Distance2Pre: Personalized Spatial Preference for Next Point-of-Interest Prediction

Qiang Cui[1], Yuyuan Tang[2], Shu Wu[1(✉)], and Liang Wang[1]

[1] Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
cuiqiang2013@ia.ac.cn, {shu.wu,wangliang}@nlpr.ia.ac.cn
[2] University of Science and Technology Beijing, Beijing, China
tangyyuanr@gmail.com

Abstract. Point-of-interest (POI) prediction is a key task in location-based social networks. It captures the user preference to predict POIs. Recent studies demonstrate that spatial influence is significant for prediction. The distance can be converted to a weight reflecting the relevance of two POIs or can be utilized to find nearby locations. However, previous studies almost ignore the correlation between user and distance. When people choose the next POI, they will consider the distance at the same time. Besides, spatial influence varies greatly for different users. In this work, we propose a Distance-to-Preference (Distance2Pre) network for the next POI prediction. We first acquire the user's sequential preference by modeling check-in sequences. Then, we propose to acquire the spatial preference by modeling distances between successive POIs. This is a personalized process and can capture the relationship in user-distance interactions. Moreover, we propose two preference encoders which are a linear fusion and a non-linear fusion. Such encoders explore different ways to fuse the above two preferences. Experiments on two real-world datasets show the superiority of our proposed network.

Keywords: POI · Sequential preference · Spatial preference · Non-linear

1 Introduction

Point-of-interest (POI) prediction is one of the most important tasks in location-based social networks (LBSNs). With rich check-ins and contextual information, physical movements of users can be predicted, which is beneficial to explore POIs for users, launch advertisements, and so on. In this work, we focus on successive POI prediction by modeling check-in sequences and incorporating spatial influence in a personalized way.

Q. Cui and Y. Tang—These authors contributed equally to this paper.

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11441, pp. 289–301, 2019.
https://doi.org/10.1007/978-3-030-16142-2_23

Spatial influence has been considered in lots of works and mostly modeled by computing the distance between two POIs. The distance can be computed as a weight to reflect the relevance of two POIs [5,11]. Usually, the smaller the distance, the stronger the relevance. Besides, people can apply the distance to find nearby locations. Neighbors around a visited POI can be considered as negative samples for BPR optimization criterion [1], used to construct a hierarchical preference [20], and so on. People can also divide multiple locations close to each other into the same region [4]. Furthermore, recent works try to acquire spatial influence between POIs in other formats. Wang et al. [17] apply three factors to model spatial influence: geo-influence, geo-susceptibility, and distance. Geo-influence acquires a POI's ability to spread its spatial influence to other POIs. Geo-susceptibility captures how a POI is spatially influenced by others.

Although the aforementioned studies achieve successful results, they still have a critical limitation. These modelings of spatial influence are conducted within POIs and do not consider the relationship with users. They capture the sequential preference by modeling user-poi check-in sequences, but people have preferences for distances. For example, if a user wants some spicy food in a restaurant, how far would he want to go? It is likely that there will be several restaurants which all satisfy the user interest at different distances. Under such a situation, it is beneficial to predict user's preference for the distance that a user would take at next time. Previous works almost ignore the user's personalized choice of distance, while we propose to model the spatial preference.

In this paper, we propose a Distance-to-Preference (Distance2Pre) network to predict the next POI. First, we apply the recurrent neural network to model check-in sequences and construct the sequential preference. Then, based on distances of successive POIs, spatial preference can be computed to indicate the probabilities of different distances for the next time. This preference can explore the relationship between user and distance. Then, we devise different preference encoders, which can explore the influence of different combinations of the two preferences on the performance of POI prediction. Specifically, we propose a linear fusion and a non-linear fusion. Next, a pair-wise ranking framework is used to optimize the two preferences. The contributions are as follows:

– We first introduce and compute the personalized spatial preference, which can effectively capture the relationship between the user and spatial distance.
– We propose a linear way and a non-linear way as preference encoders to combine sequential preference with spatial preference.
– Experiments on two real-world datasets reveal that our network is effective and outperforms the state-of-the-art methods.

2 Related Work

In this section, we briefly review the POI prediction, including modeling successive POIs and incorporating spatial influence.

We can arrange a user's successive POIs into a check-in sequence and it is important to model the sequential pattern. Many studies apply the Markov chain to predict POIs. Cheng et al. [1] recommend POIs based on first-order Markov chain. Recently, the neural network is also investigated to model sequences. The work [14] applies the word2vec to model context of locations (Fig. 1).

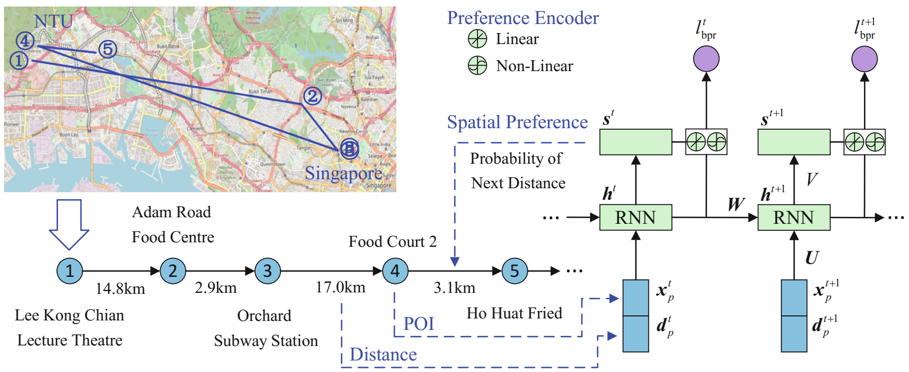


Fig. 1. A user's POIs in Singapore and the framework of our Distance2Pre network. At t-th time, the input is a POI vector $\boldsymbol{x}_p^t$ and a distance vector $\boldsymbol{d}_p^t$. The hidden state $\boldsymbol{h}^t$ is used to compute the spatial preference $\boldsymbol{s}^t$ for next time. User's sequential preference and spatial preference are fused by our preference encoders.

Liu et al. [12,13] employs recurrent neural network (RNN) to model POIs by using different contexts. RNN can model the recent check-ins. Because of the gradient vanishing and exploding problem, gated activation function like gated recurrent unit (GRU) [2] and long short-term memory (LSTM) [8] are developed to better capture the long-term dependency.

The spatial influence has been proven to be a significant factor in POI prediction. Firstly, some works convert the distance to a weight. Feng et al. [5] incorporate the spatial influence by using the weight of distance. The smaller the distance between the last POI and a POI, the more likely this POI to be recommended. Li et al. [11] build a Rank-GeoFM model to capture the user preference as well as spatial influence score, but the distance is still used as a weight between a POI and its neighbors. Secondly, people apply distance to find neighbors for a visited POI. The study [4] builds a binary tree by distances. Nearby POIs are clustered into the same region in this POI2Vec model because they are highly relevant. Zhao et al. apply the POIs that are nearby and far away to construct a hierarchical pairwise preference relation [20]. Thirdly, some studies other spatial information in addition to distance. Wang et al. [17] model a POI's ability to spread its visited users to other POIs (geo-influence) and receive users from other POIs (geo-susceptibility). However, spatial influence mostly works between POIs now and no work has studied the user's spatial preference.

3 The Distance2Pre Network

In this section, we begin with the problem formulation of the next POI prediction, then introduce the proposed Distance-to-Preference (Distance2Pre) network. In detail, we model the check-in sequence to obtain sequential preference and model the distance sequence to capture spatial preference for each user. Then, we fuse sequential and spatial preferences linearly and non-linearly.

3.1 Problem Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ be the sets of users and POIs respectively. Use $\mathcal{I}_u = (\mathcal{I}_u^1, ..., \mathcal{I}_u^{|\mathcal{I}^u|})$ to represent the check-ins of user u in the time order. Given each user's check-ins $\mathcal{I}_u$, the latitude and longitude of each POI, our goal is to generate a list of POIs for u at next time.

3.2 Sequential Preference

In this part, we model user-POI sequences and capture the sequential preference. Previous work has indicated that the sequential pattern is important for POI prediction [5].

Instead of using traditional Markov chain, we apply RNN to model each user's check-ins $\mathcal{I}_u$.

$$\boldsymbol{h}^t = f\left(\boldsymbol{U}\boldsymbol{x}_p^t, \boldsymbol{W}\boldsymbol{h}^{t-1}, \boldsymbol{b}\right), \qquad \boldsymbol{h}^t \in \mathbb{R}^d, \tag{1}$$

where $\boldsymbol{h}^t$ is the hidden state, $\boldsymbol{U}, \boldsymbol{W}$ are transition matrices and $\boldsymbol{b}$ is the bias. A vector $\boldsymbol{x}_p^t \in \mathbb{R}^d$ is used to represent the POI at the t-th time, where the subscript p indicates this POI is in $\mathcal{I}_u$. Function $f(\cdot)$ is non-linear, and we choose the gated recurrent unit (GRU) in order to better capture the long-term dependency [2].

$$\begin{aligned}
\boldsymbol{z}^t &= \sigma\left(\boldsymbol{U}_1\boldsymbol{x}_p^t + \boldsymbol{W}_1\boldsymbol{h}^{t-1} + \boldsymbol{b}_1\right) \\
\boldsymbol{r}^t &= \sigma\left(\boldsymbol{U}_2\boldsymbol{x}_p^t + \boldsymbol{W}_2\boldsymbol{h}^{t-1} + \boldsymbol{b}_2\right) \\
\tilde{\boldsymbol{h}}^t &= \tanh\left(\boldsymbol{U}_3\boldsymbol{x}_p^t + \boldsymbol{W}_3\left(\boldsymbol{r}^t \odot \boldsymbol{h}^{t-1}\right) + \boldsymbol{b}_3\right) \\
\boldsymbol{h}^t &= \left(1 - \boldsymbol{z}^t\right) \odot \boldsymbol{h}^{t-1} + \boldsymbol{z}^t \odot \tilde{\boldsymbol{h}}^t
\end{aligned} \tag{2}$$

where $\boldsymbol{U}_{1\sim3}, \boldsymbol{W}_{1\sim3} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_{1\sim3} \in \mathbb{R}^d$. GRU has an *update* gate $\boldsymbol{z}^t$ and a *reset* gate $\boldsymbol{r}^t$ to control the flow of information. $\tilde{\boldsymbol{h}}^t$ is the candidate state.

In our network, we consider $\boldsymbol{h}$ and $\boldsymbol{x}$ as latent vectors for a user and a POI respectively. Inspired by matrix factorization, a user's preference for a POI by considering sequential preference is denoted as

$$\hat{x}_{up}^t = \left(\boldsymbol{h}^t\right)^{\mathrm{T}} \boldsymbol{x}_p^{t+1} \tag{3}$$

where $\boldsymbol{h}^t$ is used as the current user latent vector when modeling $\mathcal{I}_u$. The preference $\hat{x}_{up}^t$ is an inner product between $\boldsymbol{h}^t$ and the next POI vector $\boldsymbol{x}_p^{t+1}$.

3.3 Spatial Preference

We acquire the spatial preference from user-distance sequences. Previous works show that spatial influence is helpful, but it is usually modeled only among POIs [1,4,5,11,17,20]. We go a step further and build the relationship between users and distances. In previous studies, we find that people define a return time prediction problem and propose to apply survival analysis [3,9,10]. These works model time gaps from a user's visiting sequence to predict when a user will return to the service. However, they usually predict a certain time value as a single regression task which does not help to recommend items. Inspired by these works but different from them, we model a user's spatial preference for a wide range of distances and promote the task of recommending POIs.

To model the spatial preference, we map each distance value to an interval. Firstly, all distances between two successive POIs in each $\mathcal{I}_u$ are computed. We define two values δd and M_D to represent the minimum interval and maximum interval. Then, we have a vector $[0, \delta d, 2\delta d, ..., M_D]$ to indicate all intervals. Each distance is converted to an interval. If a distance is bigger than M_D, it is also represented by M_D. Then, the modeling of distance is converted to the modeling of interval. Just like each POI has a vector $\boldsymbol{x}$, we define a latent vector $\boldsymbol{d} \in \mathbb{R}^d$ for each interval d, and this operation forms a latent matrix $\boldsymbol{D} \in \mathbb{R}^{(M_D+1)\times d}$ for all intervals $[0, \delta d, 2\delta d, ..., M_D]$. The non-bold d is a value, while the bold $\boldsymbol{d}$ is a vector. Given $\mathcal{I}_u$ for each user, we will have a sequence of intervals $\left[d_p^1, d_p^2, ...\right]$ and a sequence of vectors $\left[\boldsymbol{d}_p^1, \boldsymbol{d}_p^2, ...\right]$. Next, we update the computation of $\boldsymbol{h}^t$.

$$\boldsymbol{h}^t = f\left(\boldsymbol{U}\left[\boldsymbol{x}_p^t; \boldsymbol{d}_p^t\right], \boldsymbol{W}\boldsymbol{h}^{t-1}, \boldsymbol{b}\right), \qquad \boldsymbol{h}^t \in \mathbb{R}^d, \tag{4}$$

where $\boldsymbol{U} \in \mathbb{R}^{d\times 2d}$, $\boldsymbol{d}_p^t$ is concatenated with $\boldsymbol{x}_p^t$.

At each time, we calculate spatial preference of all intervals for next time.

$$\begin{aligned} \boldsymbol{s}^t &= \mathrm{SoftReLU}\left(\boldsymbol{V}_{\mathrm{s}}\boldsymbol{h}^t + \boldsymbol{b}_{\mathrm{s}}\right) \\ &= \left[\boldsymbol{s}^t(0), \boldsymbol{s}^t(\delta d), \boldsymbol{s}^t(2\delta d), ..., \boldsymbol{s}^t(M_D)\right] \end{aligned} \tag{5}$$

where each value in $\boldsymbol{s}^t$ is the spatial preference for a certain interval. Accordingly, the spatial preference for next ground truth interval is $\boldsymbol{s}^t\left(d_p^{t+1}\right)$, where the value d_p^{t+1} is the distance interval between $\boldsymbol{x}_p^t$ and $\boldsymbol{x}_p^{t+1}$.

3.4 Preference Encoders

As we have two preferences, we need to encode them together and we propose a linear way and a non-linear way. Our network considers not only which POIs a user would like at next time, but also how far he wants to go.

By introducing a weight w_d, the sequential preference and spatial preference can be combined together linearly.

$$\hat{x}_{up}^t = \left(\boldsymbol{h}^t\right)^{\mathrm{T}} \boldsymbol{x}_p^{t+1} + w_d \boldsymbol{s}^t\left(d_p^{t+1}\right) \tag{6}$$

However, linear fusion is natural. It is worth exploring non-linearity to investigate correlations between two preferences. Inspired from the attention mechanism, our innovative strategy is

$$\hat{x}_{up}^{t} = \boldsymbol{v}_a^{\mathrm{T}} \tanh\left(\boldsymbol{r}_a \left(\boldsymbol{h}^t\right)^{\mathrm{T}} \boldsymbol{x}_p^{t+1} + \boldsymbol{e}_a \boldsymbol{s}^t \left(d_p^{t+1}\right)\right) \tag{7}$$

where $\boldsymbol{v}_a, \boldsymbol{r}_a, \boldsymbol{e}_a \in \mathbb{R}^{d\times d}$ are weight vectors. The attention mechanism enables a model to concentrate on critical parts and it has been widely in many tasks and fields, such as image classification [15], next item recommendation [18], and so on. However, previous works using attention usually assign an appropriate weight for each factor to tell its importance. Therefore, the attention previously models one-to-many problems. In our work, we change the attention to capture one-to-one relationship and replace two commonly used weight matrices in attention with two weight vectors $\boldsymbol{r}_a, \boldsymbol{e}_a$. By innovatively using the attention, we create a non-linear combination Eq. (7) to encode two preferences.

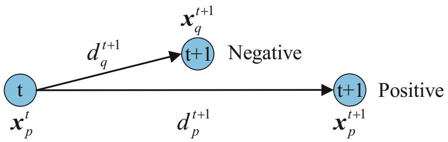


Fig. 2. Illustration of positive and negative POIs and distances from t-th time to $(t+1)$-th time. The $\boldsymbol{x}_p^{t+1} \in \mathcal{I}_u$ is positive. The negative POI $\boldsymbol{x}_p^{t+1} \notin \mathcal{I}_u$ is randomly chosen. The negative distance (interval) d_q^{t+1} is computed between $\boldsymbol{x}_p^t$ and $\boldsymbol{x}_q^{t+1}$.

3.5 Training Framework

In this subsection, we apply the widely-used pair-wise Bayesian Personalized Ranking (BPR) [1,5,16] to train the model.

$$l_{\text{bpr}}^{t} = -\ln \sigma\left(\hat{x}_{up}^{t} - \hat{x}_{uq}^{t}\right) \tag{8}$$

where $\hat{x}_{up}^{t}$ and $\hat{x}_{uq}^{t}$ are positive and negative preferences. At each time, a negative POI $\boldsymbol{x}_q \notin \mathcal{I}_u$ is randomly chosen from $\mathcal{I}$. Illustrated in Fig. 2, the negative distance d_q^{t+1} is calculated between $\boldsymbol{x}_p^t$ and $\boldsymbol{x}_q^{t+1}$. Finally, the loss function is

$$\Theta^{*} = \underset{\Theta}{\operatorname{argmin}} \sum_{u} \sum_{t=1}^{t=|\mathcal{I}_u|} l_{\text{bpr}}^{t} + \frac{\lambda_\Theta}{2}\|\Theta\|^2 \tag{9}$$

where Θ denotes a set of parameters $\Theta = \{\boldsymbol{X}, \boldsymbol{D}, \boldsymbol{U}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{V_s}, \boldsymbol{b_s}, w_d, \boldsymbol{v}_a, \boldsymbol{r}_a, \boldsymbol{e}_a\}$, where $\boldsymbol{X}, \boldsymbol{D}$ are sets of all POI vectors and all distance vectors respectively. Stochastic gradient descent (SGD) is used to learn the parameters.

4 Experiments

4.1 Experimental Settings

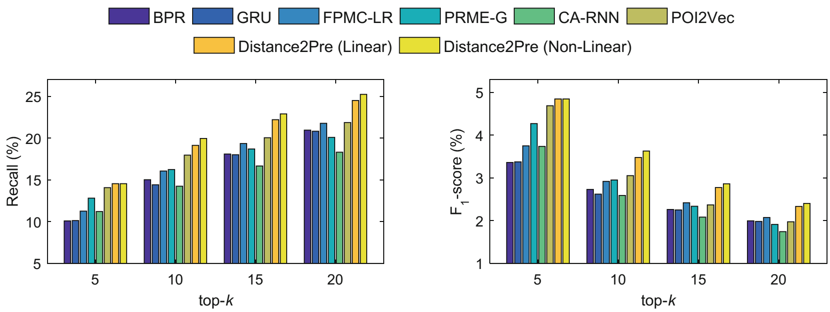
Datasets. We apply two widely-used datasets called Foursquare and Gowalla which are preprocessed in [19]. Specifically, all the information used in our work includes each user's chronological check-in sequence and the corresponding distance sequence, except for the time of check-ins. Following previous works [6,7,16], we employ the leave-one-out evaluation. For each user's check-in sequence, we treat the last POI as the test data and apply the rest POIs for training.

Comparison Methods. Our Distance2Pre network is compared with the following methods. (1) **BPR** [16]: This method refers to the BPR-MF for implicit feedback. It optimizes the difference of the user's preferences for positive and negative items. (2) **GRU** [2]: RNN is effective for successive POI prediction. We apply GRU in this work. (3) **FPMC-LR** [1]: This work is based on first-order Markov chain and uses neighbors as negative samples. (4) **PRME-G** [5]: It is a metric embedding method, and the spatial distance is considered as the weight. (5) **CA-RNN** [12]: A novel model incorporates input and transition contexts. Accordingly, we apply GRU to implement CA-RNN and compute the transition context by using distance intervals. (6) **POI2Vec** [4]: A binary tree is used to cluster the nearby POIs into the same region. Moreover, a POI is assigned to multiple regions in this model to strengthen the spatial influences of POIs. As our proposed network has a linear fusion and a non-linear fusion, it has two variants: **Distance2Pre (Linear)** and **Distance2Pre (Non-Linear)**.

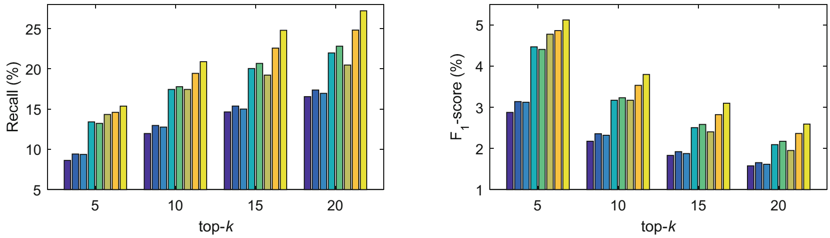
Evaluation Metrics. The top-k metrics are popular for POI prediction [1,4,5,11,20]. In this work, we apply metrics called Recall and F_1-score. Values of metrics in our work are all expressed as percentages. During the test, each user's training sequence $(\boldsymbol{x}_p^1, ..., \boldsymbol{x}_p^n)$ is recomputed by using GRU to obtain $\boldsymbol{h}^n$ and $\boldsymbol{s}^n$. Then, $\boldsymbol{h}^n$ is applied to acquire user's sequential preferences for all items $\boldsymbol{X}$. Meanwhile, all distances between $\boldsymbol{x}_i^n$ and each item in $\boldsymbol{X}$ are calculated because we do not know any information about the test set. These distances are fixed in each epoch and converted to spatial preferences for $\boldsymbol{X}$ by using $\boldsymbol{s}^n$. Then, we obtain each user's final preferences for all items and recommend *top*-k items with the highest preference.

Additionally, parameters Θ are initialized to the same range, e.g., uniform distribution $[-0.5, 0.5]$. The learning rate, regularization λ_Θ and the dimension are set as 0.01, 0.001 and 20 for all methods. Weight w_d is initialized by a positive value 1.0 and is also updated by SGD. Details of w_d are illustrated in Fig. 4. The code is written by using Theano and is available on GitHub[1].

[1] https://github.com/cuiqiang1990/Distance2Pre.



(a) Foursquare



(b) Gowalla

Fig. 3. Experimental results on two datasets.

4.2 Performance Comparison

Performances of all methods are illustrated in Fig. 3. First, we explore baselines BPR, GRU, and FPMC-LR. They are comparable but perform differently on two datasets. FPMC-LR is always better than BPR and proves the effectiveness of the spatial influence. GRU performs worst among them on Foursquare. Perhaps because there is more than one behavior at a certain time and we are unable to know the true order of these multiple check-ins. This disordered property in Foursquare hinders the sequential modeling of GRU. Fortunately, GRU is the best on Gowalla which has the right time order. This meaningful result indicates that correct sequential modeling is important for POI prediction.

In the following, we compare PRME-G, CA-RNN, POI2Vec with our Distance2Pre network. First, performances of these four methods are also adversely affected by the disordered property. Their performance on Foursquare is close to or even below the performance of BPR, GRU and FPMC-LR, especially CA-RNN. On the contrary, their performance on Gowalla is obviously better. The CA-RNN treats distance intervals as transition contexts. Accurately, CA-RNN acquires a transition matrix for every possible interval. Such a precise modeling will result in great improvement as well as great decline, which depends on whether the order is correct or not. POI2Vec has a comparable performance of *top*-5 with our Distance2Pre on both datasets, while it is obviously weaker than our network on Recall@20 and F_1-score@20. Actually, POI2Vec clusters nearby POIs into the same region, which causes a strong local correlation of POIs. Therefore, POI2vec is good at recommending a small quantity of POIs.

Overall, our Distance2Pre is optimal on two datasets. Our spatial preference is powerful for predicting next POI and robust to disordered property in Foursquare. We have a visualization of spatial preference in Sect. 4.5 and find that people have personalized moving pattern. Such a pattern is a kind of user interest which is regular and does not change dramatically. Therefore, our Distance2Pre can still get good performance on Foursquare.

4.3 Settings of Max Distance M_D and Distance Interval δd

In this part, we explore the effect of max distance $M_D(km)$ and distance interval $\delta d(km)$ on the spatial preference $\boldsymbol{s}$ in Eq. (5). The M_D and δd reflect the range and granularity of $\boldsymbol{s}$. Results are in Table 1.

The proper M_D and δd are chosen based on all the distances between successive POIs in user sequences. The distribution of distances is different on two datasets. For example, $M_D = 20$ km covers 97.6% and 79.9% distances on Foursquare and Gowalla respectively. Gowalla has a greater proportion of large distance. Finally, we set $M_D = [2.5, 5, 10], \delta d = [0.10, 0.15, 0.20]$ for Foursquare and $M_D = [10, 20, 40], \delta d = [0.10, 0.20, 0.30]$ for Gowalla. Obviously, $M_D = 5, \delta d = 0.10$ and $M_D = 5, \delta d = 0.15$ are best for our linear fusion and non-linear fusion on Foursquare. $M_D = 20, \delta d = 0.20$ and $M_D = 20, \delta d = 0.10$ are the best on Gowalla. We can see that if a dataset covers more larger distances, setting larger $M_D, \delta d$ may be more suitable. The comparison between linear fusion and non-linear fusion is discussed in the next subsection.

Table 1. Performance evaluated by Recall@10 with varying max distance M_D (km) and distance interval δd (km).

Method				Distance2Pre (linear)						Distance2Pre (non-linear)					
Evaluation				Recall@10			F_1-score@10			Recall@10			F_1-score@10		
			δd	0.10	0.15	0.20	0.10	0.15	0.20	0.10	0.15	0.20	0.10	0.15	0.20
Foursquare	M_D	2.5		18.23	18.53	18.44	3.31	3.37	3.35	19.86	19.74	19.26	3.61	3.59	3.50
		5		**19.13**	18.92	18.48	**3.48**	3.44	3.36	19.13	**19.96**	19.39	3.47	**3.63**	3.53
		10		18.70	18.57	18.48	3.40	3.38	3.36	19.57	19.65	19.70	3.56	3.57	3.58
			δd	0.10	0.20	0.30	0.10	0.20	0.30	0.10	0.20	0.30	0.10	0.20	0.30
Gowalla	M_D	10		19.23	19.23	19.01	3.50	3.50	3.46	16.79	20.71	16.06	3.05	3.77	2.92
		20		19.33	**19.44**	19.04	3.51	**3.53**	3.46	**20.89**	19.57	19.77	**3.80**	3.56	3.59
		40		19.15	19.33	19.14	3.48	3.51	3.48	19.96	20.28	20.28	3.63	3.69	3.69

Table 2. Performance of our proposed Distance2Pre network on two datasets.

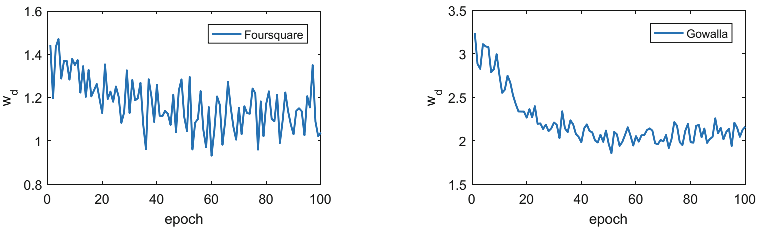
Method	Distance2Pre (linear)								Distance2Pre (non-linear)							
Evaluation	Recall@				F_1-score@				Recall@				F_1-score@			
	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
Foursquare	14.55	19.13	22.21	24.50	4.85	3.48	2.78	2.33	14.55	19.96	22.90	25.24	4.85	3.63	2.86	2.40
Gowalla	14.59	19.44	22.57	24.82	4.86	3.53	2.82	2.36	15.37	20.89	24.79	27.21	5.12	3.80	3.10	2.59

4.4 Linear Fusion Vs. Non-Linear Fusion

In this subsection, we investigate the linear fusion and non-linear fusion by analysis values in Tables 1 and 2 and Fig. 4. Assuredly, non-linear fusion is more powerful to handle two different preferences.

The performance is higher under non-linear fusion. In Table 1, there is a big difference in values between linear fusion and non-linear fusion on both datasets. Most values under non-linear fusion are obviously larger than those under linear fusion. Moreover, many values of Recall@10 are one percentage point higher than corresponding values in the left half of the table. By using best parameters, the performance of our two Distance2Pre networks is shown in Table 2. It is interesting that the difference between the two kinds of fusions on Gowalla is greater than that on Foursquare. Perhaps non-linearity is more powerful to deal with more complex situations as Gowalla has much more data than Foursquare.

Illustrated in Fig. 4, we analysis changes of weight w_d. This parameter is also updated by SGD and we preserve the value of w_d after each epoch. (1) On both datasets, w_d changes from a large value to a smaller one. Because search space of sequential preference is much bigger than that of spatial preference, it is not easy to obtain a good representation of sequential preference and spatial preference plays a major role in the beginning. During the later period of training, the effect of sequential preference gradually appears and w_d eventually stabilizes near a certain value, 1.1 and 2.1 on two datasets respectively. (2) Gowalla's curve is steeper than Foursquare's because the search space of sequential preference on Gowalla is obviously larger. (3) Both curves are concussion drops, not smooth ones. The relationship between the two preferences is actually complicated. We do not know which preference will play a bigger role when choosing the next POI. Therefore, when modeling each pair of two preferences, non-linear fusion tends to be a better fit rather than linear fusion.



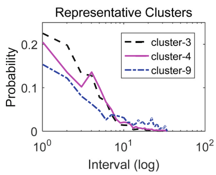
(a) Foursquare (b) Gowalla

Fig. 4. Changes of weight w_d in Eq. (6) from epoch 1 to epoch 100.

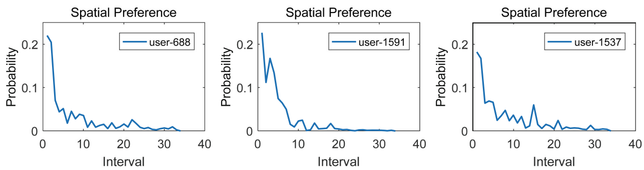
4.5 Visualization of Spatial Preference

We make visualization to study different spatial preferences on Foursquare. At the end of each user's training sequence, we compute each user's spatial preference $\boldsymbol{s}^n$ for the test set. We choose Distance2Pre (Non-Linear) as the sample. Because we have $M_D = 5\,\text{km}$, $\delta d = 0.15\,\text{km}$ in this network, each $\boldsymbol{s}^n$ is a 34-dimensional vector and the horizontal axis length is 34. First, we convert vector $\boldsymbol{s}^n$ by softmax to cause the sum of it to be 1. Then, based on all spatial preferences, we obtain 10 clusters by the k-means method. Multiple vectors of spatial preferences within one cluster are reduced to one vector by averaging. We illustrate three representative clusters cluster-[3, 4, 9] in Fig. 5(a). In order to distinguish three curves, horizontal axis uses log coordinate in Fig. 5(a). Besides, we select a user from each cluster to show its own spatial preference in Fig. 5(b) and draw his historical POIs in Fig. 5(c).

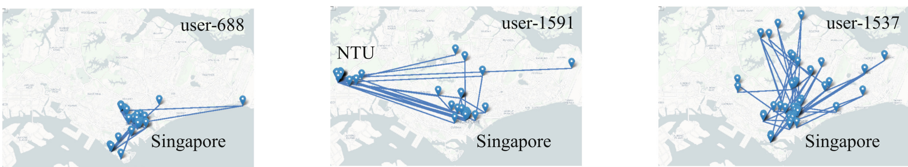
Different groups of people may have different moving patterns. Cluster-3 has large probabilities for small intervals and the probability reduces rapidly with the increase of interval. This pattern is likely to be around a point. User-688 may be a retired people and his POIs are almost around the center of Singapore. In cluster-4, there are two crests. Probabilities on large intervals are also almost zero. Such POIs of a user are mainly distributed around two points. POIs of user-1591 focus on the Nanyang Technological University and the center of Singapore. She might be a student. Cluster-9 is obviously different from cluster-3/4 because it has probabilities for many large intervals. These users may often need to go to different places for business. User-1537 prefers the center of Singapore but he also goes everywhere. By clustering, we find that patterns of movement are personalized. By looking at users one by one, the learned spatial preference in our network can effectively reflect the distribution of user's historical POIs.



(a) Three clusters.



(b) Three users' spatial preferences.



(c) Three users' check-in sequences on the map.

Fig. 5. Visualization of spatial preference on the test set. (a) shows three representative clusters. (b) are three users' spatial preferences sequentially chosen from each cluster. (c) are three users' historical POIs. Each straight line links two successive POIs and each node is a POI located by its longitude and latitude.

5 Conclusion

In this work, we have proposed a Distance2Pre network for the next POI prediction. It can mine spatial preference to model the correlation of the user-distance. Besides, we propose two preference encoders which are a linear fusion and a non-linear fusion. Both encoders can capture the relationship between two preferences and the non-linear fusion is better. Experiments demonstrate the effectiveness of our network. In the future, we will incorporate more information, like the time of check-ins and time interval.

Acknowledgment. This work is jointly supported by National Natural Science Foundation of China (61772528), National Key Research and Development Program (2016YFB1001000), National Natural Science Foundation of China (U1435221).

References

1. Cheng, C., Yang, H., Lyu, M.R., King, I.: Where you like to go next: successive point-of-interest recommendation. In: IJCAI, vol. 13, pp. 2605–2611 (2013)
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP, pp. 1724–1734 (2014)
3. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: embedding event history to vector. In: SIGKDD, pp. 1555–1564. ACM (2016)
4. Feng, S., Cong, G., An, B., Chee, Y.M.: POI2Vec: geographical latent representation for predicting future visitors. In: AAAI, pp. 102–108 (2017)
5. 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)
6. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)
7. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: SIGIR, pp. 549–558. ACM (2016)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Jing, H., Smola, A.J.: Neural survival recommender. In: WSDM, pp. 515–524. ACM (2017)
10. Kapoor, K., Sun, M., Srivastava, J., Ye, T.: A hazard based approach to user return time prediction. In: SIGKDD, pp. 1719–1728. ACM (2014)
11. 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: SIGIR, pp. 433–442. ACM (2015)
12. Liu, Q., Wu, S., Wang, D., Li, Z., Wang, L.: Context-aware sequential recommendation. In: ICDM, pp. 1053–1058 (2016)
13. 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)
14. Liu, X., Liu, Y., Li, X.: Exploring the context of locations for personalized location recommendations. In: IJCAI, pp. 1188–1194 (2016)
15. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: NIPS, pp. 2204–2212 (2014)

16. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)
17. Wang, H., Shen, H., Ouyang, W., Cheng, X.: Exploiting poi-specific geographical influence for point-of-interest recommendation. In: IJCAI, pp. 3877–3883 (2018)
18. Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., Liu, W.: Attention-based transactional context embedding for next-item recommendation. In: AAAI, pp. 2532–2539 (2018)
19. Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N.M.: Time-aware point-of-interest recommendation. In: SIGIR, pp. 363–372. ACM (2013)
20. Zhao, S., Zhao, T., King, I., Lyu, M.R.: Geo-teaser: geo-temporal sequential embedding rank for point-of-interest recommendation. In: WWW, pp. 153–162 (2017)