An Attentive Spatio-Temporal Neural Model for Successive Point Of Interest Recommendation

Khoa D. Doan[1], Guolei Yang[2], and Chandan K. Reddy[1]

[1] Dept. of Computer Science, Virginia Tech, Arlington, VA, USA.
Email: khoadoan@vt.edu,reddy@cs.vt.edu
[2] Facebook Inc., Seattle, WA, USA. Email: glyang@fb.com

Abstract. In a successive Point of Interest (POI) recommendation problem, analyzing user behaviors and contextual check-in information in past POI visits are essential in predicting, thus recommending, where they would likely want to visit next. Although several works, especially the Matrix Factorization and/or Markov chain based methods, are proposed to solve this problem, they have strong independence and conditioning assumptions. In this paper, we propose a deep Long Short Term Memory recurrent neural network model with a memory/attention mechanism, for the successive Point-of-Interest recommendation problem, that captures both the sequential, and temporal/spatial characteristics into its learned representations. Experimental results on two popular Location-Based Social Networks illustrate significant improvements of our method over the state-of-the-art methods. Our method is also robust to overfitting compared with popular methods for the recommendation tasks.

Keywords: Deep learning · spatio-temporal data · attention mechanism · recurrent neural network · long short term memory · social networks.

1 Introduction

Location-Based Social Networks (LBSNs) produce a huge amount of data, in both veracity and volume, thus providing opportunities for building personalized Point-of-Interest (POI) recommender systems. In a typical POI recommendation task, a user makes a sequence of check-ins at various POIs that are both geotagged and time-stamped, and the task is to recommend the next POI that the user is likely interested in visiting. Here a check-in comprises of which POI is visited, and additional contextual information such as the time or geotag of the visit. Finding an efficient way to represent the POI and its contextual information is essential because this can improve the performance of the model and allow a better understanding of the seemingly complex inter-relationships of the heterogeneous properties of the POIs.

The recommendation task has been studied in numerous works [12, 11, 20, 16, 6]. One of the most widely used technique is matrix factorization (MF), or a hybrid of MF and Markov Chain (MC). These methods, albeit having impressive performance, rely on strong independent assumption among different factors. Several attempts (e.g., Neighborhood-based MF methods [16, 13, 14]) have been made to overcome these limitations, but are unable to efficiently model the sequential, periodic check-in behaviors.

It has been shown that human movements usually demonstrate strong patterns in both spatial and temporal domain [3]. To take advantage of the spatio-temporal nature of check-ins, several recommendation systems have been proposed particularly for POIs (e.g., [18]). The state-of-the-art POI recommendation systems [15, 5] use neural networks to learn the latent correlation between spatio-temporal features from historical check-ins and the next check-in location of a user. By mining spatio-temporal information from such correlations, these techniques are able to significantly outperform generic recommendation systems in the POI recommendation task.

In this paper, we will tackle this challenge and try to advance the state-of-the-art in POI recommendation systems. We propose a novel **A**ttentive **S**patio-**TE**mporal **N**eural (**ASTEN**) model that is able to recommend a POI by 1) extracting useful information from the most relevant POI visits reported by a user, and 2) minimizing the influence from non-relevant POI visits from the user. At the core of the proposed system is a Long-Short Term Memory (LSTM) Network structure, which employs the attention mechanism [2, 4] to automatically select and extract information from the most relevant check-ins on a user's trajectory and make recommendations. ASTEN's network architecture overcomes the limitations of using a single hidden vector to represent a user's dynamical check-in behavior. As a result, our system is able to exploit long user trajectories without having to deal with the excessive noise. The main contributions of our work are:

- We propose a novel ASTEN model that addresses the challenge of noise handling in user trajectory data and advances the state-of-the-art of POI recommendation systems. This is achieved by combining the LSTM Network structure with a sophisticated attention mechanism specifically designed for spatio-temporal information present in LBSN datasets. To the best of our knowledge, this approach and the model design have not been studied for POI recommendation in the literature.
- We demonstrate the effectiveness of our method using three real-world LBSN datasets. Experiments show that our model outperforms existing POI recommendation systems. Our method is not only scalable but also robust to overfitting when the complexity increases.
- From our analysis of experimental results, we derive a set of practical implications that are useful for real-world applications.

2 Related Works

We describe the prior works that capture sequential, temporal and geographical influences in the context of LBSNs.

2.1 POI Recommendation

MF-based methods are arguably one of the best user-based collaborative filtering approaches [12, 6]. Neighborhood-based MF methods attempt to incorporate temporal and spatial features. TimeSVD++ [11], for example, takes advantage of both the transition effect and the long-term transition pattern by modeling the user preference as a function of time. Similarly, recent works [16, 13] model users' interest limited to the neighborhoods of the recently visited locations. In [6], PRME learns a personalized metric embedding and models the sequential POI transition. Another popular approach for modeling sequential data is MC, which learns a transition probability matrix over sequential events. In recent works [20], instead of estimating a single matrix for all users, each user can be mapped to a personalized transition probability matrix. For example, Factorizing Personalized MC (FPMC), which has the ability to model sequential data in an MF-based approach, is the state-of-the-art method [20].

Besides the cold-start problem, the common drawbacks of the MF based approaches are their strong independent assumptions among the factorized components and that their generalization strengths depend on designing a good feature space, which might not be a realistic assumption for many real-world problems.

2.2 Neural Models and Attention Mechanism

Progress in RNNs has shown impressive results in modeling sequential data [7]. Although RNN is theoretically capable of conditioning the model on all of the previous time-steps, the number of time-steps, in practice, what such a RNN model can remember is limited because of its difficulties in training.

Because RNN assumes discrete influence of the sequential events, it does not explain well real-world situations where the transition to a POI is continuously influenced by the historical spatial and temporal context. ST-RNN [15] models the continuous local temporal and spatial contexts with time-specific and distance-specific transition matrices and achieves a significant performance improvement in the recommendation task. RMTPP [5] jointly models the prediction of the time to next events and the event themselves. ST-RNN and RMTPP, however, suffer from the bottleneck problem in RNN where the use of the single hidden vector is insufficient to capture the complex characteristics of the sequences in a problem [2]. A recent success in training RNNs is a concept of attention [2, 4]. For POI recommendation, however, it is not straightforward how the attention mechanism should be modeled.

In this paper, *we attempt to formalize the concepts of POI and check-in representations and describe how such representations can be embedded and learned within an efficient spatio-temporal attentive recurrent network structure*. The proposed model is able to **capture the sequential information, and spatio-temporal influence between check-ins in an end-to-end network that is robust to noisy check-in data**.

3 Data Description and Analysis

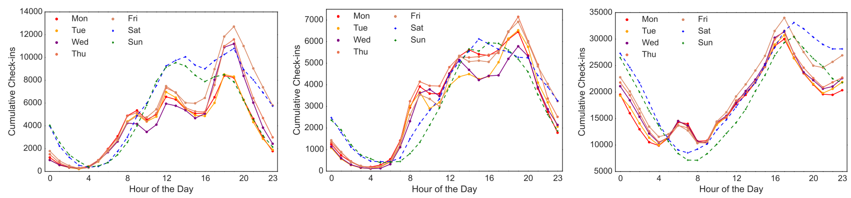
3.1 Data Description

We use three datasets collected from various activities of users on two popular LBSNs, namely, Foursquare (4SQ) and Gowalla. For 4SQ, we collect activities of users in the United States and in Europe separately and denote them as 4SQ-US and 4SQ-EU, respectively. For Gowalla, we use the dataset described in [3]. We pre-process the check-in data by filtering out POIs that were checked into by less than 10 users and users who checked-in less than 10 POIs. Table 1 summarizes the pre-processed datasets.

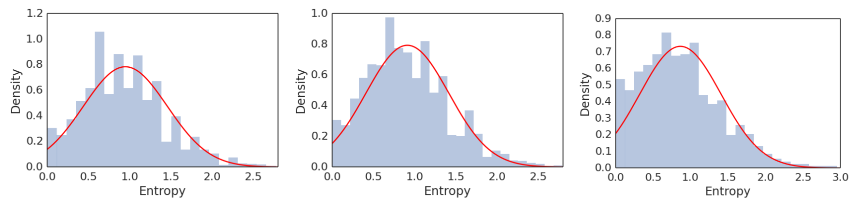
Table 1: Summary Statistics of the LBSN datasets.

	4SQ-US	**4SQ-EU**	**Gowalla**
Number of Users	21,878	15,387	52,484
Number of POIs	21,651	30,276	115,567
Number of Check-ins	569,091	56,301	3,227,845
Average Length	37	34	61

3.2 Check-in Data Exploration



(a) 4SQ-US (b) 4SQ-EU (c) Gowalla



(d) 4SQ-US (e) 4SQ-EU (f) Gowalla

Fig. 1: Statistics of check-ins in the studied datasets. Figures (a-c) show the total number of check-ins on different days of the week. Solid lines correspond to weekdays. Dashed-lines correspond to weekends. Figures (d-f) show the distribution of approximate entropies of transition distances of user check-in sequences.

Figures 1a-1c show the temporal characteristics of the check-in activities. We observe that weekdays and weekend have different patterns of cumulative check-ins, defined as the total number of observed check-ins from all users at a specific hour of the day. Moreover, check-in activities form different patterns for different hours of the day inside the weekday or weekend group. Therefore, modeling POI and check-in representations should consider the temporal periodic variances and their interaction patterns.

Next, we perform analysis on the regularity of the check-in sequences. For each user, we calculate the transition distances between the sequentially visited POIs. We employ approximate entropy [19] as a measure of the regularity and unpredictability of local fluctuations in the resulting sequences of transition distances. We set a filtering level of 1 mile. Figures 1d-1f show the histogram of approximate entropies of the sequences in the three datasets. We filter highly irregular series with infinite approximate entropies. In all datasets, the filter removes at most 25% of the users. We observe that the majority of the sequences

Table 2: Notations used in our paper.

Notation	Description
W_*	Weight matrices of the network architecture. A W_{ab} is a matrix of size $R^{m \times n}$, where m is the dimension of the input layer a and n is the dimension of the output layer b.
b_*	Bias term associated with the corresponding W_*.
l_t	One-hot encoding vector of POI at time t.
p_t	Embedding of a POI at time t.
s_{t_1,t_2}	Spatial distance between POIs at time t_1 and t_2. Note that t_1 and t_2 may not be consecutive time-steps.
t_t	Temporal periodicity vector at time t.

have low approximate entropies, which means that their transition distances exhibit not only regularity but also less fluctuation. This observation motivates us to model the distance transition behavior into a sequence's representation.

4 Proposed Methodology

4.1 Problem Definition

In this section, we formally define the successive recommendation task discussed in this paper.

Definition 1 (Check-in). *A check-in $C_u(t)$ is a tuple of $(u, l, t, s) \in U \times L \times T \times S$, where U is a set of unique users, L is a set of unique POIs, T is the continuous time domain, and S is continuous spatial domain, indexed by the latitude and longitude coordinates. $C_u(t)$ indicates that user u visited location l geo-tagged with coordinates s at time t.*

Definition 2 (User-historical check-ins). *A set of time-ordered, historical check-ins of a user u is defined as $C_u^{T_u} = \{C_u(t) : t \in [1, T_u]\}$, where T_u is the number of check-ins of user u.*

Definition 3 (Successive POI Recommendation). *Given a set of user-historical check-ins $C_u^{T_u}$, the successive point of interest recommendation task is to suggest the POI(s) that the user u will likely check-in after time T_u.*

In the following sections, we discuss our proposed method. Table 2 describes the notations used in our discussion.

4.2 POI Embedding and Check-in Representation

We propose to learn efficient representations of POIs and check-ins. Given a user u who performs a sequence of check-ins $C_u^{T_u} = (C_u(1), ..., C_u(T_u))$, where each check-in, as described, contains a POI $l^{(j)} \in L$, and the spatio-temporal information about the check-in, we learn two types of representations:

1. *POI embeddings*: we learn a function $f_{l^{(j)}} : L \mapsto R^m$ that maps every POI to a real-valued vector R^m where m is the dimension of the embedding.
2. *Check-in representations*: we learn a similar function $f_{C_u(t)} : C_u(t) \mapsto R^n$ that maps every check-in, which is a tuple of the checked POI, and its temporal information and spatial transition relationship to the previously checked POI, into a n-dimensional real-valued vector.

Given these objectives, we model a check-in x_t at time t as a function of the embedded visited-POI, its temporal context and its spatial transition distance shown in Equation 1. p_t is the one-hot encoding of the checked POI at time step t. The temporal context is a set of one-hot vectors encoding the time periodicity and denoted by $t_t = concatenate(dom_t, dow_t, hr_t)$, where dom_t is the day of the month, dow_t is the day of the week and hr_t is the hour of the day. The spatial transition context $s_{t,t-1}$ is the great-circle distance between the checked POIs at time-points t and $t-1$. The spatio-temporal model is shown in Figure 2a.

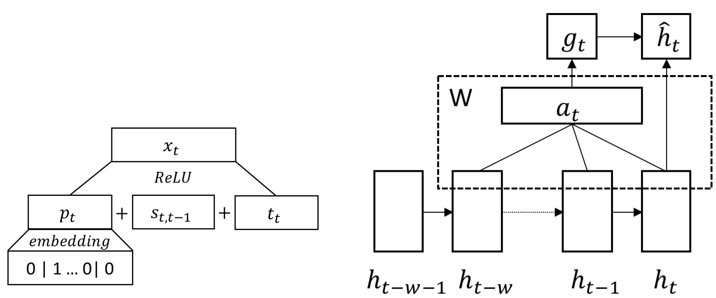
$$\begin{aligned} x_t &= ReLU(W_{vx} * c_t + b_x) \\ c_t &= concatenate(p_t, t_t, s_{t,t-1}) \end{aligned} \tag{1}$$

4.3 Recurrent Neural Networks and LSTMs

We model the hidden state as a latent representation of the past events. The predicted output, the ranked list of the recommended POIs, is a function of the hidden state:

$$\begin{aligned} h_t &= \sigma(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \\ \hat{y} &= \text{softmax}(W_{hy} h_t + b_y) \end{aligned} \tag{2}$$

where W_{hh} and W_{xh} are weight matrices of the hidden-hidden and input-hidden connections respectively, b_h is the hidden bias term, W_{hy} and b_y are the weights and bias of the hidden-output connections respectively, and σ is a Rectified Linear Unit (ReLU) [7] in our paper.



(a) Input Module (b) Attention Module

Fig. 2: Network architecture of ASTEN. (a) illustration of the recurrent input. (b) illustration of the attention module: h_t in the vanilla RNN is replaced by $\hat{h}_t$.

In our model, we use a combination of the gradient-clipping technique to overcome the gradient exploding problem [7] and the LSTM units [9] to better capture long-term dependencies.

4.4 The Proposed ASTEN Model

Most of the existing RNNs rely on the last hidden activation vector as input into a feed-forward module, such as the softmax layer used in Section 4.3. Consequently, the last hidden state becomes the primary bottleneck of the neural model as discussed in Section 2.2, which often results in non-trivial model tuning and longer training time. In our paper, we introduce an attention or memory access mechanism that allows the recommendation task to pool a fixed set of the hidden states created in the previous time-steps in order to make the recommendation.

At a time step t, we learn the unit-length alignment vector $a_t \in R^W$. W is called the window and is a hyperparameter that determines how many of the previous hidden states in the previous timesteps should play a role in constructing the pooled hidden representation. An element at position w of a_t determines the amount of information from the previous hidden state h_{t-w} the model should retain and can be calculated by:

$$a_{t,w} = \frac{\text{score}(h_t, h_{t-w})}{\sum_{k=1}^{W} \text{score}(h_t, h_{t-k})} \tag{3}$$

where there are several options for the score function. A common version of the score function can be specified as:

$$\text{score}(h_t, h_k) = h_t W_{ah} h_k \tag{4}$$

Although there are other choices of the attention score function, we have seen better performance of the proposed score function, which is similar to the findings in [17]. Since spatial check-in characteristics and temporal transition distances could influence the check-in behaviors as discussed in Section 3, we propose modeling the score as a function of the relative relationship between the spatial and temporal properties of the check-ins at time $t - w$ and t. Specifically, score can be expressed as follows:

$$\text{score}(h_t, h_k) = h_t W_{ah} h_k + h_t W_{at} t_k + h_t w_{as} s_{t,k} \tag{5}$$

where t_k is the temporal periodicity vector at time k defined in Section 4.2, while $s_{t,k}$ – similar to the definition of $s_{t,t-1}$ also in Section 4.2 – is the great-circle spatial transition distance between check-in at time k and the current check-in at time t.

Given a_t, the final attentive hidden state $\hat{h}_t$ can be calculated as follows:

$$\begin{aligned} \hat{h}_t &= W_{ch} \text{concat}(h_t, g_t) \\ g_t &= \sum_{w=1}^{W} a_{t,w} h_{t-w} \end{aligned} \tag{6}$$

Our goal, therefore, is to learn parameters of the scoring function such that the scores reflect the similarity between the past hidden states and the current hidden state t based on their temporal and spatial similarities. The architecture of our proposed ASTEN model is shown in Figure 2b.

4.5 Parameter Inference

We train an LSTM network that, given a sequence of check-ins, will recommend the next likely checked-in location. Given a sequential representation of check-ins, we minimize the cross entropy loss as follows:

$$ {}_{W_*,w_*,b_*} \frac{1}{T} \sum_{t=1}^{T} -y_{t+1} \log \hat{y}_t - (1 - y_{t+1}) \log (1 - \hat{y}_t) \tag{7} $$

where

$$ \hat{y}_t = \frac{\exp(W_{hy}\hat{h}_t + b_y)}{\sum_{j=1}^{L} \exp(W_{hy}[j,:]\hat{h}_t + b_y[j])} \tag{8} $$

and W_{hy} and b_y are the weight matrix and bias vector of the softmax classifier to predict the next checked-in POI, respectively.

To train the proposed model, we adopt the gradient based backpropagation through time training technique [8] and Adam optimizer [10]. We also employ dropout technique [7] for learning all parameters. We set the dropout value to 0.2. We train our model using an initial learning rate of 0.01 and an exponential learning rate decay of 0.96 at every 100 train steps.

5 Experimental Results

In this section, we show the performance evaluation of our proposed model through empirical experiments.

5.1 Experimental Setup

We perform our experiments on real-world LBSN datasets, namely, Foursquare (Europe and US) and Gowalla, as described in Section 3.1. We employ the 5-fold cross validation technique. The performance metrics are reported from their averages across the folds.

To evaluate the performance, we employ two popular ranking metrics, **Recall@**k and **F1-score@**k, where k is the number of recommended POIs. We also report the Area under the ROC curve (AUC) in our experiments.

5.2 Comparison Methods

We compare the effectiveness of our ASTEN model with several representative recommendation methods:

1. *Most Popular Location* (**TOP**): recommend the most popular locations.
2. *Markov Chain* (**MC**): the popular MC model for sequential data. We choose the Markov order using its generalization error on the validation set.
3. *Spatio-temporal Analysis via Low Rank Tensor Learning* (**LRTL**) [1]: an extension of Matrix Factorization into three-dimensional user, spatial and temporal information.
4. *Factorizing Personalized Markov Chains* (**FPMC**) [20]: state-of-the-art Markov chain method based on matrix factorization.
5. *Personalized Ranking Metric Embedding* (**PRME**) [6]: state-of-the-art pairwise Metric Embedding method for POI recommendation that jointly models the sequential information, user preference and geographical influence.

Table 3: Evaluation results of various methods on various LBSN datasets.

Dataset	Method	recall@1	recall@5	recall@10	F1@1	F1@5	F1@10	AUC
	TOP	0.029	0.120	0.275	0.029	0.051	0.049	0.731
	MC	0.101	0.209	0.301	0.101	0.134	0.107	0.761
	LRTL	0.125	0.237	0.307	0.125	0.135	0.128	0.787
Foursquare-US	FPMC	0.141	0.258	0.322	0.141	0.159	0.147	0.804
	PRME	0.148	0.265	0.343	0.148	0.161	0.153	0.820
	RNN	0.145	0.267	0.349	0.145	0.163	0.151	0.825
	ST-RNN	0.159	0.281	0.364	0.159	0.175	0.165	0.846
	ASTEN	**0.181**	**0.328**	**0.414**	**0.181**	**0.189**	**0.178**	**0.897**
	TOP	0.028	0.074	0.153	0.028	0.044	0.043	0.610
	MC	0.073	0.131	0.204	0.073	0.083	0.078	0.702
	LRTL	0.107	0.188	0.259	0.107	0.117	0.112	0.746
Foursquare-EU	FPMC	0.112	0.196	0.275	0.112	0.126	0.123	0.768
	PRME	0.120	0.208	0.291	0.120	0.131	0.125	0.780
	RNN	0.121	0.219	0.304	0.115	0.139	0.129	0.774
	ST-RNN	0.125	0.243	0.329	0.125	0.148	0.138	0.794
	ASTEN	**0.144**	**0.281**	**0.35**	**0.144**	**0.159**	**0.150**	**0.827**
	TOP	0.009	0.025	0.061	0.009	0.013	0.012	0.566
	MC	0.019	0.054	0.097	0.019	0.065	0.062	0.601
	LRTL	0.026	0.063	0.132	0.026	0.077	0.071	0.608
Gowalla	FPMC	0.044	0.083	0.174	0.044	0.091	0.089	0.652
	PRME	0.050	0.091	0.192	0.050	0.097	0.090	0.670
	RNN	0.048	0.098	0.189	0.048	0.121	0.095	0.673
	ST-RNN	0.061	0.120	0.223	0.061	0.138	0.120	0.695
	ASTEN	**0.081**	**0.152**	**0.266**	**0.081**	**0.165**	**0.158**	**0.735**

6. *Recurrent Neural Network* (**RNN**): RNN model for discrete temporal data.
7. *Spatial Temporal RNN* (**ST-RNN**) [15]: state-of-the-art RNN-based POI recommender system that models both local temporal and spatial transition context via time-specific and distant-specific transition matrices respectively.

For the MF models, we perform grid-search to find the best hyperparameters using a validation set, which is 20% of the training data, before evaluating their performances on a hold-out test set. For RNN and ST-RNN, we use a similar learning rate, decay schedule, and batch sizes as those of ASTEN.

5.3 POI Recommendation Performance

Table 3 shows the averaged performance results across different metrics discussed in Section 5.1. TOP has the worst performance results, as expected. MC improves over TOP since it incorporates the sequential transitions into the model. However, MC's recall, F1-scores and AUC are worse than that of the three neural models that have a better memory capacity. Since FPMC combines the successes of MF-based models and MC-based models, in our experiments, FPMC outperforms MC by at least 2% in all metrics. FPMC also outperforms LRTL. PRME improves further upon FPMC and its performance is comparable to that of RNN. However, its performance is worse than that of ST-RNN and the proposed ASTEN model in our experiments.

Among the neural models, ST-RNN expectedly outperforms RNN by 2%−5% in our results. Nevertheless, ASTEN achieves a better performance improvement compared to ST-RNN. Moreover, when K increases, ASTEN experiences the highest recall improvement compared to the other methods, suggesting that the top-ranked POIs are more relevant to the recommendation. The results are consistent across all the datasets.

5.4 ASTEN Performance Analysis

Table 4: Performance evaluation when adding spatial and temporal components.

Dataset	Method	recall@1	recall@5	recall@10
4SQ-US	LSTM	0.155	0.279	0.361
	ST-LSTM	0.161	0.303	0.378
	A-LSTM	0.159	0.309	0.371
	ASTEN	**0.181**	**0.328**	**0.414**
Gowalla	LSTM	0.049	0.105	0.192
	ST-LSTM	0.054	0.118	0.218
	A-LSTM	0.067	0.132	0.231
	ASTEN	**0.081**	**0.152**	**0.266**

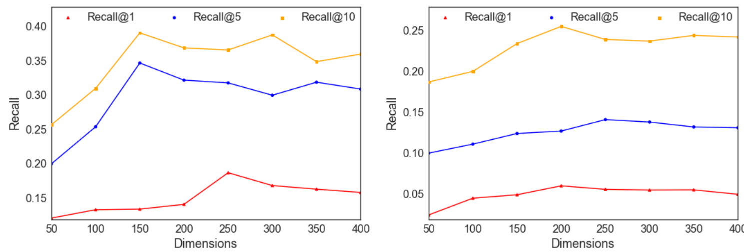
In this section, we present the performance improvements of our proposed model when various modeling components are being added in the 4SQ-US and Gowalla datasets. We look at the recall@k metric in the experiments on both 4SQ-US and Gowalla datasets in the following settings:

1. Discrete LSTM (**D-LSTM**), which is similar to the discrete (vanilla) RNN mentioned in Section 5.2 but using LSTM as hidden units.
2. Spatio-temporal LSTM (**ST-LSTM**), the ASTEN model without the spatio-temporal attention mechanism.
3. Attentive LSTM (**A-LSTM**), the ASTEN model without spatio-temporal embedding inputs as described in Section 4.2.
4. **ASTEN** model, which is our proposed model described in Section 4.4.

In Table 4, D-LSTM only slightly outperforms the previous RNN's recall discussed in Section 5.3. This result may be explained by the fact that although LSTM models are theoretically more robust to the gradient problems, in practice, this is not always the case. Both of the baseline neural models, however, have lower recall values compared to ST-LSTM and A-LSTM, both of which have comparable recall values in our experiments. We conjecture that the superior performances of these two models are due to the following reasons:

– Learning the spatio-temporal interaction of check-in sequences results in better recommendation quality.
– Our attentive mechanism captures better the check-in representation of a user, thus improving the recommendation quality.

Finally, our proposed model combines the spatio-temporal and attention mechanism and achieves the best performance in our experiments.



(a) 4SQ-US (b) Gowalla

Fig. 3: Recall@1 when varying ASTEN's dimensions.

5.5 Varying Dimensions

To determine the effect of increasing the network complexity on its performance, we vary the dimensionality of POIs and check-in representations, and the hidden layer's LSTM size from 50 to 400, and compute the network's generalization using recall@1 for each case. As the dimension increases, the network's performance increases until an optimal value is achieved, after which the recall slowly decreases, though the decrease is not very significant. We notice that the generalization recall@1's, around and after the optimal values, are still better than that of the methods compared in Section 5.3, which indicates that increasing the network's performance is not sensitive to its capacity when the dimensionality is sufficient. We conjecture that this is probably because of the dropout technique employed in our model.

6 Conclusion

We proposed a novel end-to-end learning model that takes advantage of the sequential nature and spatial/temporal contextual information of user check-ins. We also proposed a novel attention/memory access mechanism that can effectively overcome the hidden layer bottleneck of RNNs. We have shown that the proposed ASTEN model outperforms various existing methods on real-world datasets. A primary goal of our work is to find efficient representations for a learning task and our results clearly illustrate that our method could achieve this goal. Our complexity analysis shows that ASTEN outperforms state-of-the-art methods even as the number of parameters increases.

Acknowledgments

This work was supported in part by the US National Science Foundation grants IIS-1619028, IIS-1707498 and IIS-1838730.

References

1. Bahadori, M.T., Yu, Q.R., Liu, Y.: Fast multivariate spatio-temporal analysis via low rank tensor learning. In: Advances in neural information processing systems. pp. 3491–3499 (2014)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD inter-

national conference on Knowledge discovery and data mining. pp. 1082–1090. ACM (2011)
4. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Advances in Neural Information Processing Systems. pp. 577–585 (2015)
5. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1555–1564. ACM (2016)
6. Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y.M., Yuan, Q.: Personalized ranking metric embedding for next new poi recommendation. In: IJCAI. pp. 2069–2075 (2015)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning, vol. 1 (2016)
8. Guo, J.: Backpropagation through time. Unpubl. ms., Harbin Institute of Technology (2013)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Koren, Y.: Collaborative filtering with temporal dynamics. Communications of the ACM **53**(4), 89–97 (2010)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009)
13. Li, X., Cong, G., Li, X.L., Pham, T.A.N., Krishnaswamy, S.: Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp. 433–442. ACM (2015)
14. Liu, N.N., Zhao, M., Xiang, E., Yang, Q.: Online evolutionary collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems. pp. 95–102. ACM (2010)
15. Liu, Q., Wu, S., Wang, L., Tan, T.: Predicting the next location: A recurrent model with spatial and temporal contexts. In: AAAI. pp. 194–200 (2016)
16. Liu, Y., Wei, W., Sun, A., Miao, C.: Exploiting geographical neighborhood characteristics for location recommendation. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 739–748. ACM (2014)
17. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
18. Noulas, A., Scellato, S., Lathia, N., Mascolo, C.: Mining user mobility features for next place prediction in location-based services. In: Data mining, IEEE 12th international conference on. pp. 1038–1043 (2012)
19. Pincus, S.M.: Approximate entropy as a measure of system complexity. Proceedings of the National Academy of Sciences **88**(6), 2297–2301 (1991)
20. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 811–820. ACM (2010)