DAPred: Dynamic Attention Location Prediction with Long-Short Term Movement Regularity

Jiayi Liu,[1]* Quan Yuan,[2]* Carl Yang,[3] He Huang,[4] Chao Zhang,[5]* Philip Yu,[4]
[1]Purdue University, [2]Facebook, [3]University of Illinois at Urbana-Champaign, [4]University of Illinois at Chicago, [5]Georgia Institute of Technology
liu2861@purdue.edu, qyuan@fb.com, jiyang3@illinois.edu, hehuang@uic.edu, chaozhang@gatech.edu, psyu@uic.edu

Abstract

Predicting users' future locations has become an important task in various aspects, such as ride-sharing, tourism recommendation and urban planning. However, existing methods disregard that users' interest over next location is dynamic. The dynamic preference over next location involves two aspects: First, preference over distance is dynamic when users move; Second, preference over related terms vary on different target times. Hence, directly predicting next location with static network would result in unsatisfactory accuracies. Dynamic attention location prediction problem is still open now.

We propose a multilayer recurrent attention model DAPred to solve the problem. The effectiveness of DAPred is underpinned by the following reasons: (1) An embedding recurrent module to map history movements into latent place, which helps build the attention module for the following layers; (2) A historical attention module that detects multiple distance preference from dynamic movement history; (3) A prediction module for learning different weights on different time gaps. Compared to the state-of-art baselines, DAPred reaches 49.8% improvement in hitting ratio accuracy, and 18.5% improvement in average distance predictor error on three real-life datasets.

Introduction

Location prediction is a task on predicting users' movements based on their preceding GPS trace. Location prediction has many applications in real life. For example, in car-pool services, location prediction is helpful in selecting the pick up locations and destination based on the time user report, using an optimized strategy and planning. Another example is logistic planning and location-aware recommendation. The advertisements are precisely-targeted for individuals based on their past movements. Location prediction is also beneficial for urban planning and traffic jam prediction for governments. In the past, the obstacle in location prediction is the lack of data source. Recent years, with the burst of geo-annotated social media data, such as Foursquare, Facebook, Twitter(Yuan et al. 2017; Cho, Myers, and Leskovec 2011; Yuan et al. 2013; Zhang et al. 2016; Yao et al. 2017; Zhao et al. 2016), location prediction has been made possible.

Existing studies predict locations in a static way without realizing that people's preferences may change with time and their past movements. To begin with, when involved in time, both users' long-term and short-term preferences should be considered for location prediction. Most studies only consider long-term preferences for location prediction (Yao et al. 2017; Zhang et al. 2016; Feng et al. 2018). As short-term preferences indicate users' current interest, studies which lack short-term preference modeling fail to capture users' current interest, resulting in unsatisfactory prediction accuracies. Although some works proposed to predict location with both long and short-term preferences by introducing long trajectories and short trajectories (Yang et al. 2017; Feng et al. 2018), they still failed to realize that users preferences would change dynamically when they move.

We study the problem of DALP(*Dynamic Attention Location Prediction*) by discussing users' dynamic preferences. Figure 1 shows an example. Given a user, we aim to predict his or her location at different target times. In contrast to previous studies, we predict users' future locations through a dynamic attention model. For example, the user's distance preference would change when he or she moves. At different times and locations, user's would show various sensitivity to distance. Additionally, for different target times, the inffluence of previous movements differ. When it is 6pm, user would rely more on the short-term preference at 3pm, while for 10pm, long-term preference would take an advantage.

We propose DAPred (**D**ynamic **A**ttention Location **Pred**iction) to solve DALP problem. While the idea of DALP is intuitive by itself, three key challenges need to be addressed. First, *integrating diverse types of user preferences.* Users' movement preferences involves multiple terms: long-term preference and short-term preference. Before mining the users' dynamic preferences, exploiting the difference and correlation between users' long-term/short-term preferences and effectively integrating them are challenging. Second, *the dynamic influence of past movements over distance preference.* Users' past movements may have an influence on the distance selection of next movement. For example, if users keep moving in a long time, they would feel tired to visit a place far away. Extracting the dynamic influence of records over next movement distance is nontrivial. Third, *the influence of target time over user's preference over terms.* The target time could not be considered as a simple factor, it should interact with other factors. For example, if the gap time between current time and target time is short (e.g., 15 min), users tends to visit nearby locations that can meet their short-term interests, while if the gap time is long (e.g., 5 hours), a user's next visit is influenced more by their long-term interests.

DAPred adopts three modules to tackle the above challenges. The first module is an embedding-recurrent module to integrate terms into latent place and capture transitions. Then, DAPred employs the historical attention module to discover the movement influence over distance preference. Finally, DAPred set the prediction module to capture different interests over terms on different time gaps.

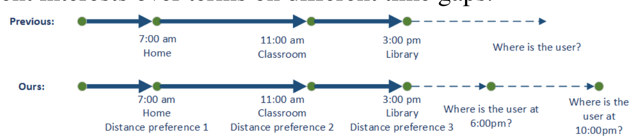


Figure 1: A comparison between next location prediction and dynamic attention location prediction

In summary, our major contributions are summarized as follows:

1. We propose dynamic attention location prediction(DALP) problem, where users' location preferences would change dynamically when they move.
2. We propose a long-short memory enriched attention recurrent model DAPred to solve the DALP problem. DAPred learns the users' dynamic preferences in two terms: first, users' preferences over distance is dynamic when they move; second, users' long/short-term preferences are dynamic when target times are different.
3. We have tested our algorithm on three large-scale geotagged tweet datasets: Foursquare, Gowalla New York and Gowalla Los Angeles. Comparing with other algorithms, DAPred improves the accuracy significantly.

Related Work

In this section, we review existing works related to our problem, including: (1) Long-term movement prediction; (2) Short-term movement prediction; (3) Long-term and short-term movement prediction.

Long-term location prediction

Most studies (Noulas et al. 2012; Cheng et al. 2013; Lian et al. 2015; Monreale et al. 2009; Yang et al. 2017; Yao et al. 2017; Liu et al. 2017) predict locations based on users' long-term behavior. For those who use only long-term behavior, we call them long-term location prediction.

Generally, existing model-based long-term location prediction could be classified into two categories: HMM model and RNN model. HMM model use the Hidden Markov Model to predict users' locations(Chen, Liu, and Yu 2014; Killijian 2012; Mathew, Raposo, and Martins 2012). Their transition matrices consist of different factors: location type (Cheng, Ye, and Zhu 2013), personalized point-of-interest locations(Cheng et al. 2013), social/geo-distance knowledge for unvisited location prediction(Lian et al. 2015), grouping information(Zhang et al. 2016).

RNN model uses the recurrent network to model trajectories. By introducing recurrent network, these methods are able to capture the sequential characters for each movement(Du et al. 2016; Liu et al. 2016; Yao et al. 2017; 2018; Mei and Eisner 2016; Yao et al. 2016). Liu et al. (Liu et al. 2016) construct a temporal recurrent network with distance/time-specific transition matrices; Yao et al. (Yao et al. 2017) introduce textual information into recurrent network to improve the performance; Yao et al. (Yao et al. 2016) introduce a unified deep learning framework for mobile sensing data.

The above methods are all designed for discovering the long-term preference of users. However, there exists long-term preference and short-term preference for a user. For example, a user prefers to go to the gym from home at 6 pm, which is his or her long-term preference. When there is a mid-term exam to prepare, he or she would go to the library instead, which is the short-term preference. Hence, users' short-term preference is also an important factor which should be applied separately.

Short-term location prediction

Lots of studies separate long-term movements into pieces to fully investigate the short-term preference of users. (Choi and Hebert 2006) segment long-term trajectories into short ones and concat them again to find the noisy movements. However, this method only works in short trajectory prediction and unable to predict the next location in a wide time range. Besse et al.(Besse et al. 2017) also use the segmentation trajectories for their next location prediction. By clustering the short trajectories, they are able to predict the next location.

Although the above studies take short-term movements into consideration, they fail in integrating long-term movements together. Short-term movement and long-term movement indicate the short and long-term interest respectively. The lack of either of them would result in an unconformity with real life. Hence, the results of methods with only short-term location prediction stay unsatisfactory.

Long-term and Short-term location prediction

For long-term and short-term location prediction, most models adopt recurrent network(RNN) to model trajectories, e.g. long-term and short-term trajectories recurrent network (Yang et al. 2017), attention periodicity (Feng et al. 2018). Although they are all designed with both long-term and short-term, but they still fail to solve this problem: how to predict the locations at different target time? While many studies take the history timestamp as an important feature for location prediction, they neglect the future target time as an important input. They could only predict next location in the future but unable to predict future locations with different target time stamps(e.g. 8 pm at the library and 6 pm at the restaurant). Here we propose the problem as target-time location prediction, which is designed for predicting multiple future locations at different time stamps.

Preliminaries

In this section, we formulate the dynamic attention location prediction problem, and explore its characteristics, which motivate the design of DAPred.

Problem Definition

For users' check-ins, we denote them as $\boldsymbol{R_u}$. Let $\boldsymbol{R_u} = (r_1, r_2, ...r_n)$ be sequential records in chronological order for user u. Each record r_i is a tuple of $< l_{r_i}, t_{r_i}, u >$, where l_{r_i} is the geo-location, t_{r_i} is the post time and u is the user id. Given records $\boldsymbol{R_u}$, we aim to predict the locations users u would be at multiple future time stamps with a dynamic framework. To construct the dynmaic framework, we transform $\boldsymbol{R_u}$ into long trajectory and short trajectory to represent user's long-term and short-term interests relatively (Yang et al. 2017).

Definition 0.1 (Long Trajectory) *For a user u, long trajectory $\boldsymbol{L_u}$ is his or her whole movement history. Here,*

$$\boldsymbol{L_u} = [l_{r_1} \quad l_{r_2} \quad ... \quad l_{r_n}] \tag{1}$$

where l_{r_k} represents the location of the k-th record.

Definition 0.2 (Short Trajectories) *For a user u, short trajectories $\boldsymbol{S_u}$ are his or her fragmented movements. We split long trajectory into a sequence of short ones if the gap time between consecutive visits is greater than a threshold(e.g., 6 hours), then we would cut them into different group.*

Formally,

$$\boldsymbol{S_u} = \begin{Bmatrix} [l_{r_1} & l_{r_2} & ... & l_{r_{i-1}}] \\ [l_{r_i} & l_{r_{i+2}} & ... & l_{r_j}] \\ [... & ... & ... & ...] \end{Bmatrix} \tag{2}$$

Note that, the length of each short trajectory could be different.

Definition 0.3 (Time) *Given a user u with records $\boldsymbol{R_u}$, the time stamps for each records are $\boldsymbol{T_u} = [t_{r_1}, t_{r_2}, ...t_{r_n}]$.*

The Overall Architecture

In a nutshell, DAPred embeds all terms into a latent space, and uses recurrent network to capture the sequential information. By using attention mechanism, DAPred chooses what to pay attention with based on different timestamps.

The intuition behind our architecture is: Users' interests over next location are dynamic in two terms: 1) Users' preferences over distance vary when users move. Comparing to the users move in a relatively short route, those who move a long route would be more sensitive to distance. 2) The next locations users would visit differ with target times. Different target times would influence the users' preference over long/short-term memory and distance. For example, when the time gap between target time and current time is small, users tend to make their decision based on distance and short-term memory, otherwise, their preference would be more on long-term memory.

To construct our model, our steps are as following: Given a user u, we transform the records $\boldsymbol{R_u}$ into $\boldsymbol{L_u}, \boldsymbol{S_u}$ and $\boldsymbol{T_u}$. As mentioned earlier, dynamic attention location prediction still poses several challenges: 1) How to introduce long-term and short-term interests for dynamic attention location prediction? 2) How to detect the dynamic attention over distance preferences when users move? 3) For different target times, how to apply dynamic preferences over multiple terms? In the following, we introduce embedding-recurrent, historical attention and prediction modules to address the three challenges above respectively. Figure 2 shows a concise architecture of our model.

Proposed Method

DAPred aims to predict users' location dynamically. To this end, we exploit users' movements and spatial-temporal features in a unified framework, which consists of embedding-recurrent module, historical attention module and prediction module.

Embedding-Recurrent Module

Embedding-recurrent network has been investigated in lots of studies. In this section, we would not introduce the detailed procedures about how and why embedding-recurrent network works, instead, we discuss how long-term/short-term interests work in dynamic attention location prediction(Yang et al. 2017).

Multimodal embedding jointly maps all the features into latent space. To represent users' mobility, exisiting algorithms only embed three terms: long trajectory $\boldsymbol{L_u}$, time stamps T_u and user u, which represent movements, time and user's personal preferences respectively. Nevertheless, this strategy is problematic because long trajectories $\boldsymbol{L_u}$ could not reveal the user's short-term preference over mobility. Motivated by (Yang et al. 2017), we add short trajectories $\boldsymbol{S_u}$, which are fragmented from long trajectory $\boldsymbol{L_u}$, to uncover the short-term preferences of users. To help model the transitional relationship between the above four features, we design the multimodal embedding module to jointly embed them. Then, we get $\boldsymbol{e_{st}}, \boldsymbol{e_{lt}}, \boldsymbol{e_t}$ and $\boldsymbol{e_u}$ as the embedding results of short trajectories, long trajectory, times and user.

Given $\boldsymbol{e_{st}}, \boldsymbol{e_{lt}}$, we then capture their sequential information through GRU and RNN respectively and obtain the hidden states of each step i, denoting $\boldsymbol{hl^{(i)}}$ and $\boldsymbol{hs^{(i)}}$. Motivated by previous work on long/short-term location prediction(Yang et al. 2017), our choice of GRU for long-term is by its effectiveness of memorizing and learning long-term hidden state dependencies. Similarly, the adoption of RNN for short-term trajectories is based on its to model sequential data in a short time window .

Historical Attention Module

The historical attention module helps learn users' dynamic interest over locations. Instead of keeping a static interest over time (Yao et al. 2017; Zhang et al. 2016; Feng et al. 2018), we model users' dynamic preferences over distance when they move. The key idea is to find the attention of distance preference on previous records (Luong, Pham, and Manning 2015; Sutskever, Vinyals, and Le 2014). However, it would be too expensive to directly iterate over all possible locations. To alleviate this problem, for each location

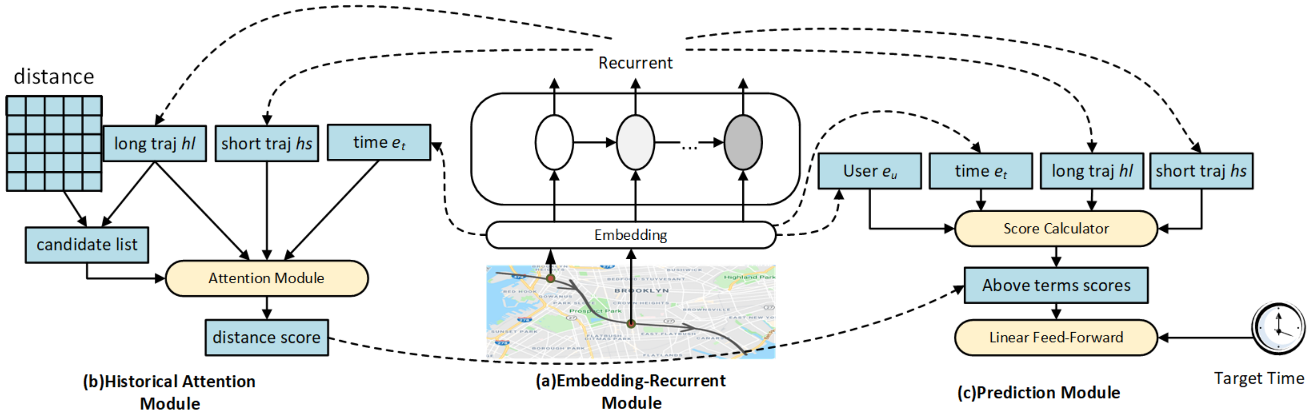


Figure 2: The overall architecture. The inputs consist of users and all their check-ins. The outputs are the probability lists of locations for different target time. (a) Embedding-Recurrent Module uses embedding and RNN/GRU to encode inputs for other modules. (b) Historical Attention Module leverages embedding-recurrent results of trajectories and time to learn users' preference over distance. (c) Prediction Module learn the locations' possiblities by capturing all previous modules' results.

l_{r_i}, we generate the next-location candidate list $F_{l_{r_i}}$ from the whole location set based on geo-distance and previous records. That is, we first estimate the probablity of locations to be chosen as candidates:

$$dis(l_k|l_{r_i}) = \frac{e^{-d(l_k,l_{r_i})}}{\sum_{k=1}^{n} e^{-d(l_k,l_{r_i})}} \tag{3}$$

$$pop(l_k|l_{r_i}) = \frac{f(l_k \wedge l_{r_i})}{f(l_{r_i})} \tag{4}$$

In which $d(l_k, l_{r_i})$ is the distance between l_k and l_{r_i}, n is the total number of locations, $f(\cdot)$ is the frequency.

After we select candidates for l_{r_i}, we introduce the distance-aware attention module to find the correlation between mobility and distance preferences. When involved with distance, previous studies usually concat distance factors with others(Liu et al. 2016). However, users' preference over distance may change when they move. Thus, directly concating these factors lacks in discovering the intrinsic interaction between distance and other factors.

Inspired by human attention mechanism, we develop a distance-aware attention module to solve the problem. The detailed flow of this layer is shown in Figure 3. The at-

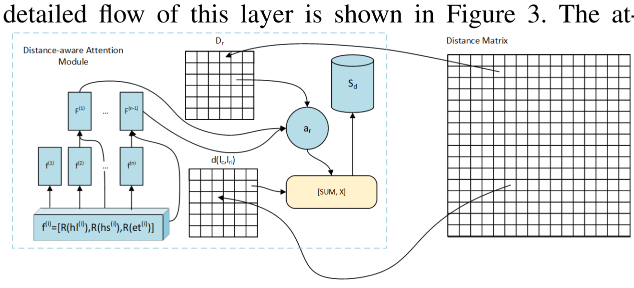


Figure 3: The detailed architecture of Distance-aware Attention Module

tention model aims at summarizing the influence of movement history over users' distance preference. To begin with, the influence containing two parts: 1) Trajectory and time. 2) Distance. For trajectory and time, we adopt the results from embedding-recurrent network. For distance, we select the geo-distance between next location and other locations in trajectory to measure the influence. Let $\boldsymbol{f^{(i)}} = [\mathcal{R}(\boldsymbol{hl^{(i)}}), \mathcal{R}(\boldsymbol{hs^{(i)}}), \mathcal{R}(\boldsymbol{e_t^{(i)}})]$ be the concatenation of embedding result $\boldsymbol{e_t^{(i)}}$, long-term GRU result $\boldsymbol{hl^{(i)}}$ and short-term RNN result $\boldsymbol{hs^{(i)}}$ at $i\!-\!th$ step, in which, $\mathcal{R}(\cdot)$ indicates the ReLU funct(i)on: $\mathcal{R}(x) = x^+$.

Formally, for record r_i, given the embedding-recurrent feature $f^{(i)}$ at a time step t_i and location l_i, to capture its attentional score, we consider the correlations between its previous step and all movement history. Then we define the attentional score as:

$$\boldsymbol{F_{r_i}} = \boldsymbol{f^{(i+1)}} \cdot \left[\boldsymbol{f^{(1)}} \quad \boldsymbol{f^{(2)}} \quad \dots \quad \boldsymbol{f^{(n)}}\right]^T \tag{5}$$

$$\boldsymbol{D_{r_i}} = [d(l_{r_{i+1}}, l_{r_1}) \quad d(l_{r_{i+1}}, l_{r_2}) \quad \dots \quad d(l_{r_{i+1}}, l_{r_n})] \tag{6}$$

Then we encode $\boldsymbol{F_{r_i}}$ and $\boldsymbol{D_{r_i}}$ to get the attentional vector over distance.

$$\boldsymbol{a_{r_i}} = \boldsymbol{W_a} \cdot \boldsymbol{F_{r_i}} + \boldsymbol{W_b} \cdot \boldsymbol{D_{r_i}} + \boldsymbol{C} \tag{7}$$

where $\boldsymbol{W_a}, \boldsymbol{W_b}, \boldsymbol{C}$ respectively represent the weight for past history, geo-distance and bias.

After obtaining the attentional vector $\boldsymbol{a_{r_i}}$, our goal is to optimize the distance scores of next-location candidates such that we could predict users next target-time location based on their movement history and the geo-distance between locations. Suppose the current location is l_{r_i}, for a location candidate l_c in the candidates list $F_{l_{r_i}}$, we define its distance score $\boldsymbol{S_d^{l_c}}$ as:

$$\boldsymbol{S_d^{l_c}} = \sigma(\boldsymbol{a_{r_i}}) \cdot [e^{-d(l_c,l_{r_1})} \quad e^{-d(l_c,l_{r_2})} \dots e^{-d(l_c,l_{r_n})}]^T \tag{8}$$

where, $\sigma(\cdot)$ is a normalization. The intuition behind this formula is that: the likelihood of a candidate to be chosen is based on two aspects: 1) The attention vector, which represents the history preference over next location. 2) The distance between this candidate and the locations user visits. We apply attention vector on distance preference to indicate the possibility of this candidate to be chosen.

Prediction Module

The prediction module aims to predict the next locations at different target times. By using a linear feed-forward neural network, we concatenate embedding-recurrent scores above and distance score together. Note that, for different time gaps, we train different weights over feed-forward neural network (Yuan et al. 2017). Here, we set 1 hour as the smallest unit to deal with underfit. For those with decimal, we adopt the following strategy: Given a gap time gt, suppose its upper bound is $gt_u = ceil(gt)$ and its lower bound is $gt_l = floor(gt)$ the weight w_{gt} of its linear network would be:

$$w_{gt} = w_{gt_u} \cdot (gt_u - gt) + w_{gt_l} \cdot (gt_l - gt) \quad (9)$$

where w_{gt_u} is the weight at gt_u and w_{gt_l} is the weight at gt_l.

Finally, for different time gap, we obtain their own preferences over the terms by:

$$S_l = [\boldsymbol{S_t} \;\; \boldsymbol{S_u} \;\; \boldsymbol{S_{hs}} \;\; \boldsymbol{S_{hl}} \;\; \boldsymbol{S_d}]^T \cdot \sum_{1}^{n} w_{gt} \mathrm{L}(\mathrm{W_L}, \mathrm{b_L}) \quad (10)$$

where $S_t, S_u, S_{hs}, S_{hl}, S_d$ indicate the prediction score for time, user, short trajectories, long trajectories and distance, $\mathrm{L}(\mathrm{W_L}, \mathrm{b_L})$ refers to the linear network for a time gap, W_L and b_l are the weight and bias for this network.

Experiment

Experiment Setting

Datasets Our experiments are based on three datasets: Foursquare, Gowalla New York and Gowalla Los Angeles. The source of Foursquare dataset is the same as (Yao et al. 2017; Zhang et al. 2014). Gowalla dataset is a location-based social networking website similar to Facebook, the source of this dataset is the same as (Cho, Myers, and Leskovec 2011). For foursquare dataset, it consists of 1.4 million check-ins from 2009-01 to 2012-01 in New York City. For Gowalla dataset, it consists of 1.95 million check-ins in New York and 3.33 million check-ins in Los Angeles from 2009-01 to 2012-01. For each dataset, we firstly merge records with the same user to set trajectories for users. Then, we remove the users with less than 5 records and the locations with less than 10 records(Yao et al. 2017; Feng et al. 2018). This operation guarantees that each trajectory is long enough to be cut into the training set and testing set.

After such preprocessing, for Foursquare dataset, we obtain 500 users, 3555 locations and 9968 trajectories in training set, 2829 trajectories in the testing set. For Gowalla dataset at New York, we obtain 500 users, 5670 locations and 5019 trajectories in training set, 22851 trajectories in the testing set. For Gowalla dataset at Los Angeles, we obtain 404 users, 777 locations and 18405 trajectories in training set, 5202 trajectories in the testing set.[1]

Experimental Protocol For each dataset, we randomly select 70% records of users for training, 10% for tunning and the remaining records for testing. To evaluate the performance of each method, we use the hitting ratio @k and average distance error δ_d. Hitting ratio is the percentage of the ground-truth location appears in our top-k location result list, average distance error is the average distance between our top-1 prediction and ground-truth. These two evaluation methods are the same as (Feng et al. 2018; Yao et al. 2017).

Parameter Settings DAPred owns the following major parameters: (1) For the embedding layer, the latent dimension D_v for both long and short trajectories, D_t for time and D_u for users. (2) For the recurrent layer, the recurrent dimension D_h for both RNN and GRU. (3) For attention layer, the number of candidates N. (4) The dropout probability O (5) The batch size of minibatch M. After tuning, we set D_v as 16, D_t as 8, D_u as 32, D_h as 16, N as 16, O as 0.5, and M as 50. We tested on various parameter settings and did not find much difference, the details of our tuning process would be discussed in .

Metrics As aforementioned, we use the ground truth locations in the remaining 20% testing data to evaluate all methods. To quantify the performance of all the methods, we use the hitting ratio @k as our criteria. Here, hitting ratio refers to the percentage of ground truth appears in our top-k list. Here, we present the hitting ratio of top 1, top 5 and top 10. The other criterion is δ_d, which is the average geographical distance between the ground-truth location and the top-1 prediction. The metrics we adopt are same as previous work (Yao et al. 2017; Feng et al. 2018).

Comparisons to the State-of-the-Art We compare DAPred with the following methods: (1)DSSM(Huang et al. 2013) (2)JNTM(Yang et al. 2017) (3)ST-RNN(Liu et al. 2016) (4)SERM*(Yao et al. 2017) (5)DeepMove(Feng et al. 2018).

Quantitative Results

Table 1: Performance Comparison between DAPred and STRNN, DSSM, JNTM, SERM*, DeepMove. HR is the hitting ratio, δ_d is the average distance predictor error.

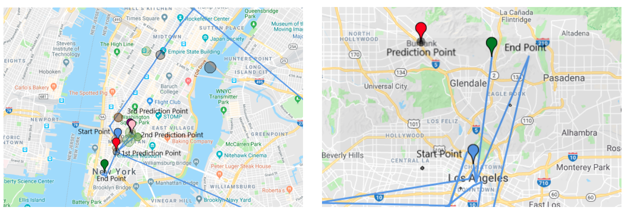
Data	Method	HR@1	HR@5	HR@10	$\delta_d(^\circ)$
4SQ	STRNN	0.016	0.054	0.083	0.058
	DSSM	0.128	0.245	0.286	1.010
	JNTM	0.060	0.121	0.156	1.099
	SERM*	0.137	0.353	0.486	0.046
	DeepMove	0.148	0.306	0.352	0.060
	DAPred	**0.191**	**0.524**	**0.801**	**0.040**
GNY	STRNN	0.000	0.004	0.008	0.061
	DSSM	0.052	0.110	0.141	0.995
	JNTM	0.038	0.092	0.126	1.063
	SERM*	0.098	0.185	0.306	0.057
	DeepMove	0.100	0.248	0.313	0.053
	DAPred	**0.146**	**0.362**	**0.678**	**0.030**
GLA	STRNN	0.000	0.004	0.008	0.128
	DSSM	0.066	0.145	0.188	1.133
	JNTM	0.020	0.061	0.086	1.158
	SERM*	0.126	0.223	0.440	**0.086**
	DeepMove	**0.198**	0.262	0.565	0.090
	DAPred	0.182	**0.289**	**0.619**	0.088

Table 1 reports the performance comparison of our methods and the State-of-the-Art algorithms on our three datasets. In Foursquare(4SQ) dataset, compared to the best baseline DeepMove, DAPred yields around 29.1% improvement in top 1 hitting ratio, 71.2% improvement in top 5 hitting ratio and 127.6% improvement in top 10 hitting ratio.

[1] **The data and code will be publicly available if accepted. We could share it privately if reviewers ask for.**

In Gowalla New York(GNY) dataset, compared to the best baseline DeepMove, DAPred yields around 46.0% improvement in top1 and top 5 hitting ratio and 116.6% improvement in top 10 hitting ratio. In Gowalla Los Angeles(GLA) dataset, DAPred yields around 10.3% improvement in top 5 hitting ratio and 9.6% improvement in top 10 hitting ratio. For the distance predictor error δ_d, DAPred outperfroms the best baseline SERM* by 13% in Foursquare and DeepMove by 43.3% in Gowalla at New York. Compared to the strongest baseline DeepMove, the huge improvements in DAPred are mainly attributed to two main reasons: (1) The dynamic preferences over distance when users move; (2) The introduction of various interests over all terms on different target times. Related results for these two conclusions would be shown in Figure 5 and Figure 6.

Illustrative Cases



(a) Location Prediction at New York

(b) Los Angeles

Figure 4: Visualization of dynamic attention location prediction with different interest over long-term and short-term prefernce.

In this section, we present several illustrative cases for our algorithm. Figure 4 shows the location prediction for three different users from Foursquare dataset, GowallaNY dataset, and GowallaLA dataset. In these figures, the red markers indicate the ground truth of next location and the blue lines are users' trajectories. The blue markers indicate the start points and the green markers indicate the endpoints. Black circles in the figures refer to the results of our predictions. A larger circle indicates a higher ranking of the prediction.

The movements in figure 4(b) are based at Los Angeles with single target time. The prediction results of this figure show a preference of users on long-term preference. Here, we predict BurBank to be the location user would visit. Neither does BurBank appear in past trajectory, nor does it close to the visited locations, which indicate a long-term preference.

The movements in figure 4(a) are based at New York with multiple target time. In which, the red marker is the true location of the first target time, the black circles are the first target time predicting results; the pink marker and brown circles are for the second target time; the purple marker and green circles are for the third target time. To further understand the influence of gap time over the preference for long-term and short-term memory, we may look at figure 4(a) again. For the first target time with black circles indicate our prediction, we find the black circles are almost around the short trajectory, showing a preference for short-term memory. For the second target time with green circles, they are almost around the endpoint, indicating a preference over locations in the neighborhood. While for the third target time with brown circles, most of the predicted locations are far away, showing a preference on long-term memory.

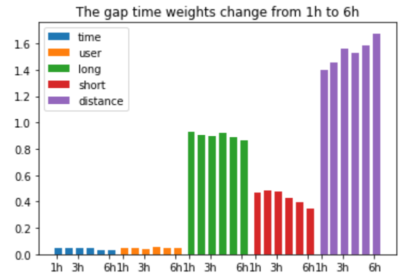


Figure 5: The change of weights with gap times on Foursquare

Effect of dynamic attention mechanism

As we discussed above, we attribute our improvements to two aspects. In this section, we would further discuss the effects of these two parts.

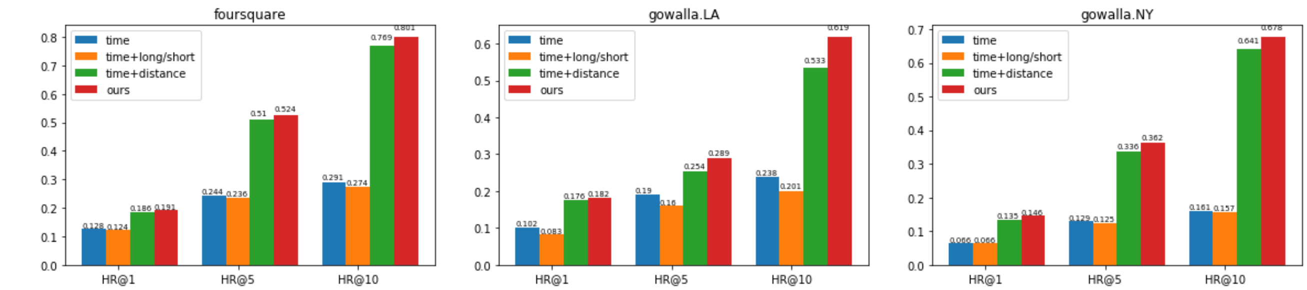
To begin with, in Figure 5, we present the change of weights on different time gaps (from 1 hour to 6 hours). When the time gap becomes larger, the weight of distance tends to grow, and the weight of short-term preference tend to decrease. Such a phenomenon further validates the change of users' preference over target time. Due to limitations of space, we don't present the results for all datasets

We discussed the influence of past movements over distance preference in Figure 6. Since there are no previous works analyze the influence of past movements over distance preference, we are interested in how DAPred could improve the performance in this aspect. In Figure 6, we compare the accuracy among model with only time, model with the time and long/short trajectory, model with the attentional effects of time over distance, and the proposed DAPred (*the attentional effects of time and long/short preference over distance*). From the figure, we could find draw the conclusion that the introduction of dynamic attentional preference improves the prediction accuracy in all three datasets.

Parameter Tuning

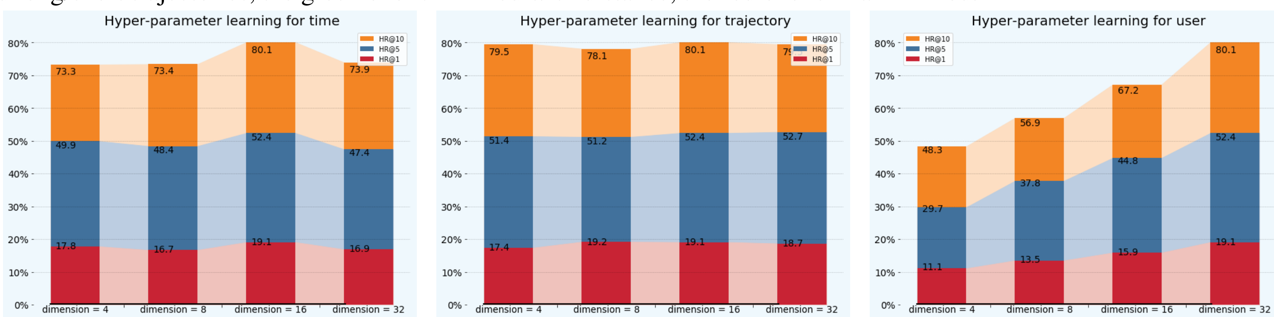
As mentioned above, there are three main parameters of DAPred: the embedding demension for long-term and short-term trajectory E_v, the embedding demension for time E_t and the embedding demension for user E_u.

We first study the effects of E_t and E_v. In Figure 7, the orange bar implies the accuracy of HR@10, the blue bar implies the accuracy of HR@5, while the red bar implies the accuracy of HR@1. From the figure, we could find that the when $E_t = 16$ and $E_v = 16$, we could reach the highest accuracy. We could also draw the conclusion that the effects of parameters of E_t and E_v are limited on accuracy. Then we study the effects of E_u. Comparing to the parameters above, the value of E_u influence the accuracy a lot. Based on the figure, we select $E_u = 32$.



(a) Test on Foursquare (b) Test on Gowalla.NY (c) Test on Gowalla.LA

Figure 6: Comparison among variation of DAPred: The blue one is DAPred with only time, orange one is DAPred with time and long/short trajectories, the green one is DAPred with distance, the red one is final DAPred.



(a) Time dimension (b) Trajectory dimension (c) User dimension

Figure 7: Hyperparameter Tuning

Conclusion

In this paper, we studied the problem of dynamic attention location prediction problem with a new algorithm DAPred. To the best of our knowledge, DAPred is the first method to predict next location with multiple target time. To solve the target-time aware location prediction problem, DAPred enjoys two novel characteristics: 1) An attentional module to model the temporal and historical movements influence over next movement selection 2) Various target time preference over multiple factors. Our extensive experiments with three real-life datasets have proved that DAPred owns a significant improvement over the accuracy of the state-of-the-art method in terms of HR@1, HR@5, HR@10 and average distance predictor error. Further more, we also conduct comparison between different models origined from our algorithm, which further proved the significance of dynamic attention on time.

As part of our future work, we plan to discuss more on the dynamic attention location prediction problem. We would incorporate our dynamic location prediction with inner-purpose in time cycle (e.g. habits, travels, etc).

References

Besse, P. C.; Guillouet, B.; Loubes, J.-M.; and Royer, F. 2017. Destination prediction by trajectory distribution-based model. *IEEE Transactions on Intelligent Transportation Systems* (99):1–12.

Chen, M.; Liu, Y.; and Yu, X. 2014. Nlpmm: A next location predictor with markov modeling. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 186–197.

Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, 2605–2611.

Cheng, H.; Ye, J.; and Zhu, Z. 2013. What's your next move: User activity prediction in location-based social networks. In *SDM*, 171–179. SIAM.

Cho, E.; Myers, S. A.; and Leskovec, J. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1082–1090. ACM.

Choi, P. P., and Hebert, M. 2006. Learning and predicting moving object trajectory: a piecewise trajectory segment approach. *Robotics Institute* 337.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1555–1564.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *World Wide Web Conference*, 1459–1468.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2333–2338. ACM.

Killijian, M. O. 2012. Next place prediction using mobility markov chains. In *The Workshop on Measurement, Privacy, and Mobility*, 3.

Lian, D.; Xie, X.; Zheng, V. W.; Yuan, N. J.; Zhang, F.; and Chen, E. 2015. Cepr: A collaborative exploration and periodically returning model for location prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6(1):8.

Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. Predicting the next location: a recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*, 194–200.

Liu, Q.; Wu, S.; Wang, D.; Li, Z.; and Wang, L. 2017. Context-aware sequential recommendation. In *IEEE International Conference on Data Mining*, 1053–1058.

Luong, M. T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *Computer Science*.

Mathew, W.; Raposo, R.; and Martins, B. 2012. Predicting future locations with hidden markov models. In *ACM Conference on Ubiquitous Computing*, 911–918.

Mei, H., and Eisner, J. 2016. The neural hawkes process: A neurally self-modulating multivariate point process.

Monreale, A.; Pinelli, F.; Trasarti, R.; and Giannotti, F. 2009. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 637–646. ACM.

Noulas, A.; Scellato, S.; Lathia, N.; and Mascolo, C. 2012. Mining user mobility features for next place prediction in location-based services. In *Data mining (ICDM), 2012 IEEE 12th international conference on*, 1038–1043. IEEE.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. 4:3104–3112.

Yang, C.; Sun, M.; Zhao, W. X.; Liu, Z.; and Chang, E. Y. 2017. A neural network approach to jointly modeling social networks and mobile trajectories. *ACM Transactions on Information Systems (TOIS)* 35(4):36.

Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; and Abdelzaher, T. 2016. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. 351–360.

Yao, D.; Zhang, C.; Huang, J.; and Bi, J. 2017. Serm: A recurrent model for next location prediction in semantic trajectories. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2411–2414. ACM.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction.

Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; and Thalmann, N. M. 2013. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 605–613. ACM.

Yuan, Q.; Zhang, W.; Zhang, C.; Geng, X.; Cong, G.; and Han, J. 2017. Pred: periodic region detection for mobility modeling of social media users. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 263–272. ACM.

Zhang, C.; Han, J.; Shou, L.; Lu, J.; and Porta, T. L. 2014. Splitter: mining fine-grained sequential patterns in semantic trajectories. *Proceedings of the Vldb Endowment* 7(9):769–780.

Zhang, C.; Zhang, K.; Yuan, Q.; Zhang, L.; Hanratty, T.; and Han, J. 2016. Gmove: Group-level mobility modeling using geo-tagged social media. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1305–1314. ACM.

Zhao, S.; Zhao, T.; King, I.; and Lyu, M. R. 2016. Gt-seer: Geo-temporal sequential embedding rank for point-of-interest recommendation.