SPENT^{+}: A Category- and Region-aware Successive POI Recommendation Model

Hsu-Chao Lai†, Yi-Shu Lu†, Mu-Fan Wang†, Yi-Cheng Chen*, Wen-Yueh Shih†, Jiun-Long Huang‡,
†Inst. of Computer Science and Engineering, National Yang Ming Chiao Tung University, Taiwan
‡Inst. of Data Science and Engineering, National Yang Ming Chiao Tung University, Taiwan
*Department of Information Management, National Central University, Taiwan
Email: hsuchao.cs05g@nctu.edu.tw, barryrodkimo@gmail.com, odisme0804@gmail.com,
ycchen@mgt.ncu.edu.tw, wyshih@cs.nctu.edu.tw, jlhuang@cs.nctu.edu.tw

***Abstract*—To facilitate successive Point-of-Interests (POI) recommendation, the categories of POIs and the regions where POIs are located are seldom considered in existing models. In view of this, we extend a state-of-the-art model SPENT, named SPENT^{+}, by taking the category and the region into considerations. In SPENT^{+}, we formulate category- and region-aware check-in sequences, design the similarity trees to aggregate similar features, and finally establish the category latent vectors and region latent vectors, respectively. The above two latent vectors are aggregated as the category-region-aware latent vectors. Therefore, the category-region-latent vectors are sent to an LSTM together with conventional check-in sequences to improve successive POI recommendation. We conduct two real datasets, Gowalla and Foursquare, and compare with state-of-the-art methods in experiments. Results show that SPENT^{+} outperforms the baselines in terms of precision and recall.**

***Index Terms*—Successive POI recommendation, embedding, recommendation**

I. Introduction

Successive POI recommendation task [1] is to recommend sequential POIs to users to visit. Motivated by recent works showing that the Word2Vec technique [2], [3] has great capability to extract check-in sequences and then generate the latent vectors of POIs, we adopt the Word2Vec framework as our model foundation to capture representations of POIs. In addition, the neural network (NN) has been widely used in many fields due to its outstanding performance compared with other conventional methods. Since the recurrent neural network (RNN) is able to model complex sequential behavior, we can model users' successive transition behavior well through this powerful framework. Note that although some prior studies have adopted the RNN to model users' successive transition behavior [2], [3], they did not comprehensively exploit important features on LBSNs.

Specifically, *POI category* potentially affects users' willingness to visit a POI [4], [5]. On the other hand, the *region* where a POI is located is also an important factor influencing the users' visiting willingness [2]. Consider two POIs within the same category. Suppose that one is located in a famous region, and the other is not. Since the famous region can attract more visitors, the POI in the famous region has a higher probability of having more check-ins than the other. Thus, we argue that the influence of category and region should be taken into consideration in successive POI recommendation.

In this paper, based on [3], we propose a new model *SPENT^{+}* to take advantage of category and regional influence in successive POI recommendation. Specifically, we derive the the category check-in sequences from traditional historical check-in records. The latent vectors of categories are further built on the basis of the category check-in sequences. Similarly, we build the region checc-in sequences and the latent vectors of regions based on the region check-in sequences. SPENT^{+} establishes a *category (region) similarity tree* based on the category (region) check-in sequences to aggregate similar categories (regions). Therefore, compared to one-hot encoding technique, the dimension of features can be effectively reduced. Different from [3], SPENT^{+}*inserts the category latent vectors and region latent vectors into the POI similarity tree*, and uses the resultant POI similarity tree to *establish the category-region-aware POI latent vectors*. These vectors can be further used to recommend successive POIs with LSTM-based model [3]. The contributions are are two-fold: (1) we design a novel method to generate region latent vectors and category latent vectors based on the similarity tree, and (2) we aggregate region and category latent vectors as category-region-aware latent vectors to support sophisticated succesive POI recommendation. To evaluate SPENT^{+}, we conduct two real datasets to measure the performance and the results show that SPENT^{+} outperforms baseline methods.

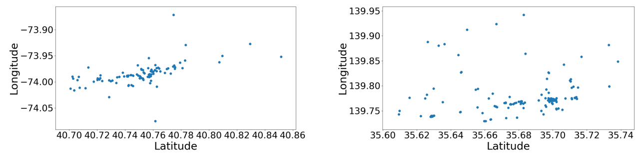
II. Related Work

Prior studies adopted additional POI information (e.g., text descriptions) to improve the performance of successive POI recommendation. CAPE [6] analyzes the influence of text descriptions of POIs. TMCA [7] introduces an LSTM-based encoder-decoder framework and multi-level attentions to capture the spatial-temporal representations for check-in activities. UGSE-LR and PEU-RNN [2] combined user preference, regional influence and successive transition influence to perform successive POI recommendation to build POIs' latent vectors to support next POI prediction. Word embedding [8] captures the contextual relationship between words, so people have been applying this method for successive POI recommendation

©Copyright IEICE - APNOMS 2021

TABLE I
STATISTICS OF THE USED DATASETS

Dataset	Gowalla	Foursquare
#Users	29,875	40,039
#POIs	8,460	11,828
#Categories	211	297
#Check-ins	1,416,703	1,633,647



(a) Locations of the Check-ins in New York City in Gowalla

(b) Locations of the Check-inss in Tokyo City in Foursquare

Fig. 1. Observation of Regional Influence

[9], [10]. Nickel et al. [11] learned POI representations by simultaneously capturing hierarchy and similarity. Lu et al. [2] integrated the latent vectors obtained from Word2Vec into the LSTM to perform successive POI recommendation. However, they do not aggregate similar regions (or categories), and hence may cause high dimensionality and underfitting.

III. Preliminaries

A. Dataset Statistics

Two real check-in datasets, Gowalla [12] and Foursquare [13], are used in this paper. The inactive users whose check-ins are less than 100 and the POIs checked in by less than 100 users are pruned. Table I shows the statistics of the datasets after pruning. Figures 1(a) and 1(b) show the locations of the check-in records in New York City in Gowalla and in Tokyo City in Foursquare. The distributions manifest that the check-in records locate in several hot regions. It motivates us to consider regional effect in this paper.

B. Problem Definition

Denote the set of users as $U = \{u_1, u_2, \ldots\}$, the set of categories of POI as $C = \{c_1, c_2, \ldots\}$ and the set of POIs as $L\{l_1, l_2, \ldots\}$. Each area is divided into several disjoint grid cells, named regions, by following [2], and these regions are denoted as $R = \{r_1, r_2, \ldots\}$. Each POI l_i is associated with a category, a region, and a geographical coordinate in the form of $l_i = (c_m, r_k, l_i^{lat}, l_i^{lng})$. (u, l, t) denotes a check-in that user u had checked in at POI l at time t.

Definition 1 (POI Check-in Sequence): A check-in sequence of user u, denoted as $S_u = \langle (l_1, t_1), (l_2, t_2), \cdots, (l_n, t_n) \rangle$, is a set of check-ins of user u. For each two successive check-ins, say (l_j, t_j) and (l_{j+1}, t_{j+1}), in a check-in sequence S_u, $t_n - t_1$ must be less than or equal to a pre-defined time threshold τ.

Definition 2 (Category and Region Check-in Sequence): Similar to Definition 1, we can replace each POI, say l_i, in a check-in sequence with the corresponding category c_i and the corresponding region r_i, respectively, to obtain the corresponding category check-in sequence and the corresponding region check-in sequence, respectively. For a check-in sequence $S_u = \langle (l_1, t_1), (l_2, t_2), \cdots, (l_n, t_n) \rangle$, the corresponding category check-in sequence and the corresponding region check-in sequence are denoted as $C_u = \langle (c_1, t_1), (c_2, t_2), \cdots, (c_n, t_n) \rangle$ and $R_u = \langle (r_1, t_1), (r_2, t_2), \cdots, (r_n, t_n) \rangle$, respectively, where the category of l_i is c_i and l_i is located in region r_i.
The successive POI recommendation is defined as follows:

Definition 3 (Successive POI Recommendation): Given a user $u \in U$, her current POI $l_i \in L$, the category c_i of l_i, the region r_i where l_i is located and check-in time t, the successive POI recommendation is to recommend N POIs where user u has never visited and is willing to visit within τ hours.

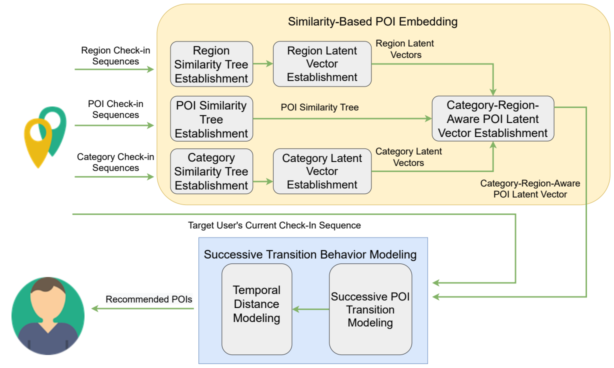


Fig. 2. Overview of the Proposed SPENT$^+$ Model

IV. SPENT$^+$

Figure 2 shows the two-phased architecture of SPENT$^+$. SPENT$^+$. In the first phase, SPENT$^+$ extracts region and category features with similarity trees and integrate those features into the POI latent vector in [3] as the *Category-Region-Aware POI Latent Vector* with Word2Vec. Note that we use CBoW instead of Skip-gram based model here after comparing their performance. The second phase exploits LSTM to model successive check-in behavior and temporal distance based on the category-region-aware POI latent vectors. Finally, SPENT$^+$ recommends top N POIs to users.

In this paper, we focus on extracting and integrating category and region features to enhance the recommendation performance, which lies in the first phase of SPENT$^+$, and please refer to [3] for the details of the second phase.

A. Similarity Tree Establishment

To build the similarity tree, we first build the User-Region Matrix. Figure 3(a) shows an example User-Region matrix for the case that $U = \{u_1, u_2, u_3, u_4\}$ and $R = \{r_1, r_2, r_3, r_4\}$. The value of each entry (u_i, r_j) is set to 1 when user u_i had checked in at r_j, and zero otherwise. Then the similarity matrix can be built based on the User-Region matrix. Figure 3(b) shows an example with cosine similarity. The value of the entry (r_i, r_j) in the similarity matrix is the similarity of l_i and l_j. Similarity metrics can be changed based on scenarios.

	r_1	r_2	r_3	r_4
u_1	1	0	1	0
u_2	0	1	0	0
u_3	1	0	1	1
u_4	0	0	1	0

(a) User-Region Matrix

	r_1	r_2	r_3	r_4
r_1	1.0	0.0	0.82	0.71
r_2	0.0	1.0	0.0	0.0
r_3	0.82	0.0	1.0	0.58
r_4	0.71	0.0	0.58	1.0

(b) Similarity Matrix

Fig. 3. An Example of Region Similarity Matrix

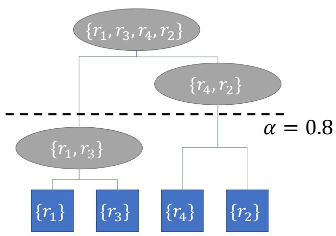


Fig. 4. A Region Similarity Tree with Merge Bound

Motivated by the agglomerative hierarchical clustering, similarity tree aggregates similar regions hierarchically. Initially, each region r_i is set to one group $\{r_i\}$. Two groups of the highest similarity are merged. Denote the similarity of regions r_i and r_j as $sim(r_i, r_j)$. Maximum, which is defined as follows, are used in this paper to measure the similarity of two groups of regions G_1 and G_2.

$$sim_{MAX}(G_1, G_2) = \max_{\forall l_i \in G_1 \ and \ \forall l_j \in G_2} \{sim(l_1, l_2)\}. \quad (1)$$

Figure 4 illustrates an example of the building procedure. We use Maximum to measure the similarity of two groups. Initially, we have four groups: $\{r_1\}$, $\{r_2\}$, $\{r_3\}$ and $\{r_4\}$. Since being of the highest similarity, $\{r_1\}$ and $\{r_3\}$ are merged into $\{r_1, r_3\}$. The merging steps can repeat until all groups are merged into one group. However, the tree will be too high when the data include enormous regions, and thus greatly increases the training time of Word2Vec. To avoid this situation, we set a predefined *merge bound* α. That is, we only merge region groups that has similarity greater than α. As shown in Figure 4, when α is set to 0.8, the resultant groups will be $\{r_1, r_3\}$, $\{r_2\}$, and $\{r_4\}$, since none of their mutual similarity exceeds 0.8. These three groups are then merged by their frequencies by Word2Vec.

B. Region Latent Vector Establishment

To establish the region latent vectors, for each $Sequence(l) = \langle l_1, l_2, \ldots \rangle$, we build the corresponding region sequence, termed as $Region\_Sequence(l)$ replacing each POI in $Sequence(l)$, say l_i, with the region, say r_i, where l_i is located. With the region similarity tree, SPENT$^+$ use it to learn the latent vector of each region r, termed as $RegionVec(r)$, based on the following objective function.

$$\hat{\theta^r} = \arg\max_{\theta^r} P(r|Region\_Sequence(l)). \quad (2)$$

Furthermore, the probability of $P(r|Region\_Sequence(l))$ can be formulated as follows:

$$P(r|Region\_Sequence(l)) = \prod_{j=1}^{R(w)-1} [[\sigma((X_v^r)^T \theta_{j-1}^r)]^{1-d_j} \cdot [\sigma((X_v^r)^T \theta_{j-1}^r)]^{d_j}], \quad (3)$$

where X_v^r is the average of the embedding vectors of the regions in $Region\_Sequence(l)$, and θ_{j-1}^r is the vector of the j-th node on the path from the root to r in similarity tree.

C. Category Latent Vector Establishment

The category latent vectors are built similarly to the region latent vectors. We first built the category similarity tree as Section IV-A. For each $Sequence(l) = \langle l_1, l_2, \ldots \rangle$, we build the corresponding category sequence, denoted as $Category\_Sequence(c)$ by replacing each POI in $Sequence(l)$, say l_i, with the category, say c_i, which l_i belongs to. We build the category similarity tree, and learn the latent vector of each category c, termed as $CategoryVec(c)$, based on the following objective function.

$$\hat{\theta^c} = \arg\max_{\theta^c} P(c|Category\_Sequence(l)). \quad (4)$$

We formulate $P(c|Category\_Sequence(c))$ as follows:

$$P(c|Category\_Sequence(l)) = \prod_{j=1}^{C(w)-1} [[\sigma((X_v^c)^T \theta_{j-1}^c)]^{1-d_j} \cdot [\sigma((X_v^c)^T \theta_{j-1}^c)]^{d_j}], \quad (5)$$

where X_v^c is the average of the embedding vectors of the categories in $Category\_Sequence(l)$, θ_{j-1}^c is the vector of the j-th node on the path from the root to c in similarity tree.

To integrate category influence and regional influence, for each POI l, we aggregate the corresponding category latent vector and the region latent vector into the initial *category-region-aware POI latent vector* of l, denoted as $CateRegVec(l)$, by the following equation.

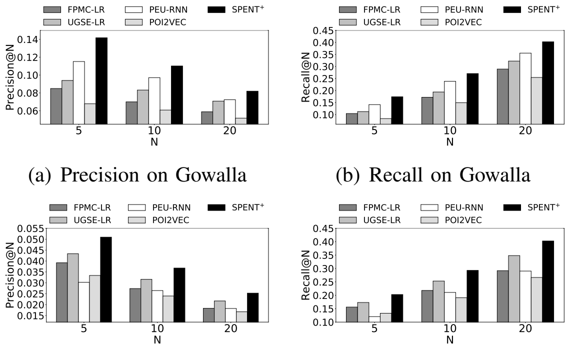
$$CateRegVec(l) = CategoryVec(c) + RegionVec(r), \quad (6)$$

where r is the region where l is located and c is the category which l belongs to. Afterwards, $CateRegVec(l)$ includes comprehensive features, and is further feed to the second phase for successive POI recommendation.

V. Experiments

A. Experimental Settings

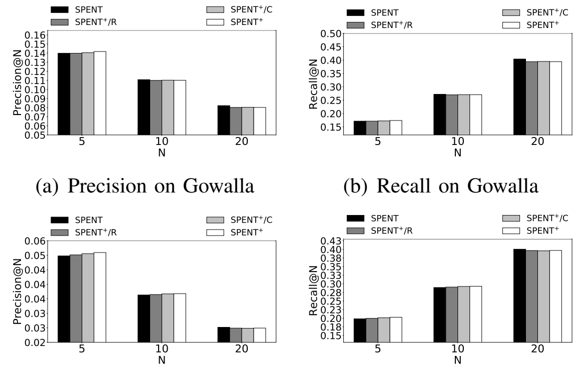
We conduct Gowalla and Foursquare to evaluate the proposed SPENT$^+$. The statistics and preprocess of the datasets are described in Section III-A. In each dataset, the first 70% check-ins as used as the training data, the next 10% as the tuning data and the remaining 20% as the testing data. Following [2], the grid size is set to 0.5 km. The default value



(a) Precision on Gowalla (b) Recall on Gowalla

(c) Precision on Foursquare (d) Recall on Foursquare

Fig. 5. Performance Comparison with Prior Methods



(a) Precision on Gowalla (b) Recall on Gowalla

(c) Precision on Foursquare (d) Recall on Foursquare

Fig. 6. Effect of Category and Region

of the time threshold τ is 6 hours [1] and the number of recommended POIs N is set to 10. Two widely-used metrics, $Precision@N$ and $Recall@N$ [3], are used to evaluate the performance of successive POI recommendation methods. We compare with the following state-of-the-art methods:

FPMC-LR [1]: A method that employs Markov chain and matrix factorization with a localized region constraint.

UGSE-LR [2]: A user-based CF method that considers grid influence, grid transition, and POI transition.

POI2VEC [9]: A variant of the Word2Vec technique that incorporates the geographical influence.

PEU-RNN [2]: An LSTM-based method using the latent vectors from Word2Vec that embeds both users and POIs.

SPENT [3]: A baseline without category and region features.

SPENT$^+$: The proposed method.

SPENT$^+$/R: SPENT$^+$ not using region latent vectors.

SPENT$^+$/C: SPENT$^+$ not using category latent vectors.

B. Experimental Results

Figure 5 shows the performance of all methods on both datasets and it manifests that SPENT$^+$ outperforms all other recommendation methods by at least 15.2% in terms of both precision and recall. RNN-based methods (i.e., PEU-RNN and SPENT$^+$) perform well on Gowalla, showing the strength of RNN on the dense Gowalla dataset. From Figures 5c and 5d, PEU-RNN does not perform well since Foursquare is sparse. However, by integrating the similarity tree, category influence, regional influence, and the temporal distance gate, SPENT$^+$ still outperforms baseline methods by at least 16%.

Figure 6 presents the ablation study regarding the influence of category and region. SPENT$^+$has the best performance in terms of precision and recall in most cases. Figures 6a and 6b show that the performance of SPENT$^+$and SPENT is quite close in Gowalla , while Figures 6c and 6d show that SPENT$^+$outperforms SPENT in Foursquare. We argue that the reasons are twofold. First, the number of categories is not sufficient in Gowalla. As shown in Table I, the number of categories in Foursquare is about 50% greater than that in Gowalla. Second, Foursquare is sparser than Gowalla so category and region are less significant in Gowalla. Moreover, Figures 6c and 6d show that SPENT$^+$/C outperforms SPENT$^+$/R in terms of precision and recall in Foursquare, showing that region is more important than category.

VI. Conclusion

In this paper, we proposed SPENT$^+$ which fully exploits POI category and region information with similarity trees to generate latent vectors for all POIs for successive POI recommendation problem. Experimental results manifest that SPENT$^+$ outperforms the other prior methods in terms of precision and recall in most cases on both datasets.

Acknowledgment

This work is supported in part by MOST under grants 109-2221-E-009-118-MY3, 110-2222-E-009-005-MY2, and 109-2218-E-009-015.

References

[1] C. Cheng, H. Yang, M. R. Lyu, and I. King, “Where you like to go next: successive point-of-interest recommendation,” in *IJCAI*, 2013.

[2] Y.-S. Lu, W.-Y. Shih, H.-Y. Gau, K.-C. Chung, and J.-L. Huang, “On successive point-of-interest recommendation,” *World Wide Web*, 2019.

[3] M. Wang, Y. Lu, and J. Huang, “SPENT: a successive poi recommendation method using similarity-based poi embedding and recurrent neural network with temporal influence,” in *BigComp*, 2019.

[4] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung, “Inferring a personalized next point-of-interest recommendation model with latent behavior patterns,” in *AAAI*, 2016.

[5] I. Lin, Y. Lu, W. Shih, and J. Huang, “Successive poi recommendation with category transition and temporal influence,” in *Proceedings of the 1st International Workshop on Smart Computing and Applications*, 2018.

[6] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, “Content-aware hierarchical point-of-interest embedding model for successive poi recommendation,” in *IJCAI*, 2018.

[7] R. Li, Y. Shen, and Y. Zhu, “Next point-of-interest recommendation with temporal and multi-level context attention,” in *ICDM*, 2018.

[8] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, 2013.

[9] S. Feng, G. Cong, B. An, and Y. M. Chee, “Poi2vec: geographical latent representation for predicting future visitors.” in *AAAI*, 2017.

[10] J. Yang and C. Eickhoff, “Unsupervised learning of parsimonious general-purpose embeddings for user and location modeling,” *ACM TIST*, 2018.

[11] M. Nickel and D. Kiela, “Poincaré embeddings for learning hierarchical representations,” in *NIPS*, 2017.

[12] Y. Liu, W. Wei, A. Sun, and C. Miao, “Exploiting geographical neighborhood characteristics for location recommendation,” in *CIKM*, 2014.

[13] D. Yang, D. Zhang, B. Qu, and P. Cudre-Mauroux, “Privcheck: privacy-preserving check-in data publishing for personalized location based services,” in *Proceedings of the 2016 UbiComp*, 2016.