International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.095 ISSN: 2581-4621

A Learning-Based POI Recommendation With Spatiotemporal Context Awareness

K Divya Reddy[1], J Jagadeesh Reddy[2], S Ashish[3], M Charan Babu[4], B Gopichand[5]
1,2,3,4,5Department of CSE, Annamacharya Institute of Technology & Sciences, Tirupati-517520

Abstract
Due to the great advances in mobility techniques, an increasing number of point-of-interest (POI)-related services have emerged, which could help users to navigate or predict POIs that may be interesting. Obviously, predicting POIs is a challenging task, mainly because of the complicated sequential transition regularities, and the heterogeneity and sparsity of the collected trajectory data. Most prior studies on successive POI recommendation mainly focused on modelling the correlation among POIs based on users' check-in data. However, given a user's check-in sequence, generally, the relationship between two consecutive POIs is usually both time and distance subtle. In this article, we propose a novel POI recommendation system to capture and learn the complicated sequential transitions by incorporating time and distance irregularity. In addition, we propose a feasible way to dynamically weight the decay values into the model learning process. The experimental results show that the proposed methods significantly outperform the state-of-the-art models in all metrics
Keywords: —Human mobility, machine learning, point-of interest (POI) recommendation, recurrent neural network (RNN), spatiotemporal data.

Introduction

THESE DAYS, due to the popularity of location-based social networks (LBSNs), people can easily share their thoughts, comments, pictures, etc., along with the location with their friends. This rapid rise in online check-in data provides a great opportunity to understand the mobility behaviours of people from their historical traces, and to foresee their future footprints. Point-of-interest (POI) prediction is one essential technique for trajectory analytics which has Fig.1. Example of recommendation considered distance between two consecutive locations.
widespread applications, such as smart transportation, urban planning, and tourism recommendation, to name a few. In real applications, the spatial and temporal differences are important; in a check-in trajectory, different intervals between two locations may reveal different information. By including this essential factor, we could make more precise predictions. For example, in Fig. 1, suppose there are two hot spots X and Y, and the check-in sequences of users Ua and Ub are Sa: L1 → L2 → L5 and Sb : L3 → L4 → L5, respectively. When predicting the next interesting spot, a system may prefer recommending X for Ua since the distance difference between the consecutive locations in Sa is long. This might indicate that Ua drives a car during his or her itinerary. In contrast, due to the distance between consecutive locations being close in Sb, Ub may visit every tourist spot by walking. Hence, Y is more suitable for recommendation to Ub.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.095 ISSN: 2581-4621

Related Work

1. Material and Methods:

POI recommendation, as a natural extension of conventional recommendation, has recently been proposed and has captivated great research interest. In this section, we provide an overview of our related works as two types: 1) traditional POI recommendation and 2) deep learning POI recommendation.

A. Traditional POI Recommendation The hidden Markov model (HMM) [6], [16] is a method to model future movements by constructing a transition matrix probability between locations based on past trajectories. The Markov chain is often exploited for POI recommendations in LBSNs to model the sequence pattern. Cheng et al. [9] considered two main properties in the check-in sequence: 1) a personalized Markov chain and 2) region localization.

B. Deep Learning POI Recommendation With the impressive achievement of deep neural network (DNN) models in many domains [11], [12], [22], [51], various approaches [5], [27] that leverage DNN for recommendation systems have been proposed including the POI prediction task. In order to foresee the future footprint, the latest visited POI needs to be considered.

2.PRELIMINARY

Let U = {u1, u2,..., un} be a set of users and L = {X1, X2,..., Xm} be a set of locations (or POIs). A check-in record is a pair (X, t) where X ∈ L and t is the timestamp of the check-in. Note that each POI is associated with its coordinate {latitude, longitude}. For a user u ∈ U, the trajectory check-in sequence Su = (X1, t1), (X2, t2), . . . represents the check-in history of u.

3.PROPOSED RECOMMENDATION SYSTEM: DENAVI Fig. 2 presents the architecture of DeNavi which consists of three major components: 1) feature extracting and embedding; 2) learning model and training; and 3) prediction module.

A. Feature Extraction and Embedding For pre-processing, the trajectory is transformed into a unique latent vector which has a lower dimension and can better capture the precise semantic spatiotemporal relationship. Due to the problem of sparsity in POI data, many research works now prefer to use embedding techniques instead of one-hot-encoding to model consecutive changes among POIs.

B. Learning Model and Training Latent vectors of the trajectory are processed and trained by the recurrent model for the learning module in DeNavi. In addition, differences in time and distance between successive trajectory check-ins are calculated and used as the spatiotemporal contexts in our model. The elapsed distance is calculated by Vincent’s formula to calculate the distance between two points on the surface of a spheroid, developed in [19].

DeNavi-LSTM: We utilize the spatiotemporal information, including the time and distance intervals to model the user’s short-term interest and long-term interest simultaneously. The idea of DeNavi-LSTM is extended from the LSTM model, which consists of one cell state and three controlled gates to keep and update the cell memory.

Algorithm 1 Model Training
Input: S = {Sull, Su2,..., Sun}: A set of users’ check-in POIsequences;
Z: batch, size;
Max_E: Maximum number of epochs. Output: θ2: The well-trained learning model
01: **For each** Sui ∈ S **do**
02: Sort each POI in Sui in chronological time order;
03**:** **For each** Xj ∈ Sui **do**
04: Calculate the time difference tj between
Xj and Xj+1;
05: Calculate the distance difference dj between Xj and Xj+i;

DOI:10.46647/ijetms.2023.v07si01.095 ISSN: 2581-4621

Info	The last four POIs of S_u ($X_{i-3} \to X_{i-2} \to X_{i-1} \to X_i$)	GroundTruth (X_{i+1})
POI	…→ L_A (22 New Road) → L_B (15 Khartoum Road) → L_C (20 Station) → L_D (347 High Street)	Sun Pier
Δt	…→ L_A (40sec) → L_B (18sec) → L_C (15sec) → L_D (10sec)	1min
Δd	…→ L_A (1.2km) → L_B (2.6km) → L_C (4.7km) → L_D (2.2km)	900m

Embed Xj to latent vector Xv j from θ1 learned by Eq. (1);
End
Transform Sui to Sv ui=(Xv 1, t1, d1), (Xv 2, t2, d2);
Insert Sv ui into training set T;
End
Calculate batch index m = |T|/Z;
Randomly select training instances from T to construct BS = {Batch1, Batch2,, Batchm};
Initialize model θ2 with random parameter setting;
While (epoch ≤ Max_E)
For each Botchy ∈ BS do
Update θ2 based on minimizing the prediction error derived from Eq. (20);
End
End
Output θ2;

DeNavi-GRU: DeNavi-GRU is a lightweight version of DeNavi-LSTM, which also utilizes gating information in different ways to overcome the vanishing gradient problem. The main difference between DeNavi-GRU and DeNavi-LSTM is that the unit controls the flow of information and exposes the full hidden content without using the memory unit. Although there is no memory cell in the GRU model, we also implement subspace decomposition of history from previous results.

DeNavi-Alpha: Since each LBSN dataset captures different users' mobility preferences, using a decay function to weight the impact of contexts for prediction is effective. However, when facing multiple contexts as the input, how to tune the significance of each context is a critical issue. The motivation of DeNavi assumes that the longer the elapsed time between consecutive check-ins, the less impact on the next prediction, and vice versa. However, in some applications, this assumption is not always held for both time and distance contexts for POI recommendation.

Prediction Module of DeNavi The prediction model is the final component that combines the context from different modules to complete the prediction task. As shown in Fig. 2, when recommending POIs to user u, we first input u's trajectory (i.e., the POI history) Su = (X1, t1), (X2, t2), . . . , (Xi, ti)into the feature extraction and embedding model to transform each POI into the corresponding latent vector. Then, the DeNavi system will feed the latent vector sequence, time differences, and distance differences into the well-trained learning model (DeNavi-LSTM, DeNavi-GRU, or DeNavi-Alpha) to derive a prediction vector Xv i+1. Finally, we compare all POI latent vectors to recommend the top-N similar locations to user u.

4. Result:

Based on the assumption that the greater the elapsed time or distance, the smaller the effect of the recent POIs on the current decision, intuitively, POIs visited a long time ago and a long distance away have little influence on the next POI. We can observe that in Fig. 13(b), the model gives more weight to d during the check-in at LA (801 Kaheka St.), LB (1223 N School St.), and LC (Art Bldg.), while t puts more emphasis on LD (Music Bldg Complex). The reason is that the check-in of LA, LB, and LC happened on the same day while LD happened the following day. This indicates that during the same day, DeNavi-Alpha learned to capture the sequential pattern and the distribution of d. However, when there is a large gap of t, DeNavi-Alpha will pay more attention to t. As a result, the ground-truth POI (500 Pohukaina) is recommended correctly in the inference process.

Model Type	**Accuracy**
Extra Tree Classifier	48.57
SVM	45.71
Logistic Regression	46.53
Recurrent Neural Network	48.97

5. Conclusion:

In this article, we investigated the challenges of POI mobility prediction from sparse and lengthy POI trajectories. We proposed a novel POI recommendation system DeNavi to predict the next move. Including the time and distance intervals between POI check-ins in the memory unit, three learning models: 1) DeNavi-LSTM; 2) DeNavi-GRU; and 3) DeNavi-Alpha were developed to enhance the performance of the standard recurrent networks. Specifically, by integrating the EWMA into the model learning process, DeNaviAlpha enables a practical approach to dynamically weighting the spatial and temporal decay values. Therefore, DeNavi-Alpha can capture how much each context should be emphasized in the prediction process. The detailed experiments on two real-life mobility datasets demonstrate that DeNavi significantly outperforms all the baselines in all metrics. In particular, the experimental results also show that DeNavi-Alpha performed better than the state-of-the-art methods since it can effectively capture meaningful contexts for mobility by dynamically integrating the weight of spatial and temporal decay values. For our future works, we will consider introducing the attention mechanism,user profile, and supplementary contexts into our model for better recommendation accuracy.

6. References:

1. C. C. Aggarwal, “Context-sensitive recommender systems,” in Recommender Systems. Cham, Switzerland: Springer, 2016, pp. 255–281.
2. A. Al-Molegi, M. Jabreel, and B. Ghaleb, “STF-RNN: Space time features-based recurrent neural network for predicting people next location,” in Proc. IEEE Symp. Series Comput. Intell. (SSCI), Athens, Greece, 2016, pp. 1–7.
3. M. Abdi, G. Okeyo, and R. Mwangi, “Matrix factorization techniques for context-aware collaborative filtering recommender systems: A survey,” Comput. Inf. Sci., vol. 11, no. 2, p. 1, 2018.
4. I. Baytas, C. Xiao, X. Zhang, F. Wang, A. Jain, and J. Zhou, “Patient subtyping via time-aware LSTM networks,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD), 2017, pp. 65–74.
5. H. Cheng et al., “Wide & deep learning for recommender systems,” in Proc. 1st Workshop Deep Learn. Recommend. Syst. (DLRS), 2016, pp. 7–10.
6. M. Chen, Y. Liu, and X. Yu, “NLPMM: A next location predictor with Markov modeling,” in Proc. Adv. Knowl. Disc. Data Min., 2014, pp. 186–197.
7. E. Cho, S. Myers, and J. Leskovec, “Friendship and mobility: User movement in location-based social networks,” in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD), 2011, pp. 1082–1090.

DOI:10.46647/ijetms.2023.v07si01.095 ISSN: 2581-4621

8. C. Cheng, H. Yang, I. King, and M. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in Proc. 26th AAAI Conf. Artif. Intell. (AAAI), 2012, pp. 17–23.
9. C. Cheng, H. Yang, M. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI), 2013, pp. 2605–2611.
10. H. Dyvik, "Exploiting structural similarities in machine translation," Comput. Human., vol. 28, nos. 4–5, pp. 225–234, 1994.
11. D. Yao, C. Zhang, J. Huang, and J. Bi, "SERM: A recurrent model for next location prediction in semantic trajectories," in Proc. ACM Conf. Inf. Knowl. Manag. (CIKM), 2017, pp. 2411–2414.
12. N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD), 2016, pp. 1555–1564.
13. J. Feng et al., "DeepMove: Predicting human mobility with attentional recurrent networks," in Proc. World Wide Web Conf. (WWW), 2018, pp. 1459–1468.
14. Y. Fu, B. Liu, Y. Ge, Z. Yao, and H. Xiong, "User preference learning with multiple information fusion for restaurant recommendation," in Proc. SIAM Int. Conf. Data Min., 2014, pp. 470–478.
15. [15] A. Graves, "Supervised sequence labelling," in Studies in Computational Intelligence Supervised Sequence Labelling With Recurrent Neural Networks. Heidelberg, Germany: Springer, 2012, pp. 5–13.