Joint Representation Learning for Location-based Social Networks with Multi-Grain Sequential Context

Abstract

This paper studies the problem of learning embedding representations for Location-Based Social Networks (LBSN), which is useful in many tasks such as location recommendation and link prediction. Existing network embedding methods mainly focus on capturing topology patterns reflected in social connections, hence, the important data type, *i.e.,* check-in sequences, cannot be modeled. In this paper, we propose a representation learning method for LBSNs called as *JRLM*, which jointly model both social connections and check-in sequences. To capture sequential relatedness, JRLM characterizes two levels of sequential contexts, namely fine-grained and coarse-grained contexts. We present a learning algorithm tailored to the hierarchical architecture of the proposed model. We conduct extensive experiments on two important applications using real-world datasets. The experimental results demonstrate the superiority of our model. The proposed model can generate representations for both users and locations in the same embedding space, which can be exploited in multiple LBSN tasks.

1 Introduction

Location-based services, such as *Foursquare* and *Gowalla*, provide a convenient way to keep users' mobility information using check-in records. As reported in September 2015, Foursquare has accumulated over 7 billion check-ins made by 55 million users.[1] It is notable that such services also provide the mechanism of social connections for users to follow whom they are interested in. A combination between social networking and check-in services has led to a specific style of social networks, termed as *Location-Based Social Networks* (LBSNs) [Zheng, 2015]. Due to the increasing importance of LBSNs, extensive studies have been conducted to improve various applications in LBSNs, including location recommendation [Bao *et al.*, 2015], link prediction [Pham *et al.*, 2013] and user interest modeling [Ye *et al.*, 2013].

Although various methods have been proposed in the literature for LBSN services [Zheng, 2015], an important problem is lack of consideration, *i.e.,* how to construct an effective task-independent data representation way. Such a modeling way should be able to capture the essential data characteristics, further improve multiple related applications in LBSNs. With the popularity of deep learning techniques, a major progress made in social networking analysis is learning distributed representations of vertices in a network (*a.k.a.* network embedding) [Perozzi *et al.*, 2014; Tang *et al.*, 2015b]. These latent representations encode social relations in a continuous vector space, which can be readily exploited to solve multiple tasks as a unified data representation. Inspired by this progress, the focus of this paper is how to effectively model data characteristics for LBSNs using distributed (*a.k.a.* embedding) representation method.

Different from traditional social networks, check-in data is an important data type in LBSNs, present as a sequence of chronologically ordered records with visiting information on locations. Existing network embedding methods mainly capture the network topology information, which cannot effectively characterize check-in data in LBSNs. Several studies have made extensions on social representation learning, *e.g.,* incorporating text information [Yang *et al.*, 2015] or label information [Tang *et al.*, 2015a], however, they are not specially designed for characterizing sequential information of check-in data. As shown in [Li *et al.*, 2008; Feng *et al.*, 2015], sequential influence is an important kind of characteristics to consider for LBSN data. Hence, the representation learning method should carefully consider sequential characteristics for LBSNs.

To address the above concerns, we propose a novel Joint Representation Learning Model (JRLM) for LBSN data. Our model takes a social network together with users' check-in records as input and produces a latent representation for each user and location (*a.k.a.* POI) as output. Our approach is inspired by recent progress in deep learning for Natural Language Processing [Mikolov *et al.*, 2013a], which shows that sequential semantic relatedness can be effectively captured by surrounding word contexts. Hence, we start with a base model, *i.e.,* JRLM, characterizing check-in sequences using a similar way to model word sequences (*e.g., word2vec*). Based on such a sequence modeling, JRLM additionally sets up a specific component to characterize social connections. These two parts lying in different embedding layers are further combined through the shared users. The original sequence model-

[1] https://foursquare.com/about/

ing method only captures fine-grained location-level sequential contexts, in which coarse-grained sequential correlations have not been considered. For check-in sequences, coarse-grained correlations plays an important role to uncover the underlying data characteristics. For example, when a user is on vocation in ski-related areas, her check-in behaviors in consecutive vocation days should be highly correlated. To capture such coarse-grained correlation patterns, we extend the base model by incorporating multi-grain sequential contexts, called *JRLM++*. JRLM++ considers both fine-grained and coarse-grained sequential contexts, further generates embedding representations for both users and locations.

By jointly leveraging both social connections and check-ins, JRLM++ is effective to capture the characteristics of these two data types in LBSNs. The output can be provided as a unified data representation for multiple related tasks. We demonstrate the effectiveness of JRLM++ on two important tasks using real-world LBSN datasets. The contributions of this paper are summarized as follows:

- We develop a distributed representation model named JRLM++ to jointly model both social connections and check-in sequences. Such a model can generate representations for both users and locations in the same embedding space.
- We incorporate multi-grain sequential contexts into JRLM++, and present an effective approach to characterize hierarchical contexts for sequence data. Both location-level and segment-level sequential contexts are considered. We also present a learning algorithm tailored to the hierarchical model architecture.
- We conduct extensive experiments on two important applications using real-world datasets. Experimental results demonstrate the effectiveness of our model. The proposed model can be easily exploited in multiple LBSN tasks as a general data representation way.

2 Related Work

Our work is mainly related to the following research studies.

Mining LBSN Data. Early researchers [Pei *et al.*, 2001; Yan *et al.*, 2003] mainly focus on finding interesting patterns from trajectory sequences. Recently, various studies have been proposed to improve the location-based services, including location recommendation [Zheng *et al.*, 2010; Cheng *et al.*, 2013; Yin *et al.*, 2013; Yuan *et al.*, 2014; Lian *et al.*, 2014], link prediction [Li *et al.*, 2008; Wang *et al.*, 2011; Gao *et al.*, 2012; Pham *et al.*, 2013] and user preference modeling [Ye *et al.*, 2011; 2013]. Here, we give a brief review of several representative studies. Yuan *et al.* [2014] utilized weighted matrix factorization to further incorporated accurate geographical active locations of users for location recommendation. Yin *et al.* [2013] proposed a location-content-aware recommender system by considering both personal interest and local preference. Gao *et al.* [2012] proposed a geo-social correlation model to capture social correlations on LBSNs in order to solve "cold start" location prediction problem. Pham *et al.* [2013] calculated the diversity of co-occurences for each user pair and the weighted location entropy for each co-visited location proposed to estimate the social strength between users. Ye *et al.* [2011] observed a clustering phenomenon of visited locations of each individual person with considering their social connections.

Different from these studies, our focus is not to solve a single application, instead we aim to provide a general data representation approach. Our model can produce both embedding representations for two key kinds of entities in LBSNs, *i.e.,* user and location. As seen later, it will be easy to apply our model to various LBSN tasks. Especially, our representations can be used as input to improve other models.

Distributed Representation Learning. Recently, neural language models, such as *word2vec* [Mikolov *et al.*, 2013a], *GloVe* [Pennington *et al.*, 2014] and *paragraph2vec* [Le and Mikolov, 2014], have generalized the classic n-gram language models by using continuous variables to represent words in a vector space. Particularly, the concept of distributed representations has been extended beyond pure language words to social networks [Tang *et al.*, 2015b; Perozzi *et al.*, 2014]. Although several attempts [Tang *et al.*, 2015a; Yang *et al.*, 2015] have been made to incorporate auxiliary information into network embedding model, they mainly consider textual information.

We have made two major extensions based on these studies: (1) We jointly model both social connections and check-in sequences in LSBNs data simultaneously, and project users and locations in the same vector space. (2) We characterize multi-grain sequential contexts in the representation model to better capture the sequential relatedness.

3 Formalization

Given a location-based social network (LBSN), the social connections between users are available. Formally, the social graph can be described as $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{U}$ denotes the vertex set on the graph (*i.e.,* the user set) and $\mathcal{E}$ denotes the edge set among the user vertices. Given two users $u, u' \in \mathcal{U}$, an undirected edge $e = (u, u') \in \mathcal{E}$ indicates that u and u' are a pair of friends.[2] Further, let $\mathcal{F}^{(u)}$ denote the friend set for user u and $f_i^{(u)}$ denote the i-th friend for user u.

Besides social networking information, check-in information is also available on LBSNs. Let $\mathcal{L}$ denote a set of locations. When a user $u \in \mathcal{U}$ check-ins a location $\ell \in \mathcal{L}$ at timestamp t, the information can be recorded as a triplet $\langle u, \ell, t \rangle$. An example triplet $\langle$`USER316, Little Heaven Deli, 2016-02-02/12:10pm`$\rangle$ tells that a user (UserID 316) visited a restaurant named "*Little Heaven Deli*" at 12:10pm on February 2, 2016. Given a time span, a user u typically generates a sequence of chronologically ordered triplets: $\langle u, \ell_1, t_1 \rangle$, ..., $\langle u, \ell_i, t_i \rangle$, ..., $\langle u, \ell_{N_u}, t_{N_u} \rangle$, where N_u is the sequence length of the entire check-in history for user u.

In a general sense, embedding aims to project an informational entity into a low-dimensional vector space. In LBSNs, we have two key types of entities for concern, namely users and locations. Formally, our task is to learn $\boldsymbol{v}_u \in \mathbb{R}^D$ and

[2] We follow the typical settings in LBSNs that the friendship is bi-directional, *e.g.,* Foursquare.

$\boldsymbol{v}_\ell \in \mathbb{R}^D$ for each user u in $\mathcal{U}$ and each location $\ell \in \mathcal{L}$ respectively, where D is the number of latent dimensions (*a.k.a.* embedding dimensions). These learnt embeddings are expected to be useful as general data representations in various LBSN tasks such as link prediction and location recommendation.

4 Joint Representation Learning for Users and Locations on LBSNs

We have two types of entities for embedding representation. For users, the data patterns are characterized via connections on the social graph (*i.e.,* social connectedness), while for locations, the data is present as check-in sequences (*i.e.,* sequential relatedness). We first start with a base model which jointly models both social connectedness and sequential relatedness. Then we enhance the base model by characterizing multi-grain sequential context among locations.

4.1 Joint Embedding Users and Locations: A Base Model

For convenience, we first define a general formulation of the softmax conditional probability as follows

$$Pr(y|x) = \frac{\exp\left(\boldsymbol{v}_x^\top \cdot \boldsymbol{v}_y\right)}{\sum_{y'} \exp\left(\boldsymbol{v}_x^\top \cdot \boldsymbol{v}_{y'}\right)}, \tag{1}$$

where $\boldsymbol{v}_x$ and $\boldsymbol{v}_y$ are the D-dimensional input and output vectors (placeholders for replacement) respectively. Our model needs to preserve both social connectedness and sequential relatedness.

To characterize the *social connectedness*, we adopt the idea from Skip-Gram [Mikolov *et al.*, 2013b]. We consider the current user as the context, and then generate her friends one by one. Formally, we have the log likelihood function

$$L_{graph} = \sum_{u \in \mathcal{U}} \sum_{f_i^{(u)} \in \mathcal{F}^{(u)}} \log Pr(f_i^{(u)}|u), \tag{2}$$

where $f_i^{(u)}$ is the i-th friend of user u, $Pr(f_i^{(u)}|u)$ are modeled using Eq. 1, and we set: $\boldsymbol{v}_x = \boldsymbol{v}_u$ and $\boldsymbol{v}_y = \boldsymbol{v}_{f_i^{(u)}}$.

To characterize the *sequential relatedness*, given a check-in record at ℓ_j, we take a fixed-length window consisting of the previous K_0 and successive K_0 check-ins as contexts. Let ℓ_{j-K_0}:ℓ_{j+K_0} denote the location sequence ℓ_{j-K_0}, ℓ_{j-K_0+1}, ..., ℓ_{j+K_0} (excluding ℓ_j). This idea resembles to that in word embedding models (*e.g.,* word2vec [Mikolov *et al.*, 2013a]), where the semantics of a word is related to those of the surrounding words. Thus we have the following formulation:

$$\overline{\boldsymbol{v}}_{\ell_j} = \frac{1}{2K_0+1}\Bigg(\sum_{-K_0 \le k \le K_0, k \ne 0, 0 \le j+k < N} \boldsymbol{v}_{\ell_{j+k}} + \boldsymbol{v}_u\Bigg), \tag{3}$$

where $\overline{\boldsymbol{v}}_{\ell_j}$ is an aggregated contextual embedding vector for ℓ_j and N is the sequence length. In Eq. 3, for joint embedding, we incorporate user u into the generation of her check-in records. With this formulation, we can model the sequential relatedness by

$$L_{loc} = \sum_{u \in \mathcal{U}} \sum_{\ell_{l_j} \in \mathcal{C}^{(u)}} \log Pr(\ell_j | \ell_{j-K_0} : \ell_{j+K_0}, u), \tag{4}$$

where $\mathcal{C}^{(u)}$ is the historical check-in records for u and we characterize $Pr(\ell_j|\ell_{j-K_0} : \ell_{j+K_0}, u)$ with Eq. 1 having $\boldsymbol{v}_x = \boldsymbol{v}_{\ell_{l_j}}$ and $\boldsymbol{v}_y = \overline{\boldsymbol{v}}_{\ell_j}$. We couple both factors (*i.e.,* L_{graph} and L_{loc}) by sharing user contexts. The base model is defined as a sum of both factors as follows

$$L = L_{graph} + L_{loc}. \tag{5}$$

We denote this model by **JRLM** (Joint Representation Learning Model).

4.2 Integrating Multi-Grain Sequential Context

The base model characterizes the sequential relatedness by considering the previous K_0 and successive K_0 locations as contexts. Such a formulation only considers fine-grained (*i.e.,* location-level) contexts, while coarse-grained contexts cannot be captured. For example, when a user is on vocation in ski-related areas, her overall check-in behaviors in consecutive vocation days should be highly correlated. To better characterize check-in sequences, we need a approach to capture multi-grain sequential context.

We first split users' check-in records into ordered *segments* (or *subsequences*). Following [Ye *et al.*, 2009], we split check-in sequences with equal-length time intervals, *i.e.,* a day. The proposed method can be easily extended to other time granularity. Formally, the entire check-in sequence of user u is split into $M^{(u)}$ consecutive trajectory segments, denoted by a set $\mathcal{C}^{(u)}$. Each segment has the equal-length time span. The m-th segment is denoted by $s_m^{(u)}$, consisting of the check-in records from u in the m-th time interval.

Next, we present the proposed model. Our ideas can be summarized as two points: (1) The check-ins in a segment are influenced by the overall behaviors of the segment; (2) Check-in behaviors in consecutive segments are correlated. The first point captures fine-grained sequential contexts while the second point captures coarse-grained sequential contexts.

To characterize the first aspect, we also set a D-dimensional embedding vector $\boldsymbol{v}_{s_m^{(u)}}$ for a segment $s_m^{(u)}$. With the segment vectors, we can rewrite the Eq. 4 as follows

$$L_{loc} = \sum_{u \in \mathcal{U}} \sum_{m=1}^{M^{(u)}} \sum_{\ell_{l_j} \in s_m^{(u)}} \log Pr(\ell_j | \ell_{j-K_0} : \ell_{j+K_0}, s_m^{(u)}, u), \tag{6}$$

where ℓ_j belongs to the segment $s_m^{(u)}$, and we incorporate the segment vector into the contextual vector in Eq. 3

$$\overline{\boldsymbol{v}}_{\ell_j} = \frac{1}{2K_0+2}\Bigg(\sum_{-K_0 \le k \le K_0, k \ne 0, 0 \le j+k < N} \boldsymbol{v}_{\ell_{j+k}} + \boldsymbol{v}_u + \boldsymbol{v}_{s_m^{(u)}}\Bigg). \tag{7}$$

To characterize the second aspect, we model the sequential relatedness among segments to capture coarse-grained contexts

$$L_{seg} = \sum_{u \in \mathcal{U}} \sum_{m=1}^{M^{(u)}} \log Pr(s_m^{(u)} | s_{m-K_1}^{(u)} : s_{m+K_1}^{(u)}, u), \tag{8}$$

where $Pr(s_m^{(u)}|s_{m-K_1}^{(u)} : s_{m+K_1}^{(u)}, u)$ is modeled using a similar way as $Pr(\ell_j|\ell_{j-K_0} : \ell_{j+K_0}, u)$ in Eq. 4, *i.e.,* each segment vector is generated based on the previous K_1 and successive K_1 segments using the softmax conditional probability in Eq. 1. We use different lengths K_0 and K_1 for locations and segments respectively.

To integrate Eq. 2, 6 and 8, we can obtain the final model

$$L = L_{graph} + L_{loc} + L_{seg}. \tag{9}$$

We keep the same formulation for social connections from Eq. 2, and further characterize two kinds of sequential contexts. The user vectors connect the three coupling factors. Currently multi-grain is simply for two-levels of granularity (*i.e.,* "`daily segment` $\rightarrow$ `check-in`"), namely location level and segment level, the presented approach can be extended to model more levels of granularity (*e.g.,* "`weekly segment` $\rightarrow$ `daily segment` $\rightarrow$ `check-in`"). The user context is incorporated in both levels, hence, it plays a key role in capturing users' overall visiting behaviors.

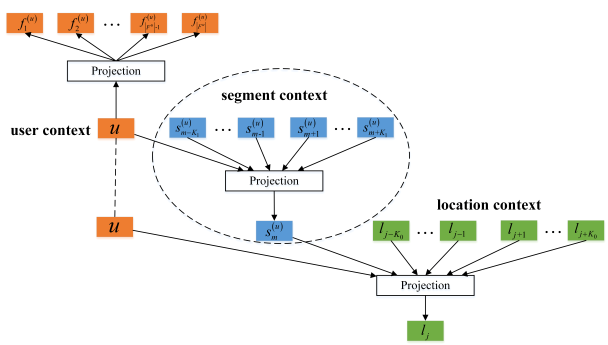


Figure 1: The architecture of the proposed model with hierarchical sequential contexts. Orange, blue and green boxes denote the embedding vectors for users, segments and locations respectively.

We denote this improved model as **JRLM++**. An overview of the proposed JRLM++ is shown in Figure 1. As a comparison, the architecture will degenerate to the base model JRLM when ignoring the part in dotted circle.

4.3 Parameter Learning

Our parameters include the embedding vectors of users $\{\boldsymbol{v}_u\}$, segments $\{\boldsymbol{v}_s\}$ and locations $\{\boldsymbol{v}_l\}$. Unlike [Mikolov *et al.*, 2013a], we do not discriminate between emitting vectors and output vectors. Our objective function involves three parts. The embedding vectors have been present in a hierarchical manner. Hence, we propose a hierarchical learning method for JRLM++ in Alg. 1.

At an iteration, it scans triplets one by one. A triplet (u, s, ℓ) denotes that a check-in location ℓ contained in the segment s generated by user u. The triplet (u, s, ℓ) will trigger the updating of $\boldsymbol{v}_u$, $\boldsymbol{v}_s$ and $\boldsymbol{v}_\ell$; when the check-ins in s have been scanned once, the pair (u, s) updates the parameters of $\boldsymbol{v}_u$ and $\boldsymbol{v}_s$; finally, we update the user embedding $\boldsymbol{v}_u$. In order to learn these parameters, we need to maximize the log probability defined in Eq. 2, 6 and 8. We adopt Stochastic Gradient Descent (SGD) to update the parameters. While directly optimizing the objective function is time consuming and impractical, since the cost of learning the parameters in Eq. 1 is extremely high. We use the hierarchical softmax technique [Mikolov *et al.*, 2013b] to accelerate the learning speed. The hierarchical softmax uses a binary tree representation for every location, segment and user as its leaves, and each node is explicitly associated with an embedding vector for computing the relative probability to take one branch. Each leaf can be reached by an appropriate path from the root of the tree. Let $\mathcal{N}$ denotes the set of all the leave nodes. Instead of evaluating all the $|\mathcal{N}|$ output nodes to obtain the probability distribution, only about $|\log_2(\mathcal{N})|$ nodes need to be evaluated. Due to space limit, we omit the detailed derivatives.

Algorithm 1: The learning algorithm for JRLM++.

Input: User set $\mathcal{U}$, location set $\mathcal{L}$, link set $\mathcal{E}$, check-in records $\mathcal{C}$, K_0 and K_1
Output: User embeddings $\{\boldsymbol{v}_u\}$, segment embeddings $\{\boldsymbol{v}_s\}$, location embeddings $\{\boldsymbol{u}_l\}$
Initialize the parameters randomly;
repeat
  for *each user* $u \in \mathcal{U}$ **do**
    for *each segment* $s \in \mathcal{C}^{(u)}$ **do**
      for *each check-in location* $\ell \in s$ **do**
        Maximize the objective function L_{loc} in Eq. 6 and then update $\boldsymbol{v}_u$, $\boldsymbol{v}_s$ and $\boldsymbol{v}_\ell$.
      end
      Maximize the objective function L_{seg} in Eq. 8 and then update $\boldsymbol{v}_u$ and $\boldsymbol{v}_s$.
    end
    Maximize the objective function L_{graph} in Eq. 2 and then update $\boldsymbol{v}_u$.
  end
until convergence;

The computational complexity for an iteration is $\mathcal{O}(N_0 \times C_0 + N_1 \times C_1 + |\mathcal{U}| \times C_2)$, where N_0 and N_1 denote the count of triplets (u, s, ℓ) and pairs (u, s) in our dataset respectively, $C_0 = (2K_0 + 2 + |\log_2(\mathcal{N})|) \times D$ denotes the cost of updating each $\boldsymbol{v}_u$, $\boldsymbol{v}_s$ and $\boldsymbol{v}_\ell$ in a triplet (u, s, ℓ) by maximizing objective function L_{loc} in Eq. 6, $C_1 = (2K_1 + 1 + |\log_2(\mathcal{N})|) \times D$ denotes the cost of updating each $\boldsymbol{v}_u$ and $\boldsymbol{v}_s$ in a pair (u, s) by maximizing objective function L_{seg} in Eq. 8, and $C_2 = \bar{F} \times (1 + |\log_2(|\mathcal{N}|) \times D$ denotes the cost of updating $\boldsymbol{v}_u$ for a user u by maximizing objective function L_{graph} in Eq. 2, where $\bar{F}$ is the average number of friends per user and D is the number of embedding dimensions.

5 Experiment

We use two public LBSN datasets in this work, namely *Foursquare* [Yin *et al.*, 2013] and *Gowalla* [Noulas *et al.*, 2012]. Both datasets contain check-in records and social connections among users. The detailed statistics of the two datasets are shown in Table 1.

Table 1: Statistics of our datasets.

Dataset	# Users	# Check-ins	# Links	# Locations
Foursquare	4,163	483,814	32,512	121,142
Gowalla	216,734	12,846,151	736,778	1,421,262

Since our model can provide both user and location representations, it will be straightforward to apply to multiple LBSN tasks. Here, we select two important and different applications for evaluation, namely *location recommendation* and *social link prediction*.

5.1 Evaluation on Location Recommendation

Experimental Setting. We first evaluate the effectiveness on location recommendation. Given a user, we split the corresponding check-in records into training and test sets with a ratio of 9:1. The check-in records are sorted ascendingly by their timestamps. Following [Yin *et al.*, 2013], each location (called a *ground truth location*) in test set is pair with 50 nearest un-visited locations (called *negative locations*). Given a user, we combine all her ground truth locations with the negative locations, and form a candidate list after randomly shuffling. A recommender system will rank the candidate list and produce a top-k recommendation list. To measure the quality of such a top-k list, we borrow the idea in the evaluation of Information Retrieval. We consider a ground truth location as *relevant* and non-visited locations as *non-relevant*. We adopt the commonly used Precision, Recall, and F1 as the evaluation metrics. Here, k is set to 10.

Methods to Compare. For location recommendation, we compare the proposed JRLM and JRLM++ models with the following methods.

- *SG* [Mikolov *et al.*, 2013a]: This is the standard *word2vec* model with the Skip-Gram architecture. A location is considered as a word, and check-in records from a user are ascendingly sorted by their timestamps.
- *GloVe* [Pennington *et al.*, 2014]: Similar to SG, this is another word embedding method, which uses both global and local contexts.
- *PVDM* [Le and Mikolov, 2014]: It is the Paragraph2Vector with Distributed Memory Model. A location is considered as a word and a set of visited locations from a user is considered as a document.
- *HRM* [Wang *et al.*, 2015]: It characterizes users' general taste along with the sequential check-in records behaviors in the hierarchical representation. It represents the state-of-the-art for next-basket recommendation. We slightly modify it to adapt to location recommendation.
- *USG* [Ye *et al.*, 2011]: It combines user based collaborative filtering and friendship based collaborative filtering to compute the ranking probability for all candidates locations.

For all the representation learning methods (including PVDM, HRM, JRLM and JRLM++), we can obtain corresponding user (*a.k.a.* doc) and location embeddings. Given a user u, we rank a candidate location ℓ by its cosine similarity with the user embedding, *i.e.,* $\boldsymbol{v}_u{}^\top \cdot \boldsymbol{v}_\ell$. Since there are no user embeddings in both SG and GloVe, we average the embeddings of all the locations from a user. Further, we adopt the hierarchical softmax for optimization, the number of embedding dimensions and context window length (for both segment and location windows) are set to 300 and 8 respectively. The parameters in USG are optimized using 10% training data. Among these baselines, SG, GloVe, PVDM and HRM only consider sequential check-in records, USG use both check-in records and social connections, but does not characterize the sequential relatedness.

Although other methods [Bao *et al.*, 2015; Yin *et al.*, 2013; Yuan *et al.*, 2014] in LBSN literature can be used as baselines, they do not consider social connectedness or sequential relatedness. We select the classic USG from these methods as a baseline because it jointly considers both check-in records and social connections. The setting of USG is more similar to ours and it performs well as shown in [Ye *et al.*, 2011]. In another view, our approach aims to provide general data representations. It may not work best in a single application but overall give a good performance on multiple LBSN tasks.

Results and Analysis. Table 2 presents the recommendation performance of different methods on both datasets. We can observe: (1) Sequential embedding methods (*i.e.,* SG and GloVe) achieve similar performance without explicit user modeling. PVDM performs better than them since PVDM incorporates user preference (*i.e.,* doc embedding). (2) HRM performs better than PVDM because it integrates the user preference with previous location in last "baskets" as contexts in a ranking model. (3) HRM and JRLM perform better than USG. The major reason is that both methods can capture sequential relatedness, which is important to location recommendation. Besides, sequential embedding methods [Mikolov *et al.*, 2013a] are particularly suitable to model sequential relatedness. (4) By incorporating social connections, our base model JRLM improves over HRM, indicating the effectiveness of social connections. (5) Further, JRLM++ is substantially better than JRLM, which shows the effectiveness of multi-grain sequential contexts.

Table 2: Performance comparison on location recommendation. "♯" and "†" indicate the improvement over the best baseline is significant at the level of 0.01 and 0.05 respectively.

Methods	Foursquare			Gowalla		
	P	R	F1	P	R	F1
SG	0.398	0.205	0.256	0.389	0.216	0.264
GloVe	0.383	0.195	0.245	0.388	0.215	0.262
PVDM	0.408	0.207	0.261	0.399	0.220	0.269
USG	0.414	0.209	0.266	0.406	0.219	0.271
HRM	0.432	0.215	0.274	0.427	0.228	0.282
JRLM	**0.442**†	**0.226**†	**0.283**†	**0.433**†	**0.237**†	**0.289**†
JRLM++	**0.481**♯	**0.246**♯	**0.308**♯	**0.468**♯	**0.260**♯	**0.318**♯

5.2 Evaluation on Social Link Prediction

Experimental Setting. The task aims to predict whether there exists a social link between a pair of users or not. As indicated in [Pham *et al.*, 2013; Ye *et al.*, 2011], locations that are frequently visited by highly socially-related persons tend to be clustered together, which indicates the close correlation between social strength and mobile patterns. Hence, we aim to derive effective user representations for link prediction using both social connections and check-in records. A social connection corresponds to a pair of friends. We split all the friend pairs into training set and test set with a ratio of 0.9.

The friend pairs are considered as positive instances, we randomly select non-linked user pairs with an equal number. As a classic binary-classification setting [Pham *et al.*, 2013], we adopt the Precision, Recall and F1 as the evaluation metrics.

Methods to Compare. For link prediction, we compare our proposed models with the following methods.

- *EBM* [Pham *et al.*, 2013]: It calculates the diversity of co-occurrences for each user pair and the weighted location entropy for each co-visited location. As shown in [Pham *et al.*, 2013], EBM performs very well for link prediction in LBSNs, serving as a competitive baseline.
- *RTM* [Chang and Blei, 2009]: A relational topical model which characterizes the document content and the links between them. Similar to PVDM, the set of visited locations by a user is considered as a doc. It can jointly model both social connections and social connections. The order of check-in locations are ignored in RTM.
- *DeepWalk* [Perozzi *et al.*, 2014]: It learns user representations from a social graph, and employs a method based on sampling multiple paths consisting of user vertices.
- *LINE* [Tang *et al.*, 2015b]: It tries to preserve both the local and global network structures in learning the network embedding vectors. LINE and DeepWalk represent the state-of-art for network embedding using only network information.

For DeepWalk, LINE and our models, the number of embedding dimensions is set to 300, and the context window lengths (for both segment and location windows) are set to 8. For RTM, the number of topics is set to 300. For EBM, we set the order of diversity parameter q to 0.1 as suggested in [Pham *et al.*, 2013]. Given a user u, each method can generate a user representation (denoted by $\mathbf{x}^u$), *e.g.,* user embedding (DeepWalk, LINE and our models) and topical distribution (RTM). Given a pair of users u and v, we can derive a feature vector $\mathbf{x}^{u,v} = \mathbf{x}^u \circ \mathbf{x}^v$, where "$\circ$" is the Hadamard product and the i-th dimension $x_i^{u,v}$ is set to the product between x_i^u and x_i^v. With such a feature representation for a user pair, we adopt the commonly used *Support Vector Machine (SVM)*[3] for classification with the label set $\{$LINK, NOT-LINK$\}$. A method with a better feature representation will lead to a higher classification performance.

Table 3: Performance comparison on link prediction. "♯" and "†" indicate the improvement over the best baseline is significant at the level of 0.01 and 0.05 respectively.

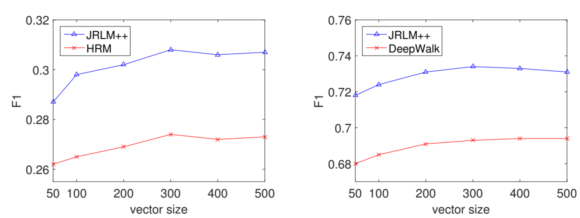
Methods	Foursquare			Gowalla		
	P	R	F1	P	R	F1
EBM	0.778	0.502	0.610	0.789	0.511	0.62
RTM	0.796	0.518	0.628	0.804	0.523	0.634
LINE	0.826	0.544	0.656	0.829	0.549	0.661
DeepWalk	0.849	0.585	0.693	0.851	0.588	0.695
JRLM	**0.878**†	**0.597**†	**0.711**†	**0.882**†	**0.606**†	**0.718**†
JRLM++	**0.891**♯	**0.624**♯	**0.734**♯	**0.905**♯	**0.632**♯	**0.744**♯

Results and Analysis. Table 3 presents the performance of link prediction on both datasets. First, previous network embedding models (*i.e.,* DeepWalk and LINE) perform better than the competitive baseline of EBM designed for LBSNs. Only using social connections, DeepWalk and LINE are even better than RTM, which uses both social connections and check-in records. Both methods directly characterize social connections using a distributed representation. In [Perozzi *et al.*, 2014; Tang *et al.*, 2015b], network embedding methods have been demonstrated to be more effective than the methods with careful feature engineering in multiple tasks. Our findings confirm this again in LBSN data. Further, our models JRLM and JRLM++ outperform all the baselines. The major reason is that we also adopt a distributed network embedding approach, which has the similar merit of DeepWalk and LINE. In addition, we also utilize both data types, which can better reflect the complex characteristics of LBSN data. The improvement of JRLM++ over JRLM is smaller than that in location recommendation. This might be because the task itself is user-oriented and the impact of multi-grain location contexts is less significant.

[3]https://www.csie.ntu.edu.tw/ cjlin/libsvm/#download

5.3 Parameter Sensitivity

We have two important parameters to tune in JRLM++: the context window lengths (*i.e.,* $2K_0$ and $2K_1$) and the number of embedding dimensions (*i.e.,* D). We only report the results on Foursquare dataset, and the results on Gowalla are similar. First, we have found that when $K_0 \geq 4$ and $K_1 \geq 4$, the performance becomes relatively stable to achieve the best. Next, we fix K_0 and K_1, and vary D from 50 to 500. We present the tuning results in Fig 2. It can observed that when $D \geq 300$, the performance becomes stable (or slightly worse on link prediction). In all cases, JRLM++ is consistently better than the selected baseline. Hence, we set D to 300. Overall, our model performance is relatively insensitive with the varying of these parameters in a large selection interval.



(a) Location Recommendation (b) Link Prediction

Figure 2: Varying the the number of embedding dimensions.

6 Conclusion

This paper presents a novel embedding model for LBSNs. It jointly models social connections and check-in sequences. To capture sequential relatedness, it has characterized two levels of sequential contexts, namely fine-grained and coarse-grained contexts. The proposed model can provide joint data representations in multiple LBSN tasks. As future work, we will consider extending the user embedding part to characterize more complicated structure patterns, *e.g.,* communities. In addition, we will try to integrate longitude and latitude information into our model. We will also study how to accelerate the learning procedure using parallel implementation.

References

[Bao *et al.*, 2015] Jie Bao, Yu Zheng, David Wilkie, and Mohamed F. Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.

[Chang and Blei, 2009] Jonathan Chang and David M Blei. Relational topic models for document networks. In *ICAIS*, pages 81–88, 2009.

[Cheng *et al.*, 2013] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, 2013.

[Feng *et al.*, 2015] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. Personalized ranking metric embedding for next new POI recommendation. In *IJCAI*, pages 2069–2075, 2015.

[Gao *et al.*, 2012] Huiji Gao, Jiliang Tang, and Huan Liu. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *CIKM*, pages 1582–1586. ACM, 2012.

[Le and Mikolov, 2014] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[Li *et al.*, 2008] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *SIGSPATIAL*, page 34. ACM, 2008.

[Lian *et al.*, 2014] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *SIGKDD*, pages 831–840. ACM, 2014.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Noulas *et al.*, 2012] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *SocialCom*, pages 144–153. IEEE, 2012.

[Pei *et al.*, 2001] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *icccn*, page 0215. IEEE, 2001.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, pages 701–710. ACM, 2014.

[Pham *et al.*, 2013] Huy Pham, Cyrus Shahabi, and Yan Liu. Ebm: an entropy-based model to infer social strength from spatiotemporal data. In *SIGMOD*, pages 265–276. ACM, 2013.

[Tang *et al.*, 2015a] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *SIGKDD*, pages 1165–1174. ACM, 2015.

[Tang *et al.*, 2015b] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[Wang *et al.*, 2011] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. Human mobility, social ties, and link prediction. In *SIGKDD*, pages 1100–1108. ACM, 2011.

[Wang *et al.*, 2015] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, pages 403–412. ACM, 2015.

[Yan *et al.*, 2003] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan: Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177. SIAM, 2003.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y. Chang. Network representation learning with rich text information. In *IJCAI*, pages 2111–2117, 2015.

[Ye *et al.*, 2009] Yang Ye, Yu Zheng, Yukun Chen, Jianhua Feng, and Xing Xie. Mining individual life pattern based on location history. In *MDM*, pages 1–10, 2009.

[Ye *et al.*, 2011] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334. ACM, 2011.

[Ye *et al.*, 2013] Jihang Ye, Zhe Zhu, and Hong Cheng. Whats your next move: User activity prediction in location-based social networks. In *SIAM*. SIAM, 2013.

[Yin *et al.*, 2013] Hongzhi Yin, Yizhou Sun, Bin Cui, Zhiting Hu, and Ling Chen. Lcars: a location-content-aware recommender system. In *SIGKDD*, pages 221–229. ACM, 2013.

[Yuan *et al.*, 2014] Quan Yuan, Gao Cong, and Aixin Sun. Graph-based point-of-interest recommendation with geographical and temporal influences. In *CIKM*, pages 659–668. ACM, 2014.

[Zheng *et al.*, 2010] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038. ACM, 2010.

[Zheng, 2015] Yu Zheng. Trajectory data mining: An overview. *ACM TIST*, 6(3):29, 2015.