distance l_d respectively. Such a linear interpolation method can solve the problem of learning transition matrices for continuous values and provide a solution for modeling the impact of continuous temporal and spatial contexts.

Parameter Inference

In this subsection, we introduce the learning process of ST-RNN with Bayesian Personalized Ranking (BPR) (Rendle et al. 2009) and Back Propagation Through Time (BPTT) (Rumelhart, Hinton, and Williams 1988).

BPR (Rendle et al. 2009) is a state-of-the-art pairwise ranking framework for the implicit feedback data. The basic assumption of BPR is that a user prefers a selected location than a negative one. Then, we need to maximize the following probability:

$$p(u, t, v \succ v') = g(o_{u,t,v} - o_{u,t,v'}), \quad (9)$$

where v' denotes a negative location sample, and $g(x)$ is a nonlinear function which is selected as $g(x) = 1/(1 + e^{-x})$. Incorporating the negative log likelihood, we can solve the following objective function equivalently:

$$J = \sum \ln(1 + e^{-(o_{u,t,v} - o_{u,t,v'})}) + \frac{\lambda}{2} \|\Theta\|^2, \quad (10)$$

where $\Theta = \{P, Q, S, T, C\}$ denotes all the parameters to be estimated, λ is a parameter to control the power of regularization. And the derivations of J with respect to the parameters can be calculated as:

$$\frac{\partial J}{\partial \mathbf{p}_u} = \sum \frac{(\mathbf{q}_{v'} - \mathbf{q}_v) e^{-(o_{u,t,v} - o_{u,t,v'})}}{1 + e^{-(o_{u,t,v} - o_{u,t,v'})}} + \lambda \mathbf{p}_u,$$

$$\frac{\partial J}{\partial \mathbf{q}_v} = - \sum \frac{(\mathbf{h}_{t,q_v}^u + \mathbf{p}_u) e^{-(o_{u,t,v} - o_{u,t,v'})}}{1 + e^{-(o_{u,t,v} - o_{u,t,v'})}} + \lambda \mathbf{q}_v,$$

$$\frac{\partial J}{\partial \mathbf{q}_{v'}} = \sum \frac{(\mathbf{h}_{t,q_{v'}}^u + \mathbf{p}_u) e^{-(o_{u,t,v} - o_{u,t,v'})}}{1 + e^{-(o_{u,t,v} - o_{u,t,v'})}} + \lambda \mathbf{q}_{v'},$$

$$\frac{\partial J}{\partial \mathbf{h}_{t,q_v}^u} = - \sum \frac{\mathbf{q}_v e^{-(o_{u,t,v} - o_{u,t,v'})}}{1 + e^{-(o_{u,t,v} - o_{u,t,v'})}},$$

$$\frac{\partial J}{\partial \mathbf{h}_{t,q_{v'}}^u} = - \sum \frac{\mathbf{q}_{v'} e^{-(o_{u,t,v} - o_{u,t,v'})}}{1 + e^{-(o_{u,t,v} - o_{u,t,v'})}}.$$

Moreover, parameters in ST-RNN can be further learned with the back propagation through time algorithm (Rumelhart, Hinton, and Williams 1988). Given the derivation $\partial J / \partial \mathbf{h}_{t,q_t^u}^u$, the corresponding gradients of all parameters in the hidden layer can be calculated as:

$$\frac{\partial J}{\partial \mathbf{h}_{t-w, v_{t-w}^u}^u} = \mathbf{C}^T \left(f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_{t,q_t^u}^u} \right),$$

$$\frac{\partial J}{\partial \mathbf{C}} = \left(f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_{t,q_t^u}^u} \right) (\mathbf{h}_{t-w, v_{t-w}^u}^u)^T,$$

$$\frac{\partial J}{\partial \mathbf{q}_{t_i}^u} = (\mathbf{S}_{q_t^u - q_{t_i}^u} \mathbf{T}_{t-t_i})^T \left(f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_{t,q_t^u}^u} \right),$$

$$\frac{\partial J}{\partial \mathbf{S}_{q_t^u - q_{t_i}^u}} = \left(f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_{t,q_t^u}^u} \right) (\mathbf{T}_{t-t_i} \mathbf{q}_{t_i}^u)^T,$$

$$\frac{\partial J}{\partial \mathbf{T}_{t-t_i}} = (\mathbf{S}_{q_t^u - q_{t_i}^u})^T \left(f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_{t,q_t^u}^u} \right) (\mathbf{q}_{t_i}^u)^T.$$

Now, we can employ stochastic gradient descent to estimate the model parameters, after all the gradients are calculated. This process can be repeated iteratively until the convergence is achieved.

Experimental Results and Analysis

In this section, we conduct empirical experiments to demonstrate the effectiveness of ST-RNN on next location prediction. We first introduce the datasets, baseline methods and evaluation metrics of our experiments. Then we compare our ST-RNN to the state-of-the-art baseline methods. The final part is the parameter selection and convergence analysis.

Table 1: Performance comparison on two datasets evaluated by recall, F1-score, MAP and AUC.

		recall@1	recall@5	recall@10	F1-score@1	F1-score@5	F1-score@10	MAP	AUC
Gowalla	TOP	0.0052	0.0292	0.0585	0.0052	0.0097	0.0106	0.0372	0.6685
	MF	0.0100	0.0538	0.1146	0.0100	0.0179	0.0208	0.0527	0.7056
	MC	0.0091	0.0543	0.1015	0.0091	0.0181	0.0184	0.0510	0.7029
	TF	0.0116	0.0588	0.1120	0.0116	0.0196	0.0204	0.0551	0.7097
	PFMC	0.0159	0.0792	0.1535	0.0159	0.0264	0.0279	0.0671	0.7363
	PFMC-LR	0.0186	0.0940	0.1823	0.0186	0.0313	0.0331	0.0763	0.7580
	PRME	0.0203	0.0990	0.1896	0.0203	0.0330	0.0344	0.0847	0.7695
	RNN	0.0257	0.1349	0.2286	0.0257	0.0450	0.0416	0.0921	0.7875
	ST-RNN	0.0304	0.1524	0.2714	0.0304	0.0508	0.0493	0.1038	0.8115
GTD	TOP	0.0290	0.2105	0.3490	0.0290	0.0702	0.0634	0.1307	0.7036
	MF	0.0784	0.2993	0.4935	0.0784	0.0998	0.0897	0.1986	0.7906
	MC	0.0733	0.2995	0.4969	0.0733	0.0998	0.0903	0.1968	0.7929
	TF	0.0861	0.3469	0.5347	0.0861	0.1156	0.0972	0.2181	0.8147
	PFMC	0.0964	0.3944	0.5741	0.0964	0.1315	0.1044	0.2385	0.8376
	PFMC-LR	0.1014	0.3988	0.5775	0.1014	0.1329	0.1050	0.2428	0.8399
	PRME	0.1147	0.4128	0.5861	0.1147	0.1359	0.1066	0.2512	0.8431
	RNN	0.1216	0.4168	0.5912	0.1216	0.1389	0.1075	0.2600	0.8470
	ST-RNN	0.1654	0.4986	0.6812	0.1654	0.1662	0.1239	0.3238	0.9042

Table 2: Performance of ST-RNN with varying window width w on two datasets evaluated by recall, MAP and AUC.

dataset	w	recall@1	recall@5	recall@10	MAP	AUC
Gowalla ($d = 13$)	6h	0.0304	0.1524	0.2714	0.1038	0.8115
	12h	0.0281	0.1447	0.2623	0.0996	0.8056
	1d	0.0271	0.1417	0.2598	0.0982	0.8042
	2d	0.0256	0.1357	0.2522	0.0953	0.7997
	3d	0.0258	0.1368	0.2543	0.0956	0.8007
GTD ($d = 7$)	15d	0.1649	0.4492	0.6232	0.3018	0.8708
	1m	0.1717	0.4798	0.6529	0.3175	0.8880
	2m	0.1698	0.4892	0.6604	0.3184	0.8919
	3m	0.1654	0.4986	0.6812	0.3238	0.9042
	4m	0.1638	0.4803	0.6606	0.3145	0.8915
	6m	0.1662	0.4898	0.6710	0.3206	0.8994

Experimental Settings

We evaluate different methods based on two real-world datasets belonging to two different scenarios:

- **Gowalla**¹ (Cho, Myers, and Leskovec 2011) is a dataset from Gowalla Website, one of the biggest location-based online social networks. It records check-in history of users, containing detailed timestamps and coordinates. We would like to predict where a user will check-in next.
- **Global Terrorism Database (GTD)**² includes more than 125,000 terrorist incidents that have occurred around the world since 1970. The time information is collected based on the day level. For social good, we would like to predict which province or state a terrorist organization will attack.

On both datasets, the first 70% elements of the behavioral history of each user are selected for training, then 20% for testing and the remaining 10% data as the validation set to

¹<https://snap.stanford.edu/data/loc-gowalla.html>

²<http://www.start.umd.edu/gtd/>

tune parameters, i.e., window width and the dimensionality of latent vectors. The regulation parameter for all the experiments is set as $\lambda = 0.01$.

Then we employ several evaluation metrics. **Recall@k** and **F1-score@k** are two popular metrics for ranking tasks. The evaluation score for our experiment is computed according to where the next selected location appears in the ranked list. We report recall@k and F1-score@k with $k = 1, 5$ and 10 in our experiments. The larger the value, the better the performance. Mean Average Precision (**MAP**) and Area under the ROC curve (**AUC**) are two commonly used global evaluations for ranking tasks. They are standard metrics for evaluating the quality of the whole ranked lists. The larger the value, the better the performance.

We compare ST-RNN with several representative methods for location prediction:

- **TOP**: The most popular locations in the training set are selected as prediction for each user.
- **MF** (Mnih and Salakhutdinov 2007): Based on user-location matrix, it is one of the state-of-the-art methods for conventional collaborative filtering.
- **MC**: The markov chain model is a classical sequential model and can be used as a sequential baseline method.
- **TF** (Bahadori, Yu, and Liu 2014): TF extends MF to three dimensions, including user, temporal information and spatial information.
- **PFMC** (Rendle, Freudenthaler, and Schmidt-Thieme 2010): It is a sequential prediction method based on markov chain.
- **PFMC-LR** (Cheng et al. 2013): It extends the PFMC with the location constraint in predicting.
- **PRME** (Feng et al. 2015): It takes distance between destination location and recent visited ones into consideration for learning embeddings.

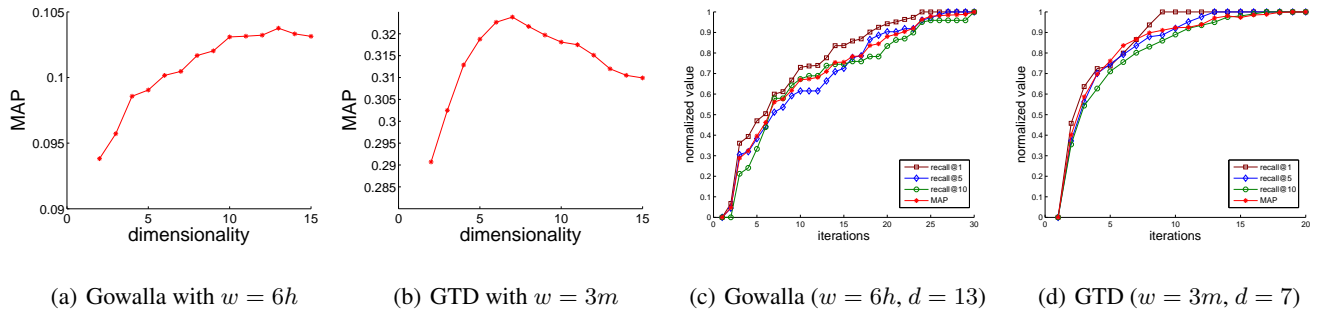


Figure 2: MAP Performance of ST-RNN on the two datasets with varying dimensionality d evaluated by MAP. Convergence curve of ST-RNN on the two datasets measured by normalized recall and MAP.

- RNN (Zhang et al. 2014): This is a state-of-the-art method for temporal prediction, which has been successfully applied in word embedding and ad click prediction.

Analysis of Experimental Results

The performance comparison on the two datasets evaluated by recall, F1-score, MAP and AUC is illustrated in Table 1. MF and MC obtain similar performance improvement over TOP, and the better one is different with different metrics. MF and MC can not model temporal information and collaborative information respectively. Jointly modeling all kinds of information, TF slightly improves the results comparing with MF and MC, but can not well predict the future. And PFMC improves the performance greatly comparing with TF. PFMC-LR and PRME achieve further improvement with via incorporating distance information. Another great improvement is brought by RNN, and it is the best method among the compared ones. Moreover, we can observe that, ST-RNN outperforms the compared methods on the Gowalla dataset and the GTD dataset measured by all the metrics. And the MAP improvements comparing with RNN are 12.71% and 24.54% respectively, while the AUC improvements are 3.04% and 6.54% respectively. These great improvements indicate that our proposed ST-RNN can well model temporal and spatial contexts. And the larger improvement on the GTD dataset shows that the impact of time interval information and geographical distance information is more significant on modeling terrorist organizations' behavior than on users' check-in behavior.

Analysis of Window Width and Dimensionality

Table 2 illustrates the performance of ST-RNN evaluated by recall, MAP and AUC with varying window widths, which can provide us a clue on the parameter selection. The dimensionality is set to be $d = 13$ and $d = 7$ respectively. On the Gowalla dataset, the best parameter is clearly to be $w = 6h$ under all the metrics. And the performances with other window width are still better than those of compared methods. On the GTD dataset, the best performance of recall@1 is obtained with $w = 1m$ while the best performance of other metrics is obtained with $w = 3m$, which indicates that the longer ranking list requires a larger window width. We select $w = 3m$ as the parameter and all the results can still defeat compared methods. These results shows that ST-RNN is not very sensitive to the window width.

To investigate the impact of dimensionality and select the best parameters for ST-RNN, we illustrate the MAP performance of ST-RNN on both datasets with varying dimensionality in Figure 2(a) and 2(b). The window width is set to be $w = 6h$ on the Gowalla dataset and $w = 3m$ for the GTD dataset. It is clear that the performance of ST-RNN stays stable in a large range on both datasets and the best parameters can be selected as $d = 13$ and $d = 7$ respectively. Moreover, even not with the best dimensionality, ST-RNN still outperforms all the compared methods according to Table 1. In a word, from these curves, we can say that ST-RNN is not very sensitive to the dimensionality and can be well applied for practical applications.

Analysis of Convergence Rates

Figure 2(c) and 2(d) illustrate the convergence curves of ST-RNN on the Gowalla and the GTD datasets evaluated by recall and MAP. To draw the curves of different metrics in one figure and compare their convergence rates, we calculate normalized values of convergence results of recall@1, recall@5, recall@10 and MAP on both datasets. The normalized values are computed according to the converge procedure of each evaluation metric which ensures the starting value is 0 and the final value is 1 for each converge curve. From these curves, we can observe that ST-RNN can converge in a satisfactory number of iterations. Moreover, on both datasets, it is obvious that the curves of recall@1 converge very soon, followed by that of recall@5, and results of recall@10 converge the slowest as well as results of MAP. From this observation, we can find that more items you would like to output in the ranked list, more iterations are needed in training.

Conclusion

In this paper, we have proposed a novel spatial temporal prediction method, i.e. ST-RNN. Instead of only one element in each layer of RNN, ST-RNN considers the elements in the local temporal contexts in each layer. In ST-RNN, to capture time interval and geographical distance information, we replace the single transition matrix in RNN with time-specific transition matrices and distance-specific transition matrices. Moreover, a linear interpolation is applied for the training of transition matrices. The experimental results on real datasets show that ST-RNN outperforms the state-of-the-art methods and can well model the spatial and temporal contexts.

Acknowledgments

This work is jointly supported by National Basic Research Program of China (2012CB316300), and National Natural Science Foundation of China (61403390, U1435221).

References

- Bahadori, M. T.; Yu, Q. R.; and Liu, Y. 2014. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *NIPS*, 3491–3499.
- Bhargava, P.; Phan, T.; Zhou, J.; and Lee, J. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *WWW*, 130–140.
- Chen, M.; Liu, Y.; and Yu, X. 2014. Nlpmm: A next location predictor with markov modeling. In *PAKDD*. 186–197.
- Chen, J.; Wang, C.; and Wang, J. 2015. A personalized interest-forgetting markov model for recommendations. In *AAAI*, 16–22.
- Cheng, C.; Yang, H.; King, I.; and Lyu, M. R. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 17–23.
- Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2605–2611.
- Cho, E.; Myers, S. A.; and Leskovec, J. 2011. Friendship and mobility: User movement in location-based social networks. In *SIGKDD*, 1082–1090.
- Ding, Y., and Li, X. 2005. Time weight collaborative filtering. In *CIKM*, 485–492.
- Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q. 2015. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, 2069–2075.
- Kapoor, K.; Subbian, K.; Srivastava, J.; and Schrater, P. 2015. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *WSDM*, 233–242.
- Koenigstein, N.; Dror, G.; and Koren, Y. 2011. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *RecSys*, 165–172.
- Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42(8):30–37.
- Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 426–434.
- Koren, Y. 2010. Collaborative filtering with temporal dynamics. *Communications of the ACM* 53(4):89–97.
- Lathia, N.; Hailes, S.; and Capra, L. 2009. Temporal collaborative filtering with adaptive neighbourhoods. In *SIGIR*, 796–797.
- Li, X.; Cong, G.; Li, X.-L.; Pham, T.-A. N.; and Krishnaswamy, S. 2015. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, 433–442.
- Liu, N. N.; Zhao, M.; Xiang, E.; and Yang, Q. 2010. Online evolutionary collaborative filtering. In *RecSys*, 95–102.
- Liu, Y.; Wei, W.; Sun, A.; and Miao, C. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *CIKM*, 739–748.
- Mikolov, T.; Karafiát, M.; Burget, L.; Cernocký, J.; and Khudanpur, S. 2010. Recurrent neural network based language model. In *INTERSPEECH*, 1045–1048.
- Mikolov, T.; Kombrink, S.; Burget, L.; Cernocký, J. H.; and Khudanpur, S. 2011a. Extensions of recurrent neural network language model. In *ICASSP*, 5528–5531.
- Mikolov, T.; Kombrink, S.; Deoras, A.; Burget, L.; and Cernocký, J. 2011b. Rnnlm-recurrent neural network language modeling toolkit. In *ASRU Workshop*, 196–201.
- Mnih, A., and Salakhutdinov, R. 2007. Probabilistic matrix factorization. In *NIPS*, 1257–1264.
- Nasraoui, O.; Cerwinski, J.; Rojas, C.; and González, F. A. 2007. Performance of recommendation systems in dynamic streaming environments. In *SDM*, 569–574.
- Natarajan, N.; Shin, D.; and Dhillon, I. S. 2013. Which app will you use next? collaborative filtering with interactional context. In *RecSys*, 201–208.
- Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 452–461.
- Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 811–820.
- Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5:3.
- Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; and Cheng, X. 2015. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, 403–412.
- Xiong, L.; Chen, X.; Huang, T.-K.; Schneider, J. G.; and Carbonell, J. G. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 211–222.
- Ye, M.; Yin, P.; Lee, W.-C.; and Lee, D.-L. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, 325–334.
- Zhang, Y.; Dai, H.; Xu, C.; Feng, J.; Wang, T.; Bian, J.; Wang, B.; and Liu, T.-Y. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI*, 1369–1376.
- Zheng, V. W.; Cao, B.; Zheng, Y.; Xie, X.; and Yang, Q. 2010a. Collaborative filtering meets mobile recommendation: A user-centered approach. In *AAAI*, 236–242.
- Zheng, V. W.; Zheng, Y.; Xie, X.; and Yang, Q. 2010b. Collaborative location and activity recommendations with gps history data. In *WWW*, 1029–1038.
- Zhong, Y.; Yuan, N. J.; Zhong, W.; Zhang, F.; and Xie, X. 2015. You are where you go: Inferring demographic attributes from location check-ins. In *WSDM*, 295–304.