HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction

Dejiang Kong and **Fei Wu**

College of Computer Science and Technology, Zhejiang University

kdjysss@gmail.com, wufei@cs.zju.edu.cn

Abstract

The widely use of positioning technology has made mining the movements of people feasible and plenty of trajectory data have been accumulated. How to efficiently leverage these data for location prediction has become an increasingly popular research topic as it is fundamental to location-based services (LBS). The existing methods often focus either on long time (days or months) visit prediction (i.e., the recommendation of point of interest) or on real time location prediction (i.e., trajectory prediction). In this paper, we are interested in the location prediction problem in a weak real time condition and aim to predict users' movement in next minutes or hours. We propose a Spatial-Temporal Long-Short Term Memory (ST-LSTM) model which naturally combines spatial-temporal influence into LSTM to mitigate the problem of data sparsity. Further, we employ a hierarchical extension of the proposed ST-LSTM (HST-LSTM) in an encoder-decoder manner which models the contextual historic visit information in order to boost the prediction performance. The proposed HST-LSTM is evaluated on a real world trajectory data set and the experimental results demonstrate the effectiveness of the proposed model.

1 Introduction

With the widespread use of positioning technology, it is increasingly possible to track and mine the movements of people. When people turn to digital map applications (Baidu Map, Google map, etc) for positioning or navigation services, their travel information is recorded and massive trajectories of them are accumulated. These trajectory data open great opportunities for location prediction research and from which location-based services (LBS) such as navigational services, traffic management and location-based recommendation and advertising are beneficial.

Location prediction has long been studied since the positioning technology is frequently used. Some researches [Lin and Hsu, 2014; Chen *et al.*, 2015; Trasarti *et al.*, 2015] fo-

cus on real time location prediction problem using continuous intensive positioning data which are mostly generated by vehicles, these researches help promote the performance of tasks like congestion prediction and route planning. Other works [Lian *et al.*, 2014; Li *et al.*, 2015; Cheng *et al.*, 2013; Liu *et al.*, 2016] pay attention to long time prediction like Point of Interest (POI) recommendation, which aims to predict potential visit pattern with a gentle time constraint.

With the fast development of urbanization and modern civilization, it is becoming more and more important to mine people's movements and predict their next destinations in a short time period in order to achieve better urban management and services. Researches [Yuan *et al.*, 2015; Vaca *et al.*, 2015] have found that the modern cities consist of functional zones such as residential areas, business districts, and educational areas, etc. People's movements among these functional zones reveal different socioeconomic activities like working, shopping, entertaining, etc. In this paper, we address the problem of discovering movement patterns among these functional zones in order to predict the next move people will go ahead in minutes or hours. Figure 1(a) illustrates an example of several trajectories different users made over different functional zones. People move from one zone to another one and may stay in one zone for a period of time. For simplicity, we assume that consecutive visits in the same zone represent the same socioeconomic activity and the point trajectories then can be compressed to zone visiting sequences as shown in Figure 1(b).

Temporal and spatial contexts are key factors in location prediction problem. Figure 2 gives an example of next locations' visiting frequencies after users' visiting location 22 in consecutive time slots. It is obviously seen that the top locations visited after the location 22 differ in each time slot and some locations like 10, 957 just appear in the first several slots while other locations such as 678, 1028 appear in the last several slots. These observations reveal the importance of spatial-temporal contexts in location prediction problem.

A lot of works [Li *et al.*, 2015; Liu *et al.*, 2016] leverage temporal and spatial features to boost prediction performance. Some of them [Lian *et al.*, 2014; Li *et al.*, 2015] leverage these features as constraints to learn a better prediction model. Others [Liu *et al.*, 2016] utilize embedding meth-
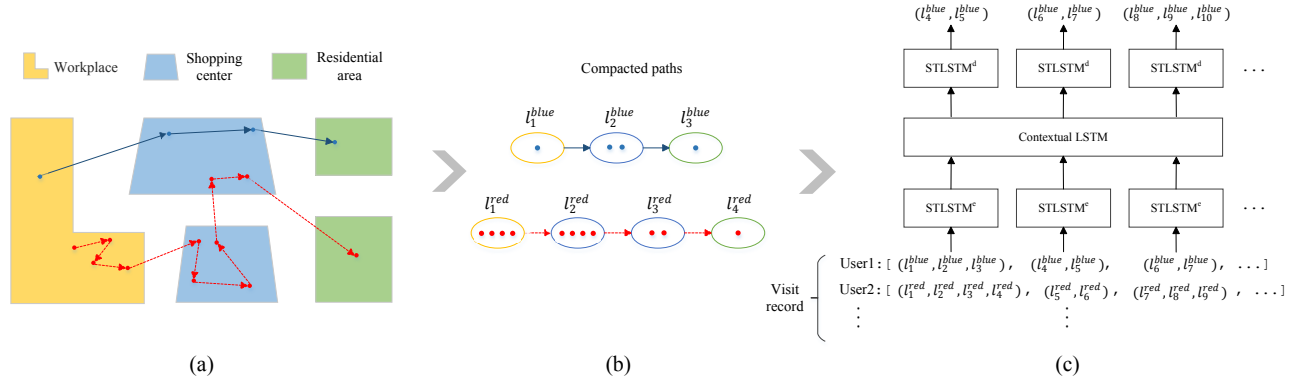
Figure 1: An overview of HST-LSTM for location prediction. Figure (a) is a simple example of users' trajectories. Blocks with different colors and shapes represent different functional zones. Users' trajectories are described in dotted lines. Figure (b) shows the AOI paths which are generated from two trajectories in Figure (a) by grouping consecutive positioning points in a same functional zone into one. Each ellipse represents an AOI and each AOI contains one or more positioning points. Figure (c) is the proposed HST-LSTM model, the detail algorithm is illustrated in section 3.

ods which embed time interval and geographical distance using vectors or tensors to encode spatial-temporal effects and achieve good prediction performance. Recently, deep neural networks especially Recurrent Neural Network (RNN) based models are more and more fancy to deal with sequential data. The introducing of the gate mechanism solves vanishing gradient problem and many tasks achieve state-of-the-art performance by using LSTM based models. However, most sequential models just emphasize the dependencies among sequence nodes, ignoring other corresponding correlations like spatial and temporal relations among them. We argue that these spatial-temporal relations are essential for mitigating the data sparsity which exists in most deep models. In this paper, we propose the idea of introducing these spatial-temporal relations into gates to guide the learning of deep models .

Recently, many prediction tasks benefit from contextual information. [Sordoni *et al.*, 2015; Song *et al.*, 2017] find that people's historical behaviors effect their choices in the future, they both take advantages of historical information as contexts and model them via hierarchical architectures and show the effectiveness. Considering the periodicity of people's movements that people tend to visit the same or similar places, we also propose a hierarchical model to encode people's visit historical information as contexts to boost prediction accuracy.

In this paper, we address the location prediction problem in a weak real time condition and the contributions can be summarized as follows:

- We propose a novel Spatial-Temporal Long-Short Term Memory (ST-LSTM) model which combines spatial-temporal influences into LSTM model naturally to mitigate the data sparsity in location prediction problem.

- To model the visit context, we propose HST-LSTM, a hierarchical extension of ST-LSTM. HST-LSTM models users' historical visiting sequences in an encoder and decoder manner to boost the prediction performance.

- Finally, we show empirically that the proposed model outperforms several location prediction methods on a re-

al world trajectory data set.

2 Related Work

POI recommendation is the task most related to our work and it focuss on recommending POIs for users with a gentle time constraint while our location prediction problem is under a weak real time condition. Most of conventional methods are based on Collaborative Filtering (CF) technique and the most popular one is user-based CF [Yuan *et al.*, 2013], which takes advantage of check-ins of similar users for recommendation. Others are the Matrix Factorization (MF) based methods [Lian *et al.*, 2014; Li *et al.*, 2015], which factorize the user-item preference matrix to learn the general taste of a user.

Recently, sequential influence has been considered for location prediction. Markov chain property is the favorite one for traditional approaches. FPMC [Rendle *et al.*, 2010] is the state-of-the-art personalized Markov chain algorithm. [Cheng *et al.*, 2013] employs FPMC to model the personalized POI transition and aims to recommend POIs for the next hours by merging consecutive check-ins in the previous hours. Based on the current POI, their method only considers the POIs in the defined region as candidates, which might be a little thoughtless. [Lian *et al.*, 2013] breaks the region restriction and also adopts FPMC to represent the short-term and long-term preference to predict the next check-in.

Embedding technique which embeds words into low dimension latent space to handle semantic gap has become popular and transfers to many studying areas. PRME [Feng *et al.*, 2015] is the state-of-the-art embedding method for POI Recommendation. It learns both users' and locations' latent vector representations at the same time by minimize the distances of successive POI check-ins and the similarities of users' representations and their own visited POIs. However, PRME merely models short-term transition patterns and ignores the long-term dependency within users' movements.

Recurrent Neural Networks (RNN) [Williams and Zipser, 1989] achieved great advance to deal with sequential data.
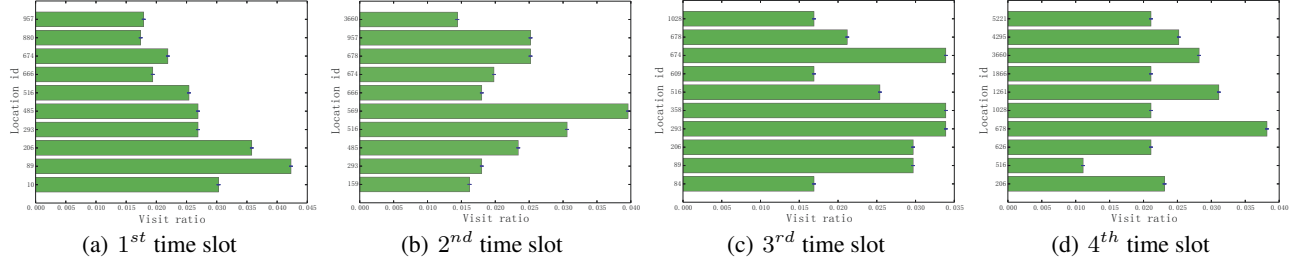
| (a) 1st time slot | (b) 2nd time slot | (c) 3rd time slot | (d) 4th time slot |

Figure 2: An example of next locations' visiting frequencies after visiting location 22 in consecutive time slots. Each time slot width is set to one hour and figures above show the next top 10 locations visited by all people in each time slot after visiting location 22. It is obvious to see that different time slots contain different location distributions, which indicates that temporal contextual influences people's visit destinations.

Many study fields, such as Language Modeling [Mikolov *et al.*, 2010], Speech Recognition [Graves *et al.*, 2013], etc. achieved state-of-the-art performance with the help of RNNs. Recently, recurrent model is applied to POI recommendation tasks. [Liu *et al.*, 2016] is the fisrt one to introduce deep neural network to location prediction problem, they extend recurrent networks (RNN) with spatial and temporal information and propose the ST-RNN to recommend POIs and achieve good results. The hierarchical architecture is favorite for RNN models recently and it produces a efficient way to memorize and utilize historical information and boosts performance of tasks such as query suggestion [Sordoni *et al.*, 2015].

3 The Proposed Model

To better convey the proposed HST-LSTM model, we first introduce some relevant definitions will be used:

- **Area of Interest (AOI)**. An area of interest is a functional zone that offers same geographical function, such as a shopping center or a workspace, which covers certain area on digtial maps and contains various individual POIs.

- **Visit Record**. The visit record consists of all the AOIs that one user visited in a certain time (several weeks or months).

- **Visit Session**. An AOI sequence which visited by a user in a certain time interval (one day in this paper as it is a natural segmentation slot of visit record) is defined as a visit session. The AOIs in one visit session have strong correlations with each other since they are all visited by a same user and they reveal the user's movement patterns.

- **Visit Session Sequence**. A visit session sequence consists of a user's consecutive visit sessions. A user's historical visit sessions of a user could act as contextual information to help predict next AOI the user is about to visit in current and future visit sessions.

Assume that a user u has n visit sessions and one visit session S_i ($i \in [1, n]$) consists of an AOI sequence $\{l_1^i, l_2^i, ..., l_{m_i}^i\}$, where m_i is the length of ith visit session. Then the visit record V_u of user u can be represented as $\{..., l_1^i, l_2^i..., l_{m_i}^i, ..., l_1^n, l_2^n..., l_{m_n}^n\}$. For simplify, we denote V_u as $\{l_1^u, l_2^u, ..., l_N^u\}$, where N is the total number of locations that user u has visited. Our problem can be formulated

as follows: *Given a user u and his/her first j ($j \in [1, N]$) visiting locations $\{l_1^u, ..., l_j^u\}$, our task is to predict next $N - j$ locations $\{l_{j+1}^u, ..., l_N^u\}$ that the user most likely to visit next.*

3.1 Spatial-Temporal LSTM

Standard LSTM [Hochreiter and Schmidhuber, 1997] is an optimized RNN model that can avoid the vanishing gradient problem by introducing the gate mechanism. Given the input l_t at time t, the output of LSTM hidden layer h_t is computed as following composite function:

$$g_t = \phi(\boldsymbol{W}_{lg}l_t + \boldsymbol{W}_{hg}h_{t-1} + b_g) \tag{1}$$
$$i_t = \sigma(\boldsymbol{W}_{li}l_t + \boldsymbol{W}_{hi}h_{t-1} + b_i) \tag{2}$$
$$f_t = \sigma(\boldsymbol{W}_{lf}l_t + \boldsymbol{W}_{hf}h_{t-1} + b_f) \tag{3}$$
$$o_t = \sigma(\boldsymbol{W}_{lo}l_t + \boldsymbol{W}_{ho}h_{t-1} + b_o) \tag{4}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{5}$$
$$h_t = o_t \cdot \phi(c_t) \tag{6}$$

where $h_0 = \boldsymbol{0}$, $\sigma(\cdot)$ is the logistic sigmoid function and $\phi(\cdot)$ is the hyperbolic tangent function, i, f, o and c are *input gate*, *forget gate*, *output gate* and *cell state* respectively. These multiple gates allow the cell in LSTM to keep, update or forget information over time. The three gates control the information flow run through the cell. In detail, the input gate i decides whether to feed the input data, the forget gate f decides whether to forget history and the output gate o decides whether to produce current state. These gates are trained from massive data to learn how to function normally. However, it is always hard to learn well functioned gates due to the data sparsity and our visit AOI sequences are such sparse data. As mentioned in former sections, different time intervals have different impacts on users' move intentions and the length of time interval is essential for predicting future movements. This is the same when it comes to spatial interval. We argue that temporal and spatial influences can work as implicit information to guide the learning of gate mechanism and we propose to add spatial-temporal factors into three gates and the equations are as follows:

$$i_t = \sigma(\boldsymbol{W}_{li}l_t + \boldsymbol{W}_{hi}h_{t-1} + F_i(s_{t-1}, q_{t-1}) + b_i) \tag{7}$$
$$f_t = \sigma(\boldsymbol{W}_{lf}l_t + \boldsymbol{W}_{hf}h_{t-1} + F_f(s_{t-1}, q_{t-1}) + b_f) \tag{8}$$
$$o_t = \sigma(\boldsymbol{W}_{lo}l_t + \boldsymbol{W}_{ho}h_{t-1} + F_o(s_{t-1}, q_{t-1}) + b_o) \tag{9}$$

where $s, q \in \mathbb{R}^d$ are vectors with dimension d, $s_0 = \mathbf{0}$, $q_0 = \mathbf{0}$ and they represent the spatial impact factor and the temporal impact factor respectively. s_{t-1}, q_{t-1} denote the vector representations of intervals between l_{t-1} and l_t. $F(\cdot)$ is a function of these two factors and we define it as an add operation:

$$F_k(s_{t-1}, q_{t-1}) = \mathbf{W}_{sk}s_{t-1} + \mathbf{W}_{qk}q_{t-1}, k = i, f, o \quad (10)$$

where $\mathbf{W}_{sk}, \mathbf{W}_{qk} \in \mathbb{R}^{|c|*d}$ are linear transition matrix with respect to spatial and temporal factors, $|c|$ is cell state size. We have tried tons of $F(\cdot)$ operations such as concatenation, element-wise addition or multiplication, etc., and find that the add operation defined above is the most concise and efficient way. If we learn a distinct vector for each possible continuous time interval and geographical distance, the parameters are uncountable and the proposed model will face more severe data sparsity problem. We solve this problem by partitioning time interval and geographical distance into discrete slots and only encoding the upper and lower bound of these slots. For a certain time interval v_q or geographical distance v_s in a slot, we calculate the corresponding vector q and s with a linear interpolation using the following equations:

$$q = \frac{\mathbf{Q}_{u(v_q)}[u(v_q) - v_q] + \mathbf{Q}_{l(v_q)}[v_q - l(v_q)]}{u(v_q) - l(v_q)} \quad (11)$$

$$s = \frac{\mathbf{S}_{u(v_s)}[u(v_s) - v_s] + \mathbf{S}_{l(v_s)}[v_s - l(v_s)]}{u(v_s) - l(v_s)} \quad (12)$$

where $u(v_q)$ and $l(v_q)$ are the upper bound and lower bound values of the time slot with respect to q, and $u(v_s)$ and $l(v_s)$ are the upper bound and lower bound values of the distance slot with respect to s. $Q \in \mathbb{R}^{N_q*d}$ and $S \in \mathbb{R}^{N_s*d}$ are the temporal factor matrix and spatial factor matrix and $N_q, N_s \in R$ are the total number of time and distance slots respectively.

3.2 Location Prediction with HST-LSTM

Based on the proposed ST-LSTM, we employ a hierarchical extension of the proposed ST-LSTM to model the periodicity of visit sequence in an encoder and decoder manner.

The Encoding of Visit Session

Given a user u and his visit record V_u, and V_u consist of visit sessions $\{S_1, S_2, ...S_n\}$, for $i \in [1, n-1]$, we first encode visit session $S_i = \{l_1^i, l_2^i, ..., l_{m_i}^i\}$ as follows:

$$l_j^i = \mathbf{W}_l \cdot l_j^i \quad (13)$$

$$h_j^{e,i} = STLSTM^e(h_{j-1}^{e,i}, l_j^i) \quad (14)$$

$$h_0^{e,i} = \mathbf{0}; j = 1, ..., m_i;$$

$$h_e^i = h_{m_i}^{e,i}; h_e^0 = \mathbf{0} \quad (15)$$

where $STLSTM^e(\cdot)$ denotes one step forward pass of encoding and l_j^i is the one-hot representation of the jth visit AOI in visit session S_i, $\mathbf{W}_l$ is the AOI embedding matrix and l_j^i is the AOI embedding representation w.r.t. l_j^i. h_j^i is the output at time step j and we take the final output $h_{m_i}^i$ as the representation of the whole visit session S_i which is denoted as h_e^i.

Global Context Encoding

After encoding the visit sessions, we get their vector representations, which are $\{h_e^0, h_e^1, ...h_e^{n-1}\}$. Then we feed them into a global contextual LSTM to model the long-term evolution of visit sequence. The equations are as follows:

$$h_c^i = LSTM^c(h_c^{i-1}, h_e^{i-1})$$

$$h_c^0 = \mathbf{0}; i = 1, ..., n; \quad (16)$$

where $LSTM^c(\cdot)$ is the global contextual LSTM and denotes one step forward pass of encoding. h_c^i represents the output of the global contextual LSTM at time step i.

Decoding and Prediction

We suppose that h_c^i acts as contextual visiting information and could be decoded to help predict potential visit patterns of visit session i. For a user u and his/her visit sessions $\{S_1, S_2, ...S_n\}$ and contextual vectors $\{h_c^1, h_c^2, ..., h_c^n\}$, we decode each visit session S_i using following equations:

$$l_j^i = \mathbf{W}_l \cdot l_j^i \quad (17)$$

$$h_j^{d,i} = STLSTM^d(h_{j-1}^{d,i}, l_j^i)$$

$$h_0^{d,i} = h_c^i; j = 1, ..., m_i; \quad (18)$$

where $STLSTM^d(\cdot)$ denotes one step forward pass of decoding. l_j^i is the one-hot representation of the jth visit AOI in visit session S_i. We assume that the output $h_j^{d,i}$ is a summary of past j travelling areas in visit session S_i and represents the travel intention at time j and it can be decoded as a distribution over AOIs which are likely to go next:

$$p_j^i = softmax(\mathbf{W}_p \cdot h_j^{d,i} + b_p) \quad (19)$$

where p_j^i is the probability distribution of AOIs. $\mathbf{W}_p$ and b_p are the transformation matrix and bias of the prediction layer and they are shared across all time steps. The probability of the most likely visit AOI after visiting l_j^i is calculated as follows:

$$p_j^i = max(p_j^i) \quad (20)$$

3.3 Learning

Given all of training visit records $T = V^1, V^2, ..., V^{\mathcal{N}}$, the objective function of HST-LSTM can be defined as follows:

$$J(\theta) = \sum_{k=1}^{\mathcal{N}} \sum_{S_i \in V^k} \sum_{j=2}^{m_i} \log p(l_j^i | l_1^i, l_2^i, ..., l_{j-1}^i, \boldsymbol{\theta}) \quad (21)$$

where $\mathcal{N}$ is the number of visiting records, l_j^i represent j^{th} AOI of visit session S_i and m_i is the length of S_i. $\boldsymbol{\theta}$ represents all of the model parameters which are learned by maximizing $J(\boldsymbol{\theta})$.

4 Experiments

4.1 Data set and settings

Our experimental data set are generated from Baidu map [1] log data. We gathered all users' traveling records in city Peking

---

[1] http://map.baidu.com

| Method	Acc@1				Acc@5				Acc@10				Acc@20			
	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%	20%	40%	60%	80%
Top	0.0106	0.0439	0.0473	0.0506	0.0544	0.0937	0.1130	0.1332	0.0627	0.1118	0.1345	0.1695	0.0711	0.1324	0.1627	0.2064
MF	0.0305	0.0487	0.0588	0.0810	0.0629	0.0982	0.1259	0.1653	0.1211	0.1526	0.1615	0.1968	0.1795	0.1850	0.1924	0.2275
VOM	0.0299	0.0436	0.0612	0.0833	0.0610	0.0923	0.1387	0.1712	0.1193	0.1496	0.1703	0.2049	0.1773	0.1812	0.2038	0.2394
FPMC	0.0473	0.0676	0.0883	0.1077	0.0617	0.0938	0.1407	0.1699	0.0786	0.1377	0.1834	0.2199	0.0829	0.1479	0.2119	0.2853
PRME	0.0425	0.0684	0.0950	0.1164	0.0666	0.1042	0.1437	0.1754	0.0889	0.1523	0.1901	0.2247	0.1199	0.1672	0.2323	0.2876
ST-RNN	0.0796	0.1072	0.1259	0.1437	0.1869	0.2715	0.3276	0.3639	0.2448	0.3604	0.4300	0.4657	0.3027	0.4421	0.5194	0.5556
LSTM	0.0728	0.0964	0.1103	0.1242	0.1875	0.2498	0.2796	0.3039	0.2554	0.3384	0.3734	0.4001	0.3298	0.4296	0.4681	0.4929
ST-LSTM	0.0875	0.1174	0.1356	0.1574	0.2312	0.3019	0.3410	0.3711	0.3132	0.3986	0.4406	0.4677	0.3988	0.4917	0.5325	0.5537
HST-LSTM	**0.0993**	**0.1331**	**0.1546**	**0.1703**	**0.2475**	**0.3250**	**0.3655**	**0.3916**	**0.3231**	**0.4138**	**0.4593**	**0.4847**	**0.3963**	**0.4959**	**0.5434**	**0.5657**

Table 1: Performance in terms of $Acc@k$ for all methods with training set size of 20%, 40%, 60%, 80%.
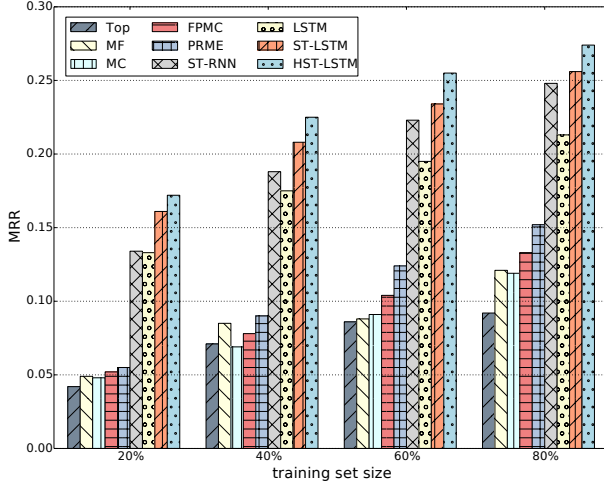
Figure 3: The mean reciprocal rank (MRR) for all methods with training set size of 20%, 40%, 60%, 80%.

during one week(from Dec. 2, 2015 to Dec. 8, 2015), which contains 311,310 usable point paths and 16,775,817 positioning points. We transform the point paths into AOI paths by grouping positioning points within the same AOI and get our final experiment data set consisting of 144,320 unique AOI paths which share 9000 semantic AOIs. To evaluate the robustness of the proposed model, we choose 20%,40%,60% and 80% of data as training set respectively, and the remaining data are as test data to evaluate the model. The parameters to learn are initialized with normal distribution. Hyper parameters such as embedding dimension $|l|$, cell state size $|c|$, spatial and temporal embedding size d are tuned using grid search. In detail, $|l|, |c| \in \{8, 16, 32, 64, 128\}$, $d \in \{8, 16, 32\}$. Parameter optimization is done using mini-batch Adam and the early stop strategy is employed, the batch size is 100 and the initial learning rate is set to 0.01 for deep models.

4.2 Comparison Methods

To prove the effectiveness of the proposed HST-LSTM, we compare our model with following prediction methods:

- **Top.** We simply choose the most popular location for prediction.
- **Matrix Factorization (MF)**[Mnih and Salakhutdinov, 2008]. Matrix Factorization based methods have been

widely used in recommendation and prediction tasks. In our location prediction problem, we use MF to mine the relationship between each two consecutive visited AOIs, and then use it for prediction.

- **Variable-Order Markov (VOM) model**[Begleiter *et al.*, 2004]. Variable-order Markov based models are the state-of-art methods for location prediction problems. We learn conditional distributions with a varying order $k - 1$ (k is the max length of an AOI path) and provides means of capturing different orders of dependencies based on training data and we adopt the principle of longest match for prediction.

- **FPMC**[Rendle *et al.*, 2010]. FPMC is the state-of-the-art personalized Markov Chain algorithm for recommendation or prediction tasks.

- **PRME**[Feng *et al.*, 2015]. PRME is the state-of-the-art embedding method for POI recommendation. It embeds users and POIs into latent spaces to calculate similarities between users and POIs and similarities among POIs for better recommendation performance.

- **ST-RNN**[Liu *et al.*, 2016]. ST-RNN is the state-of-the-art deep model for POI Recommendation, it fuses geographical influence and time influence into RNN model with distance transition matrix and time transition matrix respectively for better POI Recommendation.

- **LSTM**[Hochreiter and Schmidhuber, 1997]. LSTM network has shown its ability in handling sequential data and has superiority in encoding long-term dependencies and it is natural to apply LSTM to our sequential AOI path data for location prediction.

4.3 Evaluation Criteria

We evaluate the performance of the model and the baselines in terms of two kinds of criteria. The first one is Acc@k which can be defined as: $Acc@k = \frac{\#hit@k}{\#tests}$, where $\#hit@k$ means the number of predicted locations ranking at top k and $\#test$ stands for the number of total testing samples. In this paper, we choose $k = 1, 5, 10, 20$. Another evaluation criteria is mean reciprocal rank (MRR), which is defined as: $MRR = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{1}{Rank_i}$, where N is the total number of test instances and $Rank_i$ is the predict rank of instance i.

| Acc@1 \\ s, t	$s = 100$	$s = 200$	$s = 300$	$s = 400$	$s = 500$	$s = 600$	$s = 700$	$s = 800$	$s = 900$	$s = 1000$
$t = 0.5$	0.1480	0.1504	0.1513	0.1528	0.1529	0.1518	0.1503	0.1477	0.1478	0.1462
$t = 1$	0.1482	0.1510	0.1532	0.1540	0.1539	0.1528	0.1515	0.1501	0.1495	0.1479
$t = 1.5$	0.1490	0.1505	0.1549	**0.1574**	0.1571	0.1534	0.1526	0.1503	0.1495	0.1480
$t = 2$	0.1481	0.1514	0.1541	0.1566	0.1550	0.1529	0.1497	0.1494	0.1489	0.1466
$t = 2.5$	0.1477	0.1499	0.1518	0.1521	0.1524	0.1507	0.1482	0.1492	0.1473	0.1454
$t = 3$	0.1478	0.1483	0.1498	0.1519	0.1505	0.1492	0.1479	0.1468	0.1464	0.1457

Table 2: Performance of ST-LSTM with varying time window t (hour) and spatial window s (meter) in terms of Acc@1.

(a) Loss curves   (b) Acc@1 curves
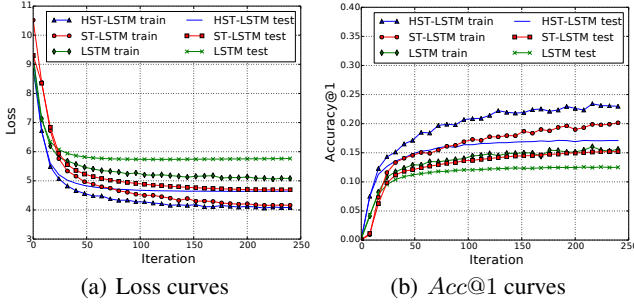
Figure 4: Figure (a) and (b) show the cross entropy loss curves and the Acc@1 curves of HST-LSTM, ST-LSTM and LSTM during first 250 learning iterations on both train and test sets. For train set, both the loss and Acc@1 values of each iteration are obtained by averaging results of all mini-batches in one iteration.

Methods	Acc@1	Acc@5	Acc@10	Acc@20
LSTM	0.1242	0.3039	0.4001	0.4929
ST-LSTM-i	0.1456	0.3572	0.4589	0.5493
ST-LSTM-f	0.1300	0.3227	0.4225	0.5176
ST-LSTM-o	0.1462	0.3562	0.4567	0.5467
ST-LSTM-if	0.1511	0.3628	0.4631	0.5519
ST-LSTM-io	0.1522	0.3665	0.4653	0.5524
ST-LSTM-fo	0.1479	0.3579	0.4581	0.5473
ST-LSTM	**0.1574**	**0.3711**	**0.4677**	**0.5537**

Table 3: Performance of ST-LSTM with different spatial-temporal factors.

4.4 Results and Analysis

Overall Performance

We report the comparison results between the proposed model HST-LSTM and other methods, the overall performances are shown in Table 1 and Figure 3. From the statistics we can conclude that:

- HST-LSTM and ST-LSTM are superior to methods with a certain large margin which demonstrates the effectiveness of the proposed spatial-temporal model.

- HST-LSTM achieves much better prediction performance than ST-LSTM in all evaluation criteria with different data settings which shows that the historical visit information are important clues for location prediction.

- Comparing to ST-RNN with a different spatial-temporal modeling mechanism, the proposed ST-LSTM model performs better and it proves the effectiveness of the idea of combining spatial-temporal factors with gate mechanism to mitigate data sparsity.

- Embedding methods are superior to non-embedding methods in most cases, this shows the advantages of embedding methods when dealing with sparse data.

- Sequential models perform quite better than non-sequential models which proves that long term dependencies are essential in sequence modeling.

Besides, we show the cross entropy loss curves and Acc@1 curves in Figure 4. It is obviously seen from Figure 4(a) that HST-LSTM and ST-LSTM model converge much better than standard LSTM model, which proves the proposed

spatial-temporal models have better abilities to fit sequential data. Figure 4(b) shows similar results with evaluation criteria Acc@1.

The Influence of Spatial-Temporal Factors on Gate Mechanism

We did experiments that merely introducing spatial-temporal factors into one or two gates, for example, ST-LSTM-i means only applying spatial-temporal factors into input gate. The results are shown in Table 3. We can see that the more gates learned by spatial-temporal factors, the better the prediction performance is. When all three gates are learned, the model performs best, which proves the fact from another perspective that three gates of a memory cell work together to process the information passing through it.

Choosing the Best Windows of Temporal and Spatial Bins

We also did batch of experiments with different temporal and spatial slot size settings and the Acc@1 results are shown in Table 2. We can observe that when time slot size $t \in [1, 2]$ and spatial slot size $s \in [300, 500]$, the best prediction performance is achieved, which is consistent with the reality that people's location transfer time unit is about one hour or two and the unit of distance interval of two AOIs is between 3 to 5 hundred meters.

5 Conclusion

This paper proposes a novel spatial-temporal LSTM model for location problem and it introduces spatial-temporal factors into gate mechanism to mitigate data sparsity problem. Further more, we leverage historical visit information and adopt a hierarchical architecture to boost the location prediction performance. The experimental results prove the effectiveness of the proposed method.

Acknowledgements

References

[Begleiter *et al.*, 2004] Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004.

[Chen *et al.*, 2015] Meng Chen, Yang Liu, and Xiaohui Yu. Predicting next locations with object clustering and trajectory clustering. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 344–356. Springer, 2015.

[Cheng *et al.*, 2013] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, pages 2605–2611, 2013.

[Feng *et al.*, 2015] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new poi recommendation. In *Proc. IJCAI*, 2015.

[Graves *et al.*, 2013] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Li *et al.*, 2015] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. Rankgeofm: a ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 433–442. ACM, 2015.

[Lian *et al.*, 2013] Defu Lian, Vincent W Zheng, and Xing Xie. Collaborative filtering meets next check-in location prediction. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 231–232. ACM, 2013.

[Lian *et al.*, 2014] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 831–840. ACM, 2014.

[Lin and Hsu, 2014] Miao Lin and Wen-Jing Hsu. Brownian bridge model for high resolution location predictions. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 210–221. Springer, 2014.

[Liu *et al.*, 2016] Qiang Liu, Shu Wu, Liang Wang, and Tie-niu Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Mikolov *et al.*, 2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820. ACM, 2010.

[Song *et al.*, 2017] Jun Song, Jun Xiao, Fei Wu, Haishan Wu, Tong Zhang, Zhongfei Zhang, and Wenwu Zhu. Hierarchical contextual attention recurrent neural network for map query suggestion. *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[Sordoni *et al.*, 2015] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562. ACM, 2015.

[Trasarti *et al.*, 2015] R Trasarti, R Guidotti, A Monreale, and F Giannotti. Myway: Location prediction via mobility profiling. *Information Systems*, 2015.

[Vaca *et al.*, 2015] Carmen Karina Vaca, Daniele Quercia, Francesco Bonchi, and Piero Fraternali. Taxonomy-based discovery and annotation of functional areas in the city. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[Williams and Zipser, 1989] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[Yuan *et al.*, 2013] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 363–372. ACM, 2013.

[Yuan *et al.*, 2015] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 2015.