ICIC Express Letters
Part B: Applications
Volume **8**, Number **6**, June **2017**

ICIC International ©2017 ISSN 2185-2766
pp. **919–925**

INDOOR DESTINATION PREDICTION BASED ON BIDIRECTIONAL RECURRENT NEURAL NETWORK

Shengsheng Wang and Qing Yue

College of Computer Science and Technology
Jilin University
No. 2699, Qianjin Street, Changchun 130012, P. R. China
wss@jlu.edu.cn; yueqing15@mails.jlu.edu.cn

Received December 2016; accepted March 2017

Abstract. *Destination prediction has been widely concerned in recent years, and it is also important for other technologies such as targeted advertising and recommending sightseeing places. However, destination prediction for indoor moving object is still a big challenge because indoor space structure and data sparsity make it more complex than outdoor destination prediction. Therefore, we address a new scheme for indoor destination prediction based on bidirectional recurrent neural networks. In our approach, the history trajectory dataset obtained by indoor positioning technology is used to preprocess indoor space and calibrate sampling points, and an improved bidirectional recurrent neural networks is used to predict destination by calibrated sampling points. Experiments based on real datasets show that the proposed method can predict destinations more favorably than others.*
Keywords: Wi-Fi positioning technology, Indoor moving object, Destination prediction

1. **Introduction.** With the development of positioning technologies, there is more and more location information being recorded. It makes it possible to predict moving object destination. However, its current mainstream application is outdoor trajectory based on GPS positioning technology, which is not capable of continuously reporting the indoor location [1-3]. For indoor destination prediction, there are a lot of differences from that of outdoor. Indoor space is constructed by cellular space, and it contains different entities such as rooms, doors and hallways which influence the connection of indoor space [4]. Sometimes cells are connected in space, but actually they are not connected due to the walls. Therefore, it is not possible to predict destination according to distance.

Nowadays, with the development of wireless networks, much work [5-7] has been proposed for recording much more indoor positioning information, which has significantly improved the effectiveness and universality of indoor positioning system, which is necessary for moving object destination prediction. There are some methods that can be applied to predicting indoor destination. A method to predict trajectory using Markov models is proposed in [8]. In [9], a new scheme is proposed to predict destination using a Bayesian inference. A new method is proposed which establishes a theoretical model to destination prediction with a transition tensor in [10]. However, after testing the above destination prediction methods, we found that they failed to consider the data sparsity or temporal information. The data sparsity and temporal information influence the probability of different destinations. These weaknesses finally caused great deviation from real destination.

To address these issues, a scheme for indoor destination prediction based on bidirectional recurrent neural networks (BRNN) is proposed. The historical trajectory dataset is used to preprocess indoor space. Then we calibrate the input sampling trajectories to solve data sparsity. Finally, we apply the calibrated sampling points into BRNN to predicting destination.

The rest of this paper is organized as follows. In Section 2, we introduce the method of indoor space preprocessing. In Section 3, the input sampling points are calibrated. In Section 4, we apply the calibrated sampling points into improved BRNN to predicting destination. The experimental results are shown in Section 5. Finally, Section 6 draws a conclusion.

2. **Preprocessing Indoor Space.** To solve the problem of indoor space connectivity, we propose an approach to preprocessing indoor space. First, we divide the indoor space into some same size of cells and assign the cells ID (CID) represented by (x, y), and a trajectory can be shown by a sequence of CID. The connectivity is decided by historical trajectory dataset.

However, in historical trajectory dataset, there are always some sampling errors. To deal with this error, we set a point threshold. If the number of historical points in a cell is over threshold, the points are available, else are useless.

Definition 2.1. ***(Spatial neighbor)*** *If the CID of two cells satisfy the condition of* $|x - x'| \leq 1$, $|y - y'| < 1$ *or* $|x - x'| < 1$, $|y - y'| \leq 1$, *the cells are spatial neighbors (SN).*

Definition 2.2. ***(Spatial connection)*** *If the cells are spatial neighbors and there exists a trajectory connecting the two cells directly, then we consider the two cells to be spatial connection (SC).*

Definition 2.3. ***(Connection region)*** *There is a special region which connects different regions; in other words, if a cell is marked with different region IDs (RIDs), then we consider it to be a connection region (CR) such as the door.*

The main idea of preprocessing indoor space is to assign the cells which are SC to the same RID and assign different RIDs if they are SN but not SC; if a cell is assigned two different RIDs, then it should be marked CR.

Algorithm 1: Preprocess Indoor Space

Input : Historical trajectory dataset, a cellular space;
Output: A regional indoor space

$q_i \leftarrow$ the queue of a cell's spatial neighbor cells // i represents the region id
while *q_i is not empty* **do**
    remove *the cell (x, y) from q_i*;
    if *the cell has not been marked connection region* **then**
        for *each cell $\in$ spatial neighbor of the cell (x, y) and is not marked RID(r_i)* **do**
            if *cell (x', y') is spatial connection with (x, y)* **then**
                put the cell into q_i;
                if *(x', y') already has an RID* **then**
                    (x', y') will be marked CR$(cr_{i,k})$; // i, k are RID
                else
                    marked r_i;
            else
                Weaken treatment(q_i);
                $q_{i+1} \leftarrow (x', y')$;
                i++;

Algorithm 2: Weaken Treatment(q_i)

while *q_i is not empty* **do**
  remove *the cell (x, y) from q_i*;
  for *each cell is spatial neighbor of cell (x, y)*
    if *the cell is spatial connection and not marked different RID* **then**
      the cell is marked r_i;

3. **Sampling Points Calibration.** Considering the accuracy of indoor positioning technology, data sparsity and temporal information using an approach to calibrating sampling points is proposed, which is based on the historical trajectory dataset. First, we calculate the transition probability with time stamp. And then insert some cells into sampling points.

3.1. **Transition probability.** Transition probability is the probability of objects moving from one cell or region to others. If the cells are in the same region, we can directly calculate the transition probability; else we should calculate region transition probability.

For the cells in the same region, one cell just connects to its SC cells directly; therefore, we just need to calculate the probability with the SC cells. The probability is saved in a matrix M, $|T(c_{x,y} \to *)|$ is the number of the trajectories which travel from cell (x, y) to its connecting cells, and $c_{x,y}$ is the CID of (x, y), (x', y') is one of the cells connecting to (x, y).

$$P_1\left(c_{(x,y)} \to c_{(x',y')}\right) = \frac{\left|T\left(c_{(x,y)} \to c_{(x',y')}\right)\right|}{\left|T(c_{(x,y)} \to *)\right|} \tag{1}$$

At the same time, it is easy to get the 2-step probability is $M^{1:2} = M + M^2$ which is the probability of transition from (x, y) to another cell within two steps. Analogously, we can deduce the n-step probability as

$$M^{1:n} = M + M^2 + \cdots + M^n \tag{2}$$

where $m\left(c_{(x,y)}, c_{(x',y')}\right)$ is the entry of $M^{1:n}$ representing n-step cells transition probability.

For the region transition probability, which is similar with the cell transition probability, we regard a region as a cell, and then calculate n-step region transition probability.

3.2. **Processing sampling points.** The main idea of sampling points calibration is that if one cell travels more related trajectories and the path traveling time from $c_{(x,y)}$ to $c_{(x',y')}$ is similar with the real time, it is more possible to be inserted into sampling points. In this paper, we use the average time to express the traveling time from $c_{(x,y)}$ to $c_{(x',y')}$ in historical trajectory dataset.

$$t\left(c_{(x,y)} \to c_{(x',y')}\right) = \frac{\sum_{P \in P_{c_{(x,y)} \to c_{(x',y')}}} c_{(x',y')}.t - c_{(x,y)}.t}{\left|P_{c_{(x,y)} \to c_{(x',y')}}\right|} \tag{3}$$

where $c_{(x,y)}.t$ is the time stamp on the path P which travels from $c_{(x,y)}$ to $c_{(x',y')}$, and $\left|P_{c_{(x,y)} \to c_{(x',y')}}\right|$ is the sum of paths traveling from $c_{(x,y)}$ to $c_{(x',y')}$. Then a matrix T indicates the average time cost.

The n-step path set from s_i to s_{i+1} is $\bar{P}(s_i \to s_{i+1}) = \bar{P}(s_i \to) \cap \bar{P}(\to s_{i+1})$ in which $\bar{P}(s_i \to)$ shows all n-step paths from s_i to other cells and analogously $\bar{P}(\to s_{i+1})$ shows all n-step paths from other cells to s_{i+1}.

The path probability $\mathrm{Pr}_i(P|S)$ of sampling points is computed with time similarity.

$$\mathrm{Pr}_i(P|S) = C \bullet \mathrm{Pr}_i^P \bullet S_i^t(P|S) \tag{4}$$

where C is a normalization parameter which satisfies $\sum_{P\in Ps(s_i,s_{i+1})} \Pr_i(P|S) = 1$, and $\Pr_i^P$ is the frequency of P to be the real path between s_i and s_{i+1}

$$\begin{aligned}\Pr_i^P &= \Pr_i(c_1^*, c_2^*, \ldots, c_k^*|s_1, s_2, s_3, \ldots, s_i, s_{i+1}, \ldots, s_{n-1}, s_n) \\ &= \Pr_i(c_1^*, c_2^*, \ldots, c_k^*|s_i, s_{i+1}) = \frac{m(s_i, c_1^*)m(c_1^*, c_2^*)\cdots m(c_k^*, s_{i+1})}{m(s_i, s_{i+1})}\end{aligned} \tag{5}$$

$S_i^t(P|S)$ describes the similarity between the real time cost and the time cost of P

$$\begin{aligned}S_i^t(P|S) &= \exp\left(-(\textit{real time cost} - \textit{time cost of } P)^2\right) \\ &= \exp\left(-((s_{i+1}.t - s_i.t) - [t(s_i \to c_1^*) + t(c_1^* \to c_2^*) + \cdots + t(c_k^* \to s_{i+1})])^2\right)\end{aligned} \tag{6}$$

Algorithm 3: Calibrating sampling points

Input : n-step transition probability matrix, transition probability, threshold η_{confi}, moving object sampling points $S(s_1, s_2, s_3, \ldots, s_i, s_{i+1}, \ldots, s_{n-1}, s_n)$ with the time stamp, average time matrix T.

Output: calibrated sampling points

for *each* $s_i \in S$ **do**
  Generate path set from s_i to s_{i+1};
  $Ps\,(s_i, s_{i+1}) \in$ all the paths in the n-step path set
  for *each* $P \in Ps(s_i, s_{i+1})$ **do**
    $\Pr_i(P|S)$;
  Initialize a list ℓ to record the insert cells
  for *each* $c^* \in Ps(s_i, s_{i+1})$ **do**
    $\Pr\,(c^*|S) \leftarrow 0$;
    for *each* $P \in Ps(s_i, s_{i+1})$ **do**
      if $c^* \in P$ **then**
      $\Pr(c^*|S) += \Pr_i(P|S)$;
    if $\Pr(c^*|S) > \eta_{confi}$ **then**
    Add c^* to ℓ;
  Insert the cells in ℓ into the sampling pair (s_i, s_{i+1});
return *the calibrated trajectory* $\bar{S}$;

4. **Indoor Moving Object Destination Prediction.** To fully use calibrated sampling points, we predict indoor moving object destination with bidirectional recurrent neural networks (BRNN).

4.1. **Bidirectional recurrent neural networks.** Recurrent neural networks (RNN) provides a very elegant way of dealing with (time) sequential data that embodies correlations between data points that are close in the sequence. However, with the increasing of information, the modeling power of the RNN is increasingly concentrated on remembering the input information, leaving less modeling power for combining the prediction knowledge from different input vectors. To overcome the limitation of a regular RNN, bidirectional recurrent neural networks is proposed. The idea of BRNN is to split the state neurons of a regular RNN in a part that is responsible for the positive time direction (forward states) and a part for the negative time direction (backward states). Outputs from forward states are not connected to inputs of backward states, and vice versa [11].

4.2. **Predicting moving object destination.** Compared with recurrent neural networks and multi-layer perception, BRNN makes full use of the beginning and ending information, which is important for destination prediction.

According to the characteristic of indoor destination prediction, we propose an improved BRNN structure by deleting its right and left hidden layer and employ the resilient back-propagation algorithm to train the network. Then the calibrated sampling points which include location and time information are used as input of improved BRNN. Before predicting destination, a destination set $(c_i)_{1\leq i\leq C}$ needs to be calculated with a mean-shift clustering algorithm on the destinations of all the historical trajectories. Then we use a hidden layer that associates a scalar value $(p_i)_i$ that is similar to a probability to each of these clusters and the sum of $(p_i)_i$ must be 1, and we compute them using a softmax layer:

$$p_i = \frac{\exp(e_i)}{\sum_{j=1}^{C}\exp(e_j)} \tag{7}$$

where $(e_j)_j$ are the activations of the previous layer.

At last, for our destination prediction, we calculate a weighted average of the destination cluster centers:

$$\hat{y} = \sum_{i=1}^{C} p_i c_i \tag{8}$$

5. **Experimental Results.** In this section, some relevant experiments are performed to evaluate the indoor destination prediction ability of our method with respect to the other approaches [9,10]. We use a real-world large scale pedestrian indoor trajectory dataset in a market from January 1st, 2014 to May 30th, 2014. It contains 50000 trajectories in cellular indoor smart environment such as the environment in Figure 1. We randomly select 1,000 trajectories forming a dataset to be the query trajectories and the remaining trajectories are used as training data. All following experiments are implemented in python and run on a computer with Intel Core 4 CPU (3.20 GHz) and 16 GB memory.

In our experiments, in order to qualitatively evaluate the performance of the proposed method, prediction error is proposed to quantify the accuracy of the predicted results. We use several commonly-used evaluation approaches in destination prediction, including cell precision and path completed percentage.

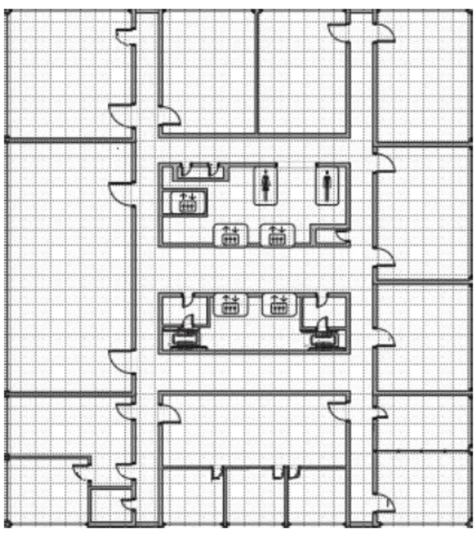

Figure 1. Cellular indoor smart environment

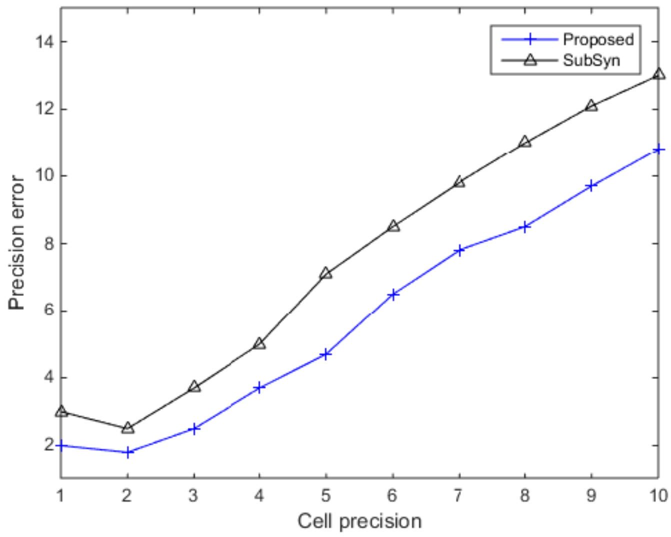

FIGURE 2. Precision error w.r.t cell precision comparing the proposed and SubSyn

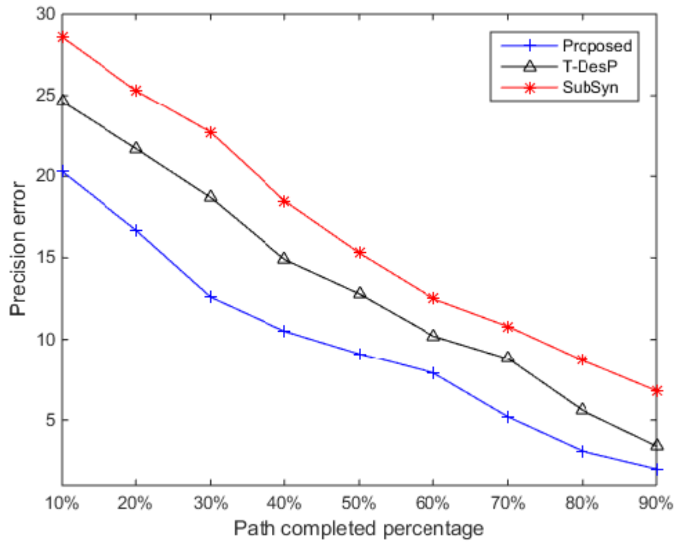

FIGURE 3. Precision error w.r.t path completed percentage comparing proposed, T-DesP and SubSyn

Varying the cell precision: It can be seen from Figure 2 that the proposed method has achieved better performance compared with SubSyn [9]. This is probably because our method has taken both indoor space connectivity and calibrating sampling points into considerations while SubSyn considers neither.

Varying the path completed percentage: Figure 3 shows that, as the path completed percentage gets higher, the prediction error is smaller. And compared with SubSyn and T-DesP [10], our method can achieve significant better results than others. Our method successfully coped with information loss based on the calibrated algorithm and BRNN makes fully use of the information contained in the sampling points.

Transition probability and completed percentage are used to predict destination in T-DesP, both of which have no relation with cell precision. So we do not compare with T-DesP in cell precision experiment.

6. **Conclusions and Future Work.** In this paper, we propose a new method for indoor destination prediction based on bidirectional recurrent neural networks. This method fully uses the feature of indoor space to predict destination more accurately. First, we propose a method to preprocess the indoor space to deal with the problem of connectivity. Second the sampling points are calibrated to address the problem of data sparsity by using historical trajectory dataset. At last, we predict indoor moving object destination with bidirectional recurrent neural networks by using calibrated sampling points. Our experimental results show that this method has achieved better performance in indoor destination prediction. In the future, we plan to combine the habit of users, then build a unified system for accurate indoor destination prediction.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (61472161, 61133011, 61402195, 61502198, 61303132, 61202308), Science & Technology Development Project of Jilin Province (20140101201JC).

REFERENCES

[1] S. Scellato, M. Musolesi, C. Mascolo, V. Latora and A. T. Campbell, NextPlace: A spatio-temporal prediction framework for pervasive systems, *Pervasive Computing*, vol.6696, pp.152-169, 2011.
[2] T. B. Wang, D. Q. Zhang, X. S. Zhou, X. Qi, H. B. Ni, H. P. Wang et al., Mining personal frequent routes via road corner detection, *IEEE Trans. Systems Man Cybernetics: Systems*, vol.46, pp.445-458, 2016.
[3] Q. Lin, D. Q. Zhang, K. Connelly, H. B. Ni, Z. W. Yu and X. S. Zhou, Disorientation detection by mining GPS trajectories for cognitively-impaired elders, *Pervasive and Mobile Computing*, vol.19, pp.71-85, 2015.
[4] S. Alamri, D. Taniar, M. Safar and H. Al-Khalidi, Spatiotemporal indexing for moving objects in an indoor cellular space, *Neurocomputing*, vol.122, pp.70-78, 2013.
[5] J. D. Domingo, C. Cerrada, E. Valero and J. A. Cerrada, Indoor positioning system using depth maps and wireless networks, *Journal of Sensors*, 2016.
[6] R. Ma, Q. Guo, C. Z. Hu and J. F. Xue, An improved WiFi indoor positioning algorithm by weighted fusion, *Sensors*, vol.15, pp.21824-21843, 2015.
[7] C. C. Yang and H. R. Shao, WiFi-based indoor positioning, *IEEE Communications Magazine*, vol.53, pp.150-157, 2015.
[8] K. Lin, M. Chen, J. Deng, M. M. Hassan and G. Fortino, Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings, *IEEE Trans. Automation Science and Engineering*, vol.13, pp.1294-1307, 2016.
[9] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang and Z. H. Xu, Destination prediction by sub-trajectory synthesis and privacy protection against such prediction, *The 29th IEEE International Conference on Data Engineering*, pp.254-265, 2013.
[10] X. Li, M. T. Li, Y. J. Gong, X. L. Zhang and J. Yin, T-DesP: Destination prediction based on big trajectory data, *IEEE Trans. Intelligent Transportation Systems*, vol.17, pp.2344-2354, 2016.
[11] M. Schuster and K. K. Paliwal, Bidirectional recurrent neural networks, *IEEE Trans. Signal Processing*, vol.45, pp.2673-2681, 1997.