Venue2Vec: An Efficient Embedding Model for Fine-grained User Location Prediction in Geo-Social Networks

Shuai Xu, Jiuxin Cao, Shancang Li, Phil Legg, Bo Liu

***Abstract*—Geo-Social networks (GSN) significantly improve location-aware capability of services by offering geo-located content based on the huge volumes of data generated in the GSN. The problem of user location prediction based on user generated data in GSN has been extensively studied. However, existing studies are either concerning predicting users' next check-in location or predicting their future check-in location at a given time with coarse granularity. An unified model that can predict both scenarios with fine granularity is quite rare. Also, due to the heterogeneity of multiple factors associated with both locations and users, how to efficiently incorporate these information still remains challenging.**

Inspired by the recent success of word embedding in natural language processing, in this work, we propose a novel embedding model called Venue2Vec which automatically incorporates temporal-spatial context, semantic information, and sequential relations for fine-grained user location prediction. Locations of the same type, and those that are geographically close or often visited successively by users will be situated closer within the embedding space. Based on our proposed Venue2Vec model, we design techniques that allow for predicting a user's next check-in location, and also their future check-in location at a given time. We conduct experiments on three real-world GSN datasets to verify the performance of the proposed model. Experimental results on both tasks show that Venue2Vec model outperforms several state-of-the-art models on various evaluation metrics. Furthermore, we show how the Venue2Vec model can be more time efficient due to being parallelizable.

***Keywords*-Location prediction, location embedding, random walk, language model, Geo-Social Networks.**

I. Introduction

With the popularity of intelligent mobile terminals and the progress of positioning technology, Geo-Social Networks (GSN), which can simultaneously provide location-based service and online social networking service, have become increasingly prevalent. Typical GSN platforms such as Foursquare enable users to check-in anytime and anywhere. Users are able to share check-in records with their friends, which bridges users' online behavior with offline mobility. As it is rich in temporal, spatial and semantic information, the huge volume of user check-in data generated in GSN makes it possible to explore intrinsic pattern of user mobility. Such patterns could predict where a user would visit in the future based on his/her historical check-in records. This application scenario has value for both individuals and the wider society. From the individual point of view, accurate location prediction can provide users with informative personalized product recommendation [1, 2]. From a societal view, such analysis can accurately predict where traffic jams would happen, thus can be helpful for urban intelligent transportation [3, 4].

The problem of user check-in location prediction has been well-studied in recent years. From the perspective of prediction timeliness, existing studies can be classified into two categories, i.e. (1) Next check-in location prediction, which focuses on predicting where a user will go next given his/her historical check-in data; and (2) Anytime check-in location prediction, which aims to predict where a user will be at a given time far in the future. In practical application scenarios, next check-in location prediction pays more attention to the real-time nature of prediction results. Based on users' recent historical trajectories, it focuses on predicting their next visit location in a timely manner before they make a request, thus facilitating advertisers to launch mobile advertisements to targeted users. In contrast, anytime check-in location prediction focuses on mining users' mobility pattern from their long-term trajectories. It is able to predict where a user will be at a certain time in the future based on his/her mobility pattern, thus providing an effective means for authorities to monitor users of interest.

Up to now, an unified model that can predict both scenarios with fine granularity is quite rare. Most of the existing related works such as [5–9] are concerning predicting users' next check-in location. For a few studies such as [10, 11] that deal with predicting users' check-in location far in the future, they often ignore the integration effect of various factors on user check-in behavior. Moreover, many existing studies mainly conduct location prediction with a coarse granularity concerning only location category or the single day as the time measurement. For example, [12] can only predict what kind of location (i.e. location category, not an exact location) a given user will visit based on his/her historical check-in data, while [5] only makes prediction where a user will go at a given day (e.g. next Friday), rather than at a specific time. The practicability of such coarse-grained location prediction is narrowed in the increasingly fierce competition environment for location-based services. For related works of both categories, they usually exploit various features such as temporal cyclic effect,

Shuai Xu and Bo Liu are with School of Computer Science and Engineering, Southeast University, Nanjing 211100, China. Email: {xushuai7, bliu}@seu.edu.cn.

Jiuxin Cao is the corresponding author. He is with School of Cyber Science and Engineering, Southeast University, Nanjing 211100, China. Email: jx.cao@seu.edu.cn.

Shancang Li and Phil Legg are with Department of Computer Science and Creative Technologies, University of the West of England, Bristol, UK. Email:{Shancang.Li, Phil.Legg}@uwe.ac.uk.

geographical influence, sequential relation and semantic information to improve location prediction performance. Current studies often incorporate different contextual factors through a simple fusion strategy, i.e. modeling these factors as weighting coefficients. Although neural network-based techniques like [7] provide a new way of modeling different factors in an unified manner, the task is still challenging that how to effectively incorporate these features in the neural network framework and meanwhile achieve an explainable model [13].

Unlike existing works, in this paper we aim at developing a unified model incorporating temporal-spatial context, semantic information, and sequential relation so as to realize both next check-in location prediction and anytime check-in location prediction. As we utilize the non-linear transformation to exploit different features in an integrated manner, the interpretability of the model can be naturally increased. Inspired by the recent success of word embedding in natural language processing and text mining, we propose a novel embedding model called Venue2Vec for fine-grained user location prediction in Geo-Social Networks. Locations of the same type and those that are geographically near or often visited successively by users will be closer in the embedding space. To fulfill this task, we firstly sample the random walk sequences over locations by using a mixture of location transition pattern (i.e. sequential relation) and geographical influence (i.e. spatial context). Then, a Skip-gram neural language model is applied to learn the embedding of each location, after which the corresponding semantic information is concatenated to obtain the final location vector. Thirdly, both the target user's preferences for his/her next visit and his/her preference at a given time (i.e. temporal context) can be calculated based on his/her past check-in records. Finally, the prediction result is produced based on the similarity between user preference and location vectors. Considering that users tend to visit geographically near places, we further incorporate a distance factor (i.e. spatial context) using kernel density estimation (KDE) in the final result. Figure 1 illustrates the graphical architecture of the proposed Venue2Vec model.

In this work, we train the proposed Venue2Vec model using three real-world GSN datasets. We visualize the obtained location embeddings and observe the clustering phenomenon, which indicates that the fusion of multiple factors improve the interpretability of the model. The main contributions of this paper are summarized as follows:

- A novel embedding model named Venue2Vec is proposed, which naturally encodes temporal-spatial context, semantic information, and sequential relation in an unified manner for fine-grained user location prediction. To our knowledge, this is the first work that develops a unified model for user next check-in location prediction and anytime check-in location prediction at the same time.
- Based on the random walk sampling process, we manage to transfer the popular word embedding technique to a new application for learning location embeddings.
- Extensive experiments have been conducted to evaluate the effectiveness of the Venue2Vec model. With regard to user next check-in location prediction task, the proposed approach achieves the best performance with $Acc@1$ being as much as 13.2%, 11.1% and 5.7% higher than state-of-the-art approach on three datasets, respectively. As for anytime check-in location prediction task, our approach still outperforms the competitors with as much as 16.9%, 41.8% and 10.6% higher advantage on $Acc@1$ metric over three datasets, respectively. These results demonstrate the superiority of the proposed Venue2Vec model over state-of-the-art models.
- We verify that Venue2Vec model is time efficient as it can be implemented in parallel. The training time of different models on three GSN datasets shows that Venue2Vec model can save up to 80% training time on average due to the multi-worker strategy, which demonstrates the scalability of the proposed model for large-scale GSNs.

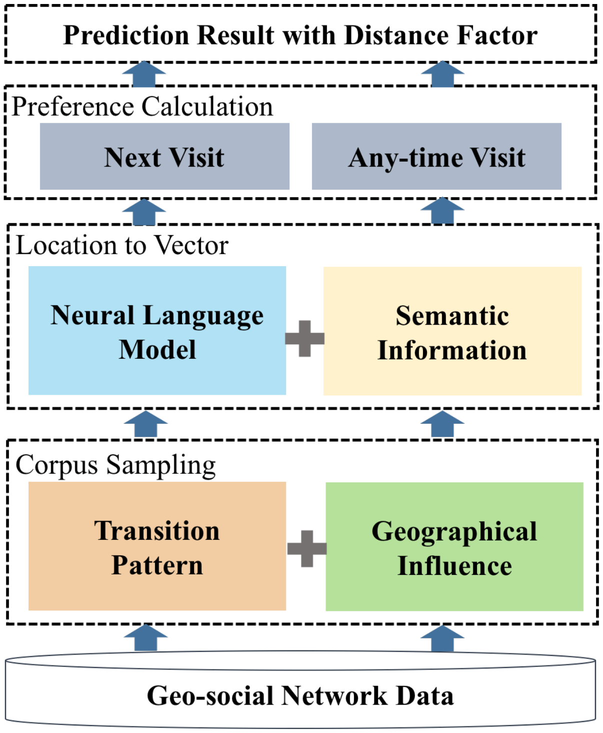


Fig. 1. Graphical architecture of Venue2Vec model.

The remainder of the paper is organized as follows. Section II reviews the related works. Section III formalizes the location prediction problem. In Section IV, the details of Venue2Vec model are illustrated. Section V elaborates the experiments. Finally, Section VI concludes the paper.

II. Related Work

It is believed that human mobility is predictable based on the historic records. In this section, we will review existing research works related to our work from two aspects, i.e. indicative features involved in user location prediction and mainstream location prediction solutions.

A. *Indicative Features for User Location Prediction*

Temporal Cyclic Effect. Adequate studies show that human activity has a strong time periodicity. Montjoye *et al.* [14]

confirmed that users' check-in pattern may be cycled around one day or one week. Gao *et al.* [15, 16] studied the temporal cyclic patterns of user check-ins in terms of temporal consecutiveness, and Noulas *et al.* [17] incorporated the time periodicity as the most fundamental feature for users' next check-in location prediction.

Geographical Influence. As addressed in [18] and [19], users' check-in behavior holds a clustering property in space, namely, there exists a check-in center for each user, and the closer a user is to the center, the higher probability he will check-in there. Research in [20] and [21] incorporated the geographical influence using kernel density estimation into a weighted matrix factorization framework to alleviate the data sparseness in location prediction.

Sequential Relation. The succession of human activities is not random, on the contrary, for example, we may visit the coffee shop after work or go to a hotel after landing at an airport. As a consequence, such sequential relation existing in human mobility is often exploited as a significant indicator for user location prediction [11, 17]. A typical assumption is that users' recent check-in locations usually have stronger influence than those locations visited long time ago [11].

Semantic Information. The textual description, which is one of the most representative meta-data of a location, has been widely utilized in user location prediction [22–24]. Zhang *et al.* [22] explored the predictive capability of location category (e.g. 'Food', 'Bar' and 'Shop') in personalized location recommendation. Wang *et al.* [23] studied the effectiveness of textual tags for location recommendation, while Yin *et al.* [24] used a semantic-coherent soft cluster of words associated with locations to predict a user's future check-in location.

More recently, a few researchers focused on trying to incorporate all the above features into one model [7, 25]. For example, Yin *et al.* [25] designed a probability generating model by jointly considering temporal cyclic effect, geographical influence and semantic information, and Liu *et al.* [7] combined these features into a neural network framework for location recommendation. However, these solutions usually fused different contextual factors using a simple strategy such as modeling them as weighting coefficients. In contrast, our work utilizes the non-linear transformation to exploit different features in an integrated manner, which can not only smoothly combine various factors in a shared latent space, but also work as a generic method.

B. Location Prediction Solutions

Existing solutions for location prediction include matrix factorization based methods [20–22], Markov Chain (MC) based methods [11], neighborhood based methods [26] and machine-learning based methods [27, 28]. Matrix factorization based methods have become the state-of-the-art approaches to collaborative filtering. The basic objective of matrix factorization in location prediction is to factorize a user-location preference matrix into two low rank matrices, each matrix represents the latent factors of users or locations. It has also been extended to be time-aware in recent years. However, it is hard for matrix factorization based models to generate latent representations of new users or new locations in the training data. Markov Chain based methods aim to predict the next check-in location of a user based on the past sequential check-in records. However, these methods generally assume that all the check-ins are linearly combined, which is indeed a strong independent assumption among factors [29]. Neighborhood based methods are widely used in collaborative filtering. They can be naturally effective for location prediction with both temporal and spatial contexts and usually have the advantage of being interpretable. However, neighborhood based methods are unable to model the underlying properties in users' sequential check-in behaviors. Machine-learning based methods such as the supervised scoring model and classification model proposed in [30] have been popular in user location prediction as they can mine indicative features automatically from users' check-in data. However, these methods often require that each user has sufficient check-in data for training, which is not realistic since users' check-in records are normally very sparse.

In recent years, embedding methods have proved to be effective in capturing linguistic regularities of how items (e.g. words in sentences) interact with each other [31]. Musto *et al.* [32] conduct an empirical study to evaluate the effectiveness of a simplified collaborative filtering recommendation framework that applies several word embedding techniques to model items and make recommendations to users. Experimental results in [33] also indicate the promise of word embedding for location recommendation. So far, only a few research efforts can be found that applies embedding methods for user location prediction. In contrast to previous works, this work proposes an embedding model called Venue2Vec that maps each location in GSN into a vector, which can be used to predict both users' next check-in location and users' future check-in location at anytime with the time granularity accurate to hour level. Based on the DeepWalk sampling technique [34], Venue2Vec incorporates multiple context features (i.e. temporal-spatial context, semantic information as well as sequential relation) in an unified framework. Since the training data used for the model is sampled from global check-in data, it does not depend on the amount of a specific user's check-in records. Last but not least, Venue2Vec model proves to be more time efficient than other competitive approaches as it can be naturally parallelizable.

III. Problem Definition

Without loss of generality, we firstly define the key data structures and notations used in this paper.

Definition 1: (**Location**). A location in GSN is defined as a uniquely identified specific site (e.g. a music bar or a coffee shop). A location contains three attributes: identifier, geographical coordinates and semantic information. For example, a movie theater in Tokyo (Japan) can be represented as $\{4b558a35f964a520eae627e3$, [35.673587359609904, 139.76270735263824], Art & $Entertainment\}$, where $4b558a35f964a520eae627e3$ is the identifier, [35.673587359609904, 139.76270735263824] is the geographical coordinates of the theater, and the attributes Art & $Entertainment$ denotes the category or type of the location.

Definition 2: (**Check-in**). In a GSN dataset, a check-in is defined as a triple (u, v, t), in which we say user u made a check-in at location v at time t.

Definition 3: (**User Check-in Records**). For each user u, his/her historical check-in records can be defined as a set of check-ins sorted using timestamp, i.e. $C_u = \{< u, v_1, t_1 >, < u, v_2, t_2 >, ..., < u, v_n, t_n >\}$.

The two categories of location prediction problems in this paper are defined as follows.

Problem Definition 1: **Next Check-in Location Prediction**. Given a user u_i with his/her historical check-in records up to time t_{n-1}, i.e. $C_{u_i} = \{< u, v_1, t_1 >, < u, v_2, t_2 >, ..., < u, v_{n-1}, t_{n-1} >\}$, the task is to predict top-N locations that u_i will visit next at time t_n, so that the exact location she will visit is ranked at the highest possible position.

Problem Definition 2: **Anytime Check-in Location Prediction**. Given a user u_i and his/her historical check-in records C_{u_i} as well as a future time t that accurate to hour, our goal is to predict top-N locations that u_i will visit at that time, so that the exact location she will visit is ranked at the highest possible position.

We note that the problem definitions given here are different from our previous work [30]. Here, we produce the prediction result using all possible candidate locations in the dataset, while our previous work only considers locations that a user has visited before, thus is limited and ineffective for predicting new places.

IV. Model Description

In this section, we firstly describe the steps taken to sample location sequences for model training. Then, we describe how to learn the vector-space representation of a location in GSN. Thirdly, two methods for user preference calculation are introduced. Finally, we explain how to produce the final prediction results.

A. Sequence Sampling

As mentioned above, we exploit work from natural language modelling to infer the embedding, i.e. the vector-space representation of a location in GSN. The motivation can be explained from two aspects. Firstly, if we consider the past check-in records of users as documents which consist of a list of locations, these locations could be naturally equivalent to words appearing in a language model corpus. With this, the sentences used by the language model can be mapped as the past preferences of users. Secondly, the purposes of the language model and the location prediction process are quite similar. Language model aims to predict the words based on the observed words, which can be considered as equivalent to predicting future locations based on those already visited.

To encode sequential relation and geographical influence of locations, we proposed a random walk method to learn the global transition probability between locations. Random walk with restart has been successfully used to measure the correlation between two nodes in a graph [35]. To achieve this goal, the foremost step is constructing the location sequences used in the Venue2Vec model. We adopt the DeepWalk sampling technique proposed in [34] to build the location network, where each node in the topology represents a location and the weight of each directed edge represents the probability of travelling from one node to another node.

As sequential relation and geographical influence are verified to be effective for user location prediction [6, 10, 28], we construct the random walk sequences over locations by using a mixture of sequential relation and geographical influence. The random walk probability, i.e. the weights of an edge from location v_i to location v_j over the network is calculated according to the following equations:

$$p(v_i|v_j) = \alpha \frac{\kappa(v_i, v_j)}{\sum\limits_m \kappa(v_i, v_m)} + (1 - \alpha) \frac{f(v_i, v_j)}{\sum\limits_m f(v_i, v_m)} \tag{1}$$

$$\kappa(v_i, v_j) = \frac{1}{(1 + e^{5 \cdot \frac{d(v_i, v_j) - \overline{d}}{\sigma(d)}})} \tag{2}$$

where $d(v_i, v_j)$ denotes the Euclidean distance between location v_i and v_j using their coordinates, $\overline{d}$ and $\sigma(d)$ are the mean and standard deviation of $d(v_i, v_j)$, respectively, and $f(v_i, v_j)$ is the transition frequency from location v_i to location v_j in the global training dataset. Indeed, $\kappa(v_i, v_j)$ depicts the geographical strength that one location v_i relates to another location v_j according to Waldo Tobler's First Law of Geography, i.e. "Everything is related to everything else, but near things tend to be more related than distant things." In equation (1), the first term in the right part captures the inherent geographical influence between locations, while the second term captures the sequential check-in behavior of users in GSN. α is used to then balance the two components.

In the sampling process, for each location node, we start the random walk and sample k sequences from this node [1]. To ensure that there is sufficient corpus for model training we use $k = 50$. The length of each sequence is randomly determined, where we bound this between 50 and 100 steps.

An obvious advantage of sampling location sequences in such a manner is that relying on information obtained from short random walks makes it possible to accommodate small changes in the location network without the need for global re-computation. After these steps, we obtain the location sequences (similar to words in the textual documents) that can be used for model training.

B. Embedding Learning

The key technique used to infer the vector-space representation of a location is similar to Word2Vec [36]. Specifically, we apply the Skip-gram language model with hierarchical softmax to learn location embeddings. The Skip-gram model uses the current location to predict the locations around the current one. A location's embedding is learnt to maximize the probability of seeing its neighbors in the sampled sequences, through which the semantic and syntactic information of location sequences could be introduced into the embedding.

[1] In practice, we use the location identifier to represent each location.

In the Skip-gram language model with hierarchical softmax, each input sample is in the form of $(v, context(v))$, where v is a location in the sequence and $context(v)$ are the neighboring locations of v. Let the embedding vector of v be x_v, the total path length from root node x_v to leaf node v is l_v, and $d_j^v \in \{0, 1\}$ is the corresponding Hoffman code for v in the path, then we can formalize the probability that predicts $context(v)$ using v as:

$$p(context(v)|v) = \prod_{u \in context(v)} p(u|v) \tag{3}$$

where the probability $p(u|v)$ can be formalized as:

$$p(u|v) = \prod_{j=2}^{l_v} \sigma(x_v^T \theta_j^v)^{1-d_j^v} \cdot [1 - \sigma(x_v^T \theta_j^v)]^{d_j^v} \tag{4}$$

Then we obtain the joint probability for all locations in the corpus using the logarithmic likelihood form:

$$\begin{aligned} L = & \sum_{v \in V} \log \prod_{u \in context(v)} \prod_{j=2}^{l_v} \{\sigma(x_v^T \theta_j^v)^{1-d_j^v} \cdot [1 - \sigma(x_v^T \theta_j^v)]^{d_j^v}\} \\ = & \sum_{v \in V} \sum_{u \in context(v)} \sum_{j=2}^{l_v} \{(1 - d_j^v) \cdot log[\sigma(x_v^T \theta_j^v)] \\ & + d_j^v \cdot log[1 - \sigma(x_v^T \theta_j^v)]\} \end{aligned} \tag{5}$$

where V is the collection of all locations in the dataset, $\sigma(\cdot)$ is the sigmoid function, and θ_j^v corresponds to the parameter of d_j^v.

To obtain the optimal vector representation of v, we need to maximize the above objective L. The stochastic gradient descend method is adopted to optimize L. We calculate the gradient of L to θ_j^v and x_v respectively as follows:

$$\frac{\partial L}{\partial \theta_j^v} = \sum_{v \in V} \sum_{u \in context(v)} \sum_{j=2}^{l_v} (1 - d_j^v - \sigma(x_v^T \theta_j^v)) x_v \tag{6}$$

$$\frac{\partial L}{\partial x_v} = \sum_{v \in V} \sum_{u \in context(v)} \sum_{j=2}^{l_v} (1 - d_j^v - \sigma(x_v^T \theta_j^v)) \theta_j^v \tag{7}$$

Thereby, the updating rule for location embedding x_v can be written as follows:

$$x_v := x_v + \eta \sum_{u \in context(v)} \sum_{j=2}^{l_v} [1 - d_j^v - \sigma(x_v^T \theta_j^v)] \theta_j^v \tag{8}$$

where η is the learning rate during the model training, and it is set to 0.025 according to [36]. As for the dimension size of location vectors and the window size (i.e. the context size) when we train the model using the Skip-gram model, they will be given in the experiments section (See section 5.4).

After the low-dimension vector representation of locations is obtained, we further concatenate the corresponding semantic information to the location vector representation. For example, with regard to the above movie theater in Tokyo (Japan), we will add the category vector Art & $Entertainment$ to the location vector $4b558a35f964a520eae627e3$. Note that the extra category vector has the same dimension as the location vector [2]. In this way, locations of the same type and those are geographically near, or often visited successively by users will be closer in the shared embedding space.

C. User Preference Learning

1) Next Visit: We assume that users' preferences do not change significantly within the short-term. To keep track of a given user's preference at his/her next visit, we adopt a time-decay manner to compute it based on the embeddings of his/her past visits. If a location is visited by this user more recently, it becomes more important and is therefore assigned a higher weight. The preference of a given user u's next visit is calculated in the form of exponential decay:

$$u_{t_n} = \sum_{(u, v_i, t_i) \in C_u \cap (t_i < t_n)} e^{-(t_n - t_i)} \cdot \mathbf{e_{v_i}} \tag{9}$$

where C_u is the collection of all u's check-in records, and $\mathbf{e_{v_i}}$ is the vector representation of location v_i. Note that, the check-ins in C_u are ranked according to their check-in timestamps in an increasing order.

2) Future Visit at A Given Time: When a specific time t is given (in hours), we can calculate the target user u's preference at that time based on his/her past visits. Here, we count the frequency of the location the user has visited during time window t in the training dataset, and then use the normalized frequency as the weight to calculate his/her preference embedding.

$${u_f}^t = \frac{1}{|{C_u}^t|} \sum_{v \in {C_u}^t} {f_v}^u \cdot \mathbf{e_v} \tag{10}$$

where $|{C_u}^t|$ is the number of u's check-ins during time window t in his/her historical records, ${f_v}^u$ is the frequency of location v being visited by this user during time window t, and $\mathbf{e_v}$ is the vector representation of location v.

D. Prediction

We designed two specific methods to predict user's future check-in location in this work. The first method is inspired from the k-nearest neighbor approach (KNN). In the prediction step, the cosine similarity [3] between the target user and all the candidate locations is calculated. Locations that are most similar to the target user are given as the prediction result.

Considering that users tend to visit places that are geographically nearby to each other, we further incorporate a distance factor using kernel density estimation in the prediction step. For each location in the prediction list calculated by KNN method above, we add a normalized distance factor to it (KNN + KDE) according to the following equation.

$$p(l|C_u) = \frac{1}{\sigma|C_u|} \sum_{v_i \in C_u} \varsigma(\frac{dist(v_i, l)}{\sigma}) \tag{11}$$

[2] We adopt the Word2Vec model pre-trained on 3.9G Google news corpus as the category vector representation. If the size of location vector trained using Skip-gram model is n, category vector of the same size n will be concatenated after it.

[3] We have also tried other metrics like the Euclidean distance to measure the similarity between user preference and location vectors, but the prediction performance is significantly worse than using cosine similarity. As a result, we choose cosine similarity as the metric.

where $\varsigma(x)$ is the standard normal distribution function, σ is the optimal bandwidth and $\sigma \approx 1.06\widetilde{\sigma}|C_u|^{\frac{1}{5}}$ according to [22], in which $\widetilde{\sigma}$ is the standard deviation of distance between locations in C_u. This distance factor takes into account the impact of each location in the users' historical records, on the candidate location. The closer the candidate location is to historical check-ins, the higher the probability of it being visited in the future. In this way, we further strengthen the effect of geographical influence on the final location prediction results given by the location embeddings.

V. Experiments

The experiments are conducted on a PC workstation with an Intel Core i7-6700 processor, 24G RAM and 64-bit Ubuntu operating system. The experimentation software is written in Python.

A. Datasets

In this work, we use three public real-world GSN datasets: **NYC**, **TKY**, and **CA** for experimental evaluation. All the datasets are crawled from Foursquare, where **NYC** stands for New York (USA), **TKY** stands for Tokyo (Japan) and **CA** represents California (USA), respectively. The former two datasets **NYC** and **TKY** are collected by [37], and the third dataset **CA** is collected by [25]. For all the three datasets, we remove the less frequent users and locations from each dataset to ensure that each user has at least 10 check-ins, and each location has been visited by at least 10 users. The statistics of the selected datasets are shown in Table I.

TABLE I
Statistics of Datasets

	NYC	**TKY**	**CA**
No. of users	950	2,274	2,259
No. of locations	1,167	2,865	3,244
No. of check-ins	45,632	333,184	111,817

B. Comparison Models

As existing research that can predict both users' next check-in location and users' future check-in location at anytime is quite rare, we will compare our model with several state-of-the-art models on these two tasks separately.

1) Comparison Models For Next Location Prediction: We compare our model against the following state-of-the-art models on next check-in location task:

- Most Popular Locations (MPL1): this model is often selected as a baseline. For prediction, the most popular location in the given user's check-in history is listed as the result.
- Geo-Recency Model (GRM) [21]: this model infuses geo-recency information into a non-negative matrix factorization model, and uses the reconstructed user matrix for location prediction.
- PRME-G [28]: this model embeds user and location into the same latent space to capture the user transition patterns. The geographical influence is incorporated in PRME-G through a simple coefficient. We use the recommended settings of 60 dimensions as given in their paper.
- Graph based Embedding (GE) [6]: this model jointly learns the embeddings of locations, regions, time slots, and auxiliary metadata (i.e. category information) in one common hidden space. The final score of each candidate location is calculated by a linear combination of the inner products for these contextual factors. We use the best hyper-parameters reported in their paper.

2) Comparison Models For Anytime Location Prediction: We compare our model against the following state-of-the-art models on anytime check-in location task:

- Most Popular Locations (MPL2): this model selects the most popular locations during the given time window as the result.
- Multi-feature Model (MfM) [17]: it employs a set of predictive features including user mobility feature, global mobility feature and temporal features to form a supervised learning framework. To exploit the union of these features, a decision tree is trained to compute the ranking score of each candidate location.
- UCGT [24]: this is a collective Bayesian generative model for user check-in behavior prediction. Based on the discovered communities, we can predict a user's future check-in location considering his spatial-temporal preferences. For UCGT model, we set the number of communities to be 40 and the number of topics to be 30 to achieve the best performance.

C. Evaluation Metrics

Given historical check-in records C_u of user u, we rank them according to their check-in timestamps. Then, we use the first 80% check-ins as the training data, the following 10% check-ins as the validation data to tune the model hyper-parameters, and the last 10% check-ins as the testing data.

Two metrics from different aspects are leveraged to evaluate the performance of location prediction in this work. The first metric is $Average\ Percentile\ Rank\ (APR)$, which is often used to measure the extent how a future check-in location can be highly ranked [17, 30]. For each single prediction case, the metric is equal to 1 when the location to be visited is ranked first, and it linearly decreases to 0 when the correct location is ranked at the bottom of the list. The calculation method is as follows.

$$PR = \frac{|L| - rank(k)}{|L|} \tag{12}$$

where $|L|$ is the total number of candidate locations and $rank(k)$ is the rank of the ground-truth location in the list. We average all the prediction cases as the final result.

The second metric is $Accuracy@N$, which is often used to evaluate the accuracy of prediction or recommendation tasks [6, 38]. We successfully predict the future check-in location for a user only when we rank that location in the top-N list. The calculation method is as follows.

$$Acc@N = \frac{\#hit@N}{|L|} \tag{13}$$

where $\#hit@N$ denotes the number of hits in the whole test set. Again, the final result is the average value over all test instances.

D. *Results and Evaluation*

1) Experimental Setup: To evaluate the prediction performance of the Venue2Vec model proposed in this paper, we first have to set suitable hyper-parameters including the balancing coefficient α in the sequence sampling process, the original vector dimension d and window size w when we train the model using Skip-gram language model. The candidate values of the above three variables are $\alpha = \{0.0, 0.1, 0.2, 0.3, ...1.0\}$, $d = \{50, 60, 70, 80, 90, 100\}$ and $w = \{5, 10, 15, 20\}$, respectively.

First of all, we adopt the proposed KNN approach to evaluate the effect of α, which is used to balance the geographical influence and sequential relation between two locaitons. With the other variables d and w remaining fixed, we find that different α values have very similar effects on the prediction results (for example, in the next check-in location prediction task, APR metric on **NYC** dataset is consistently in the range of [0.770,0.773] when α varies). Basically, the closer the value of α is to 0, the higher the predictive performance of the model. The best performance is achieved when the balanc-ing coefficient $\alpha = 0$ for both prediction scenarios, which means no geographical influence factor is exploited at all in model training. This suggests that the geographical influence and the sequential relation do not contain complementary information in the embedding process. In other words, the geographical influence may be encoded within the sequential relation. An intuitive fact may support the above finding, that is, when a user is to visit the next place, he/she will most likely choose a location nearby, rather than a place faraway. Toward the dimension of location embeddings, we find that increasing the value of d improves the performance in terms of APR metric for both prediction tasks. The effect of the increase is obvious when this parameter is less than 40, and it achieves relatively stable performance when d is in the range of [80,100]. In most cases, except for predicting next location on **NYC** dataset, the best results are obtained when the dimension of location vectors reaches the maximal value 100. It is stated in [26] that bigger vector dimension can lead to more accurate performance, but requires more data and more training time. Considering that the total number of locations in these datasets are relatively limited, we do not need to increase vector dimensions. As for the window size, which assigns the maximum distance between the current location and the surrounding locations during model training, we find that different window sizes have little effect on the final results of both prediction tasks. For rigorous consideration, we finally choose the window sizes that produce relatively better performance. The best parameter combination for both prediction scenarios is reported in Table II.

TABLE II
PARAMETER SETUP FOR TWO PREDICTION TASKS ON EACH DATASET

	Next Location Prediction	Anytime Location Prediction
NYC	$\alpha = 0.0$, $d = 80$, $w = 10$	$\alpha = 0.0$, $d = 100$, $w = 10$
TKY	$\alpha = 0.0$, $d = 100$, $w = 10$	$\alpha = 0.0$, $d = 100$, $w = 20$
CA	$\alpha = 0.0$, $d = 100$, $w = 10$	$\alpha = 0.0$, $d = 100$, $w = 20$

To visually illustrate the effect of our model, we plot the vector representation of all locations in the three datasets using t-SNE algorithm [39], which is an efficient dimensionality reduction algorithm for visualization. For all datasets, the results are plotted with the best parameters reported above. Limited by paper space, we only plot location embeddings for next location prediction task. Figure 2 shows the 2D embedding of all locations. To better distinguish the points in each subfigure, we add a categorical mark for each point based on Foursquare Category API [4], including ‘Food’ (F), ‘Art & Entertainment’ (A), ‘Travel & Transport’ (T), and ‘Shop & Services’ (S). In total, 8 different categorical marks are summarized. Figure 2 suggests that locations of the same type cluster well for all three datasets. Although the clustering effect in **CA** dataset is not as obvious as that in **NYC** dataset and **TKY** dataset, the result is still acceptable considering that California has a much larger geographical scope and more specific locations than New York City and Tokyo. As we learn the location embeddings by incorporating sequential relation, geographical influence and semantic information in an unified manner, this observation verifies our assumption that locations of the same type, those that are geographically nearby, and those visited successively by users will be closer situated within the embedding space.

2) Analysis of Experimental Results: Next, we compare our model with other approaches in terms of top-N location predictions. Note that we only show the performance when $N = \{1, 5, 10\}$ since a greater value of N does not make sense in real application scenarios. We show the comparison results among multiple models on next location prediction task in Table III, in which the first two methods KNN and KNN+KDE are implemented based on our Venue2Vec model, followed by other comparison models. The best results are highlighted in bold font on each metric. As we can see, the two proposed prediction methods based on Venue2Vec model outperform other models significantly, which means our model can generally rank the true location at a higher position in the candidate list compared against other models. This observation demonstrates the superiority and effectiveness of the proposed Venue2Vec model on next location prediction task. Also, when we incorporate the distance factor using kernel density estimation for next location prediction (i.e. KNN+KDE), it can produce a better result than if only considering user preference (i.e. KNN). Considering that in the training process, the best performance of Venue2Vec model is achieved when geographical influence factor was ignored ($\alpha = 0$), here we see that prediction process is actually improved by incorporating the distance factor.

Similar results can also be found in Table IV, where the comparison on anytime location prediction task is summarized. As can be seen from Table IV, although UCGT model performs slightly better than KNN, the best performance is still achieved by KNN+KDE. In addition, we can see that

[4] http://api.foursquare.com/v1/categories

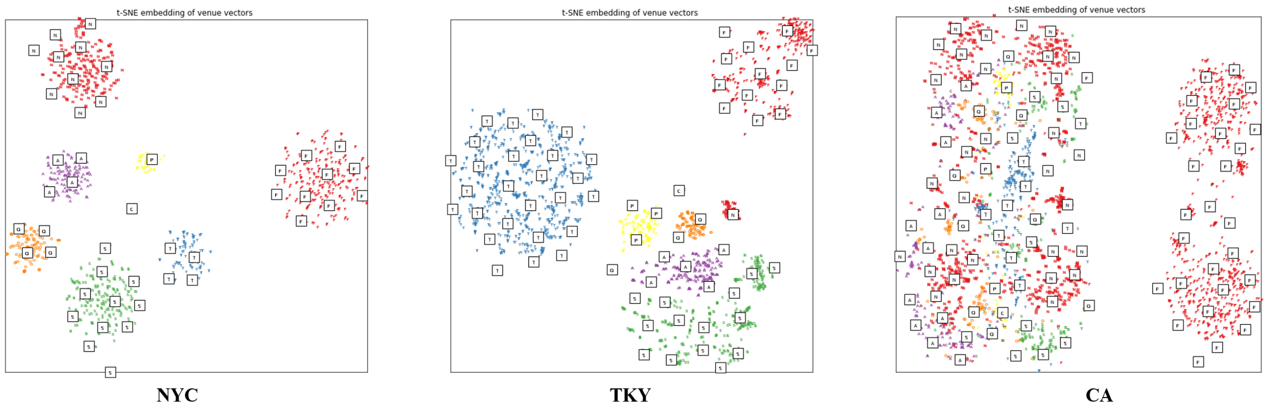


Fig. 2. 2D embedding of locations for next check-in location prediction task, where the original Skip-gram vector dimensions is 80, 100, 100 for **NYC**, **TKY** and **CA** datasets, respectively.

TABLE III
COMPARISON OF USER NEXT LOCATION PREDICTION RESULTS OVER THREE DATASETS USING APR AND $Accuracy@N$ METRICS.

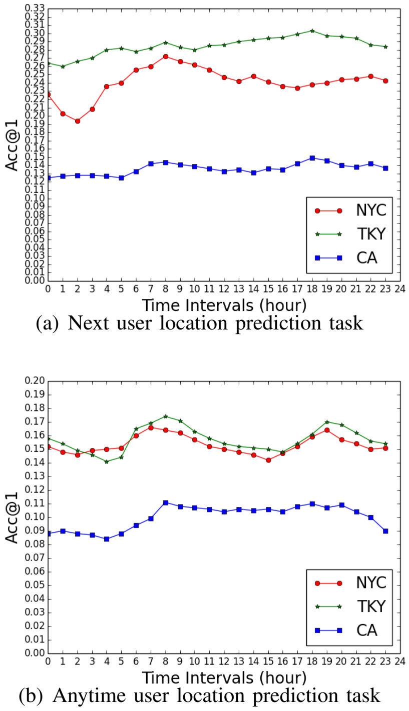
	NYC				TKY				CA			
	$Acc@1$	$Acc@5$	$Acc@10$	APR	$Acc@1$	$Acc@5$	$Acc@10$	APR	$Acc@1$	$Acc@5$	$Acc@10$	APR
KNN	0.215	0.272	0.318	**0.773**	0.213	0.330	0.417	0.846	0.134	0.180	0.215	0.830
KNN+KDE	**0.248**	**0.310**	**0.351**	0.760	**0.290**	**0.379**	**0.431**	**0.862**	**0.148**	**0.197**	**0.237**	**0.835**
MPL1	0.157	0.188	0.206	0.674	0.148	0.185	0.206	0.833	0.124	0.165	0.195	0.768
GRM	0.171	0.216	0.282	0.695	0.205	0.262	0.324	0.845	0.126	0.160	0.183	0.762
PRME-G	0.166	0.197	0.264	0.688	0.188	0.237	0.309	0.841	0.112	0.147	0.173	0.755
GE	0.219	0.297	0.346	0.767	0.261	0.340	0.392	0.858	0.140	0.183	0.222	0.828

TABLE IV
COMPARISON OF USER ANYTIME LOCATION PREDICTION RESULTS OVER THREE DATASETS USING APR AND $Accuracy@N$ METRICS.

	NYC				TKY				CA			
	$Acc@1$	$Acc@5$	$Acc@10$	APR	$Acc@1$	$Acc@5$	$Acc@10$	APR	$Acc@1$	$Acc@5$	$Acc@10$	APR
KNN	0.136	0.180	0.205	0.713	0.118	0.146	0.183	0.803	0.088	0.121	0.146	0.769
KNN+KDE	**0.166**	**0.202**	**0.237**	**0.722**	**0.173**	**0.214**	**0.265**	**0.834**	**0.104**	**0.132**	**0.165**	**0.815**
MPL2	0.106	0.129	0.152	0.671	0.088	0.111	0.136	0.774	0.078	0.110	0.139	0.714
MfM	0.137	0.186	0.203	0.679	0.114	0.138	0.172	0.792	0.083	0.117	0.142	0.728
UCGT	0.142	0.198	0.226	0.715	0.122	0.183	0.251	0.816	0.094	0.128	0.157	0.773

the metric values of anytime location prediction are generally lower than that of next location prediction. This is because the calculation of a user's preference at a given future time only considers his/her historical check-in records during the given time window, which essentially leads to a sparse profile of the user. In contrast, the calculation of a user's preference at next visit takes into account all his/her historical check-in data through the time-decay manner, which avoids the instability of the prediction results brought by this kind of data sparseness to some extent. From the aforementioned experimental results, we can also observe that the location prediction method $KNN{+}KDE$ based on Venue2Vec model has outstanding performance in the metric $Acc@1$, which has significant commercial value corresponding to practical application. Considering that we can rank a location a user will visit in the future at the Top-1 position, it is a remarkable performance as there are tens of thousands places to be ranked in a city. In summary, the location prediction techniques based on the proposed Venue2Vec model show effectiveness and superiority in both user location prediction scenarios.

We further analyze the effectiveness of our model within different time intervals during a day. We divide the time of day into 24 ordered hours, and study the location prediction performance of Venue2Vec model over time intervals in two user location prediction tasks. To make the analysis universal, we display the result curve in terms of $Acc@1$ computed by the prediction method $KNN{+}KDE$ on each dataset in Figure 3. The prediction accuracy during daytime is generally higher than that during night time. In most cases, the best performance appears in the $8th$ hour in the morning and the $19th$ hour in the afternoon. This is likely due to regular activities such as taking public transportation during these time intervals, however, when people finish work, they tend to have more personalized activities in the evening which can be much harder to predict. Also, the location prediction accuracy based on **NYC** and **TKY** datasets is significantly higher than that based on **CA**. This is because the first two datasets only cover users' check-in data in a single city, while the **CA** covers the check-in data of users throughout the state of California, hence a broader geographical scope. Under the premise that the data size of per user is not much different, the check-in data in **CA** is more sparse, which results in lower predictability.



(a) Next user location prediction task

(b) Anytime user location prediction task

Fig. 3. Curve of Acc@1 with different time intervals on both user location prediction tasks.

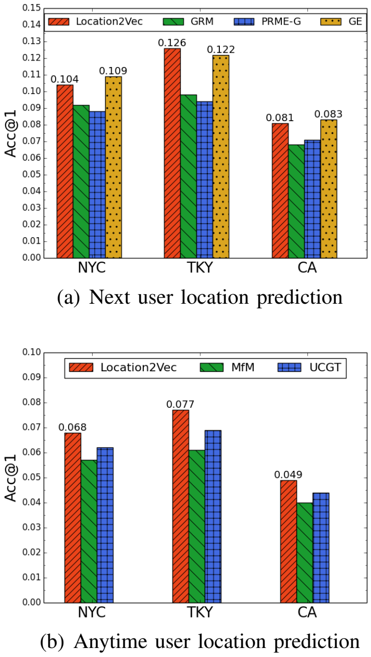
We also summarize the location prediction performance of two tasks in terms of $Acc@1$ for both weekdays (from Monday to Friday) and weekends (Saturday and Sunday) in Table V. As can be seen, the location prediction accuracy in weekdays is significantly higher than that at weekends. This is likely because people's mobility at weekends is uncertain compared with that in weekdays. They tend to take relatively unusual activities like sporting and hiking at weekends, rather than regular activities such as travelling to work on weekdays.

TABLE V
$Acc@1$ METRIC OF TWO LOCATION PREDICTION TASKS ON WEEKDAYS AND WEEKENDS.

	Next Location Prediction			Anytime Location Prediction		
	NYC	**TKY**	**CA**	**NYC**	**TKY**	**CA**
Weekdays	**0.259**	**0.298**	**0.166**	**0.185**	**0.188**	**0.121**
Weekends	0.214	0.273	0.142	0.150	0.164	0.083

3) Analysis on Cold-start Problems: As many new point-of-interests (e.g. restaurants and shops) are emerging every day in our real life and many new users who have not got any preference history are joining the Geo-social networks, location prediction under such cold-start conditions is becoming indispensable. Here, we evaluate the performance of Venue2Vec model for cold-start scenario. It is worth noting that since each dataset is preprocessed to retain only active users and locations (see Section 5.1), we therefore randomly select 100 inactive users that were excluded from the training dataset for evaluation. For evaluation purpose, when we select these inactive users, we restrict that all the locations they have visited in their historical check-in records be included in the training set.

For Venue2Vec model, when calculating users' preference under cold-start condition, we adopt the following way for both location prediction tasks: if the target user has only one check-in record, we take the average preference of all active users in the training dataset as his/her preference; and if the target user has more than one check-in records, we still calculate his/her preference according to the proposed methods in Section 4.3. The experimental results in terms of $Acc@1$ are shown in Figure 4. We can observe that, for next user location prediction task, the performance of Venue2Vec model is higher than that of GRM and PRME-G under cold start conditions, but GE performs equivalent and even slightly better than Venue2Vec model. This is mainly because GE model can still learn vector representations for cold-start users by embedding them into the shared latent space. Indeed, GE model leverages not only the geographical influence and semantic information, but also the sequential relation as well as the temporal cyclic effect lying in users' check-in data. In contrast, GRM and PRME-G can only use the geographical information. For anytime user location prediction task, however, Venue2Vec model performs consistently better than its competitors. This suggests that incorporating multiple contextual factors is positive for alleviating the cold-start problem in anytime location prediction. In general, the proposed Venue2Vec model has certain advantages for cold-start scenario.



(a) Next user location prediction

(b) Anytime user location prediction

Fig. 4. Cold-start $Acc@1$ comparison on two user location prediction tasks.

4) Analysis of Time Efficiency: Apart from having desirable prediction performance, the proposed Venue2Vec model is also time efficient as it can be scalable using a parallel implementation. The location sequence sampling process and model training as well as model testing can be easily parallelized on multicore machines. In terms of the sampling process, several random walkers in different threads can simultaneously explore different parts of the location network. As we mentioned in Section 4.1, since the sampling steps are flexible to small changes in the network structure without the need for global re-computation, the overall location prediction model can be time efficient. For model training, a multi-worker similar to Python word2vec multiprocessing [5] is adopted to parallelize the location embedding process. For model testing, as we predict the location of each check-in record per user in the testing data separately, it can be naturally parallelizable.

We compare the time consumption of Venue2Vec model and other comparison models during model training and testing. For the sake of fairness, we ignore the time consumption during the preparation process such as data preprocessing and location sampling (because only Venue2Vec model involves sampling steps). The time spent on each dataset using our PC (4 physical cores and 8 threads) to train the model and to predict the check-in location of all target users are presented in Figure 5. The presented time results for model training are calculated when the best hyper-parameters are fixed for each model. As the next user location prediction task and anytime user location prediction task share the model training stage, we only show the results for next location prediction task. As we can observe, the time consumed by each model for testing on each dataset is almost the same, because each model calculates the cosine similarity between user preference and location vector when predicting users' next check-in location, after which the final result is produced by sorting the user-location ratings. However, Venue2Vec model consumes much less time than other models during model training due to the multi-worker implementation. It can save up to 80% training time on average using different datasets, which shows that Venue2Vec model is scalable for large-scale GSNs. Comparing to other models that may delay on model updates caused by the expensive time cost of re-running, the proposed Venue2Vec model is superior in terms of time efficiency.

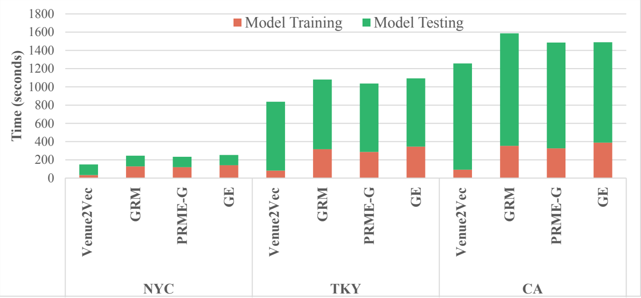


Fig. 5. Time consumption for model training and model testing in next user location prediction task.

5) Further discussion: Our experimental results reveal the effectiveness and the superiority of the proposed Venue2Vec model. Although we study the location prediction problem only based on three Foursquare datasets, Venue2Vec model has strong generalization ability for all kinds of Geo-social networks since it comprehensively considers various complimentary factors. It is a generic flexible model that can be extended to incorporate other factors, not limited to temporal-spatial context, semantic information and sequential relation.

However, there are some deficiencies in our current model. For example, we currently ignore friendship relation when calculating users' preference at next visit and at the future time, however friendships will naturally play an important role in where a user may visit. We also found that the number of unique locations in these experimental datasets is quite limited, which will affect the generalization of the experimental results. Finally, we would like to extent our approach to consider additional user-generated content such as text, pictures, and ratings, that again would likely increase the predictive performance for future use.

VI. Conclusion and Future Works

In this paper, we study the problem of users' check-in location prediction in Geo-Social Networks. We present a novel embedding model called Venue2Vec which naturally encodes temporal-spatial context, semantic information as well as sequential relation so as to improve location prediction performance and meanwhile increase the interpretability of the model. We conduct extensive experiments to evaluate the performance of our model on three real-world GSN datasets. The results reveal the effectiveness and the superiority of Venue2Vec model. Besides, our model also proves to be time efficient.

As for future work, we can further consider the impact of friendship or community on users' check-in location prediction. Additionally, the multi-modal data such as images, ratings, and textual reviews associated with the locations can be explored.

Acknowledgment

This work is supported by National Natural Science Foundation of China under Grants No. 61772133, No. 61472081, No. 61402104, No. 61370207, No. 61370208, No. 61300024, No. 61320106007, Key Laboratory of Computer Network Technology of Jiangsu Province, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9.

References

[1] H. Li, Y. Ge, H. Zhu, and H. Zhu, "Point-of- interest recommendations: Learning potential check-ins from friends," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 975–984.

[2] T. Hu, R. Song, Y. Wang, X. Xie, and J. Luo, "Mining shopping patterns for divergent urban regions by incorporating mobility data," in *ACM International on Conference on Information and Knowledge Management*, 2016, pp. 569–578.

[5] https://rare-technologies.com/parallelizing-word2vec-in-python/

[3] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering urban functional zones using latent activity trajectories," *Knowledge and Data Engineering IEEE Transactions on*, vol. 27, no. 3, pp. 712–725, 2015.

[4] Y. Jiang, W. He, L. Cui, and Q. Yang, "User location prediction in mobile crowdsourcing services," in *International Conference on Service-Oriented Computing*. Springer, 2018, pp. 515–523.

[5] A. Likhyani, D. Padmanabhan, S. Bedathur, and S. Mehta, "Inferring and exploiting categories for next location prediction," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 65–66.

[6] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based POI embedding for location-based recommendation," in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, Indianapolis, IN, USA, October 24-28*, 2016, pp. 15–24.

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

[8] Y. Long, P. Zhao, V. S. Sheng, G. Liu, J. Xu, J. Wu, Z. Cui, Y. Long, P. Zhao, and V. S. Sheng, "Social personalized ranking embedding for next poi recommendation," in *The 18th International Conference on Web Information Systems Engineering, Puschino, Russia, October 7-11*, 2017, pp. 91–105.

[9] J. B. Gomes, C. Phua, and S. Krishnaswamy, "Where will you go? mobile data mining for next place prediction," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2013, pp. 146–158.

[10] H. Zhang, Z. Chen, Z. Liu, Y. Zhu, and C. Wu, "Location prediction based on transition probability matrices constructing from sequential rules for spatial-temporal k-anonymity dataset," *Plos One*, vol. 11, no. 8, 2016.

[11] J. D. Zhang, Y. Li, and Y. Li, "Lore: exploiting sequential influence for location recommendations," in *ACM Sigspatial International Conference on Advances in Geographic Information Systems*, 2014, pp. 103–112.

[12] E. Bart, R. Zhang, and M. Hussain, "Where would you go this weekend? time-dependent prediction of user activity using social network data," in *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, 2013, pp. 669–672.

[13] S. Li, X. Wang, S. Zhao, J. Wang, and L. Li, "Local semidefinite programming-based node localization system for wireless sensor network applications," *IEEE Systems Journal*, vol. 8, no. 3, pp. 879–888, Sept 2014.

[14] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific Reports*, vol. 3, p. 1376, 2013.

[15] H. Gao, J. Tang, X. Hu, and H. Liu, "Modeling temporal effects of human mobile behavior on location-based social networks," in *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*. ACM, 2013, pp. 1673–1678.

[16] H. Gao, J. Tang, and X. Hu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of the 7th ACM Conference on Recommender Systems*. ACM, 2013, pp. 93–100.

[17] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 1038–1043.

[18] C. Petersen, J. G. Simonsen, and C. Lioma, "Power law distributions in information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 34, no. 2, p. 8, 2016.

[19] H. Gao, J. Tang, and H. Liu, "Exploring social-historical ties on location-based social networks," in *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, 2012, pp. 114–121.

[20] D. Lian, Y. Zhu, X. Xie, and E. Chen, "Analyzing location predictability on location-based social networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 102–113.

[21] R. Assam, S. Sathyanarayana, and T. Seidl, "Infusing geo-recency mixture models for effective location prediction in lbsn," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 855–863.

[22] J. D. Zhang and C. Y. Chow, "Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 443–452.

[23] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, "Geo-sage: A geographical sparse additive generative model for spatial item recommendation," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1255–1264.

[24] H. Yin, Z. Hu, X. Zhou, H. Wang, K. Zheng, Q. V. H. Nguyen, and S. Sadiq, "Discovering interpretable geo-social communities for user behavior prediction," in *IEEE International Conference on Data Engineering*, 2016, pp. 942–953.

[25] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq, "Joint modeling of user check-in behaviors for real-time point-of-interest recommendation," *Acm Transactions on Information Systems*, vol. 35, no. 2, p. 11, 2016.

[26] M. G. Ozsoy, "From word embeddings to item recommendation," *arXiv preprint arXiv:1601.01356*, 2016.

[27] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *The 26th International World Wide Web Conference, Perth, Australia, April 3-7*, 2017, pp. 173–182.

[28] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new poi recommendation," in *International Conference on Artificial Intelligence*, 2015, pp. 2069–2075.

[29] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 403–412.

[30] J. Cao, S. Xu, X. Zhu, R. Lv, and B. Liu, "Effective fine-grained location prediction based on user check-in pattern in lbsns," *Journal of Network and Computer Applications*, vol. 108, pp. 64 – 75, 2018.

[31] O. Barkan and N. Koenigstein, "Item2vec: Neural item embedding for collaborative filtering," pp. 1–6, 2016.

[32] C. Musto, G. Semeraro, M. de Gemmis, and P. Lops, "Word embedding techniques for content-based recommender systems: An empirical evaluation," in *ACM Conference on Recommender Systems*, vol. 1441, 2015.

[33] J. Manotumruksa, C. Macdonald, and I. Ounis, "Modelling user preferences using word embeddings for context-aware venue recommendation," 2016.

[34] B. Perozzi, R. Alrfou, and S. Skiena, "Deepwalk: online learning of social representations," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, August 24 - 27*, 2014, pp. 701–710.

[35] H. Tong, C. Faloutsos, and J. Y. Pan, "Fast random walk with restart and its applications," in *International Conference on Data Mining*, 2006, pp. 613–622.

[36] T. Mikolov, G. Corrado, K. Chen, J. Dean, T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013, pp. 1–12.

[37] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[38] S. Li, T. Tryfonas, G. Russell, and P. Andriotis, "Risk assessment for mobile systems through a multilayered hierarchical bayesian network," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1749–1759, Aug 2016.

[39] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.