Inferring Mobility Relationship via Graph Embedding

YANWEI YU*, Pennsylvania State University, USA
HONGJIAN WANG, Pennsylvania State University, USA
ZHENHUI LI, Pennsylvania State University, USA

Inferring social relationships from user location data has become increasingly important for real-world applications, such as recommendation, advertisement targeting, and transportation scheduling. Most existing mobility relationship measures are based on pairwise meeting frequency, that it, the more frequently two users meet (i.e., co-locate at the same time), the more likely that they are friends. However, such frequency-based methods suffer greatly from data sparsity challenge. Due to data collection limitation and bias in the real world (e.g., check-in data), the observed meeting events between two users might be very few. On the other hand, existing methods focus too much on the interactions between two users, but fail to incorporate the whole social network structure. For example, the relationship propagation is not well utilized in existing methods. In this paper, we propose to construct a user graph based on their spatial-temporal interactions and employ graph embedding technique to learn user representations from such a graph. The similarity measure of such representations can well describe mobility relationship and it is particularly useful to describe the similarity for user pairs with low or even zero meeting frequency. Furthermore, we introduce semantic information on meeting events by using point-of-interest (POI) categorical information. Additionally, when part of the social graph is available as friendship ground truth, we can easily encode such online social network information through a joint graph embedding. Experiments on two real-world datasets demonstrate the effectiveness of our proposed method.

CCS Concepts: • **Information systems** → **Data mining**; **Location based services**; **Social networks**; *Information extraction*; • **Human-centered computing** → *Mobile computing*;

Additional Key Words and Phrases: Data mining, mobility, relationship strength, spatiotemporal, social computing

ACM Reference Format:
Yanwei Yu, Hongjian Wang, and Zhenhui Li. 2018. Inferring Mobility Relationship via Graph Embedding. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 147 (September 2018), 21 pages. https://doi.org/10.1145/3264957

1 INTRODUCTION

Nowadays, human mobility data can be collected from a variety of sources including location-sharing social networks (e.g., Foursquare check-ins), geo-tagged social media (e.g., Twitter and Flickr), location-based online services (e.g., Uber and Yelp), and other smartphone applications. Different from online data, these spatial-temporal data provide us with another dimension of human behaviors in the physical space. Understanding the mobility data could benefit a broad range of applications in business, transportation, health, urban planning, and many more. In particular, several studies [7, 12, 29, 35, 36] have been conducted to discover social relationship based

*This is the corresponding author.

Authors' addresses: Yanwei Yu, Pennsylvania State University, 201 Old Main, University Park, PA, 16802, USA, yuy174@ist.psu.edu; Hongjian Wang, Pennsylvania State University, 201 Old Main, University Park, PA, 16802, USA, hxw186@ist.psu.edu; Zhenhui Li, Pennsylvania State University, 201 Old Main, University Park, PA, 16802, USA, jessieli@ist.psu.edu.

2474-9567/2018/9-ART147 $15.00
https://doi.org/10.1145/3264957

on people's interactions on the space. Discovering social relationships between users has become increasingly important for real-world applications ranging from friend recommendation, link prediction, advertisement targeting to transportation scheduling. For example, if two users frequently co-locate, it may imply that they have a strong relationship or similar interest, and we can recommend them to become online friends or suggest them with the restaurants that the other user frequently visits.

In this paper, we are interested in measuring the relationship strength between two users from the mobility data. In literature, a baseline approach is to use meeting frequency as the relationship measure [7, 8, 12]. Basically, the more frequently that two users co-locate, the stronger their relationship is. However, such a measure fails to consider *when* and *where* the meeting events (or co-locating events) occur. For example, two users co-locating in a place that is consistently visited by a large population may not necessarily know each other (i.e., the meeting event is just a coincidence), whereas meeting in an area which is seldom checked-in could indicate a strong relationship. To this end, recent studies further consider various factors based on meeting frequency measure including personal background (i.e., how frequently a person visits a location), global background (i.e., the popularity of a location), and temporal factor (i.e., the time difference in meeting events) [15, 29, 36].

Although extensive research has been done to refine the mobility relationship measure, the existing methods still have three key limitations. First, all existing measures are based on meeting frequency and do not work well for inactive users. Due to data collection mechanism, the collected mobility data might be quite sparse. For example, check-in data only collect user locations when they voluntarily share their location information. Such sparse data result in low-frequency meeting events. For example, in Gowalla dataset used in our experiment, among all user pairs with non-zero meeting frequencies, 84.92% of them meet only once. State-of-the-art methods [15, 29, 30, 36] discard the pairs with meeting frequency less than 2, which means they discard a significant portion of user pairs in experimental evaluation. How to deal with such extreme sparsity and still be able to measure the relationship strength for inactive users is a critical challenge in mobility relationship inference.

Second, existing methods measuring mobility relationship only focus on pairwise relationships. That is, each user pair is being independently modeled. However, in real world scenarios the relationship can also be inferred through relationship propagation. For example, if user A and user B have a strong relationship, and B and C have a strong relationship, we can infer that user A and C could also have a strong relationship, even if we do not observe any meeting events between them.

Lastly, none of existing work uses external data to semantically understand the meeting events. Although [5], [29] and [36] consider the popularity of locations and personal temporal patterns (i.e., global background, personal background, and temporal factor), all the factors are computed based on meeting frequency. Rich external contexts (e.g., venue information from FourSquare or event information from geo-tagged tweets) provide us with additional semantics for meeting events. For example, two users meeting at a residential place are more likely to have a stronger relationship compared with two users meeting in a public train station.

In this paper, we propose an effective relationship measure by considering both contextual information and relationship propagation. First, we use a novel graph embedding framework to model relationship propagation. In this graph, each vertex is a user and the weight of an edge is the meeting frequency between two users. By learning the embedding from such a graph, the relationship propagation is implicitly implemented and the data sparsity issue is also addressed because the relationships for low-frequency user pairs could be inferred through other users. However, we realize that simply applying the existing embedding method will actually hurt the overall performance due to a large number of noisy co-locating events. To address the issue, we propose a hierarchical sampling technique to improve the performance of mobility relationship inference. Second, to integrate the contextual information, we adjust the weight of each edge based on the POI category information of the co-locating venues. Furthermore, our approach can be easily extended to a semi-supervised setting if some of the social network information is given. In such a setting, we jointly train on two user graphs – the mobility graph and the partial social graph.

We are not the first to use graph embedding to infer mobility relationship strength. However, the existing graph embedding-based mobility relationship inference methods have their own weakness. For example, the most recent work [1] organizes users and locations into a bipartite graph, the edges between users and locations representing the users visiting the locations with weight indicating the visiting frequency, and applies a group embedding that employs random walk on the bipartite graph to infer social links. However, the method ignores the important time information in the user-location bipartite graph. For example, two users both frequently visit a public place (e.g. Walmart) at different time, but they may not be friends. Two friends may meet at a place where another user frequently visits, but both of they may not know the user. Moreover, a large number of edges are connected between users and their visited locations due to lack of time constrain, which builds the indirect connections between inactive users and active users. This actually introduces noise for relationship inference of both inactive users and active users.

We conduct extensive experiments on Gowalla and Brightkite datasets [6] to verify the effectiveness of our method. While previous methods often pick the active users for evaluation, we emphasize that our method is particularly effective for low-frequency user pairs. This is important as the inactive users dominate the dataset – the data follow a long-tail distribution (e.g., 96% user pairs have meeting frequency less than three in the Gowalla dataset).

To summarize, we make the following contributions:

- We propose a novel method emb to measure the relationship strength from mobility data using user graph embedding. Our method incorporates users' spatial-temporal interaction and relationship propagation.
- We design a hierarchical sampling strategy to better select contextual neighbors in user representation learning. The new sampling strategy explores user interactions at different levels to enhance meaningful relationship propagation and reduce noises in the user graph.
- We incorporate semantic contextual information into graph embedding by adjusting edge weights based on the category information of the co-locating venues.
- We further extend our graph embedding to a semi-supervised setting by incorporating partial social graph.
- We conduct extensive evaluations on two real-world datasets. Experimental results demonstrate that our method outperforms the state-of-the-art methods.

The rest of paper is organized as follows. Related work is discussed in Section 2. We present our problem definition in Section 3 and the proposed graph embedding method in Section 4. Experimental results are reported in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

A spatial trajectory is generally a sequence of timestamped locations. There have been many studies on how to measure the similarity between two trajectories, such as Dynamic Time Warping [39], Longest Common Subsequence [34], Edit distance with Real Penalty [3], and Edit Distance on Real sequence [4]. These trajectory measures are designed to determine the similarity between trajectories continuously moving in the free space, such as the positions of body joint collected from sensors or hurricane trajectories. *But they are not suitable for measuring the human movement trajectories because human movements are highly constrained by the space (e.g., road network) and contain many non-moving points.*

Several methods have been proposed to measure the similarity of human movement trajectories [21, 37, 40, 43]. In [21], the authors propose a hierarchical graph to model users' location history and measure the similarity among users by similar location sequences. Zheng et al. [43] propose a similar hierarchical graph to model multiple users' location histories, and then mine travel sequences from the graph model. In [37], user's GPS trajectories are first converted to semantic location history. The similarity between different users is then measured by summarizing the weighted similarity of semantic location sequences. Zhao et al. [40] propose a probabilistic generative model

to mine latent lifestyle-related patterns from human trajectory and then learn social strength from the mined lifestyle pattern features. All these methods first transform a trajectory into a sequence of semantic locations, e.g., "mall → restaurant → cinema". The similarity between two trajectories is then measured on such symbolic sequences. *Such measures find people with similar transition patterns, but do not require people to co-locate at the same place at the same time.*

Co-locating frequency (or meeting frequency) has been widely used in trajectory data mining. In particular, various methods have been proposed to detect moving object clusters that frequently co-locate, such as *moving cluster* [17], *convoy* [16], *swarm* [22], *gathering* [41, 42] and *evolving group* [19]. Kalnis et al. [17] propose one notion of moving cluster, which is a set of objects when clustered at consecutive time points, and the portion of common objects in any two consecutive clusters is not below a predefined threshold. A *convoy* pattern in [16] is defined as a set of objects that move together (always falling into the same density-based cluster) during at least k consecutive time points. *Swarm* pattern [22] is a variation of convoy pattern. It allows the moving objects to leave the group temporally. Tang et al. [33] propose the problem of discovering travelling companions in the context of streaming trajectories. The notion of travelling companion is essential the same as *convoy*. *However, these methods simply use the frequency as a distance measure without considering the semantics of meeting events.*

There is a variety of research that focuses on the relationships between geographical locations and social links[2, 10, 18, 27]. Lambiotte et al. [18] show the probability that two users are connected by a communication link follows a gravity model that decreases as the negative square of distance between the users. Onnela et al. [27] find that small social groups are geographically very tight but become much more clumped when the group size exceeds a certain number of members. Domenico et al. [10] verify that it is possible to exploit the correlation of social interactions and user movement to improve the prediction accuracy of the future location. Brown et al. [2] study the differences between individual and group co-location behavior with respect to physical location. They discover that individuals are more likely to meet with one friend at a new place but tend to meet with a large group at familiar locations. *Such studies discover certain correlations between physical location and social ties, but they do not try to measure the mobility relationship based on user movements.* In literature, several work attempt to mine social relationships based on latent structure, e.g., Dong et al. [11] use a Markov jump process to model the co-evolution of behaviors and social relationships, and Nguyen et al. [25, 26] apply Hierarchical Dirichlet Processes to infer social groups. *However, these modeling methods require lots of continuous location samples, which does not apply to sparse data for inactive users.*

Recently, a line of research work focuses on more refined measures for mobility relationship based on the meeting events in the physical world. This line of research is the most relevant to ours. Studies [12, 23] show that the meeting time could indicate different types of relationships, e.g., colleagues meeting during the daytime vs. friends meeting at night. Cranshaw et al. [8] propose to extract a set of features from both the meeting events and the individual mobility patterns and learn a supervised model to classify friendship from the check-in data. Recently, Pham et al. [29] propose an entropy-based model (EBM) to consider the diversity of meeting locations to handle cases where two subjects could meet by coincidence. In a follow-up work, they extend EBM by further considering the location semantics and stay duration [30]. Wang et al. [36] propose a unified relationship measure PGT that considers global background (similar to the location entropy in EBM), personal background (i.e., the probability of a user to visit a certain location), and temporal factor (i.e., the time difference between consecutive meeting events). Hsieh et al. [15] first construct a co-location graph based on the personal factor, collective factor (also similar to the location entropy), or temporal factor between users separately, and then compute graph features (e.g. Jaccard) to measure relationships for user pairs. Cheng et al. [5] design two friendship predictors based on logistic regression model using the co-occurrence events of users in location-based social network. *However, all of these studies consider pairwise relationships independently.*

Most recently, Backes et al. [1] first apply graph embedding into social relationship inference. Specifically, they organize users and locations into a bipartite graph, the edges between users and locations representing the users visiting the locations with weight indicating the visiting frequency, and employs random walk based embedding on the bipartite graph to infer social links. *However, the method ignores the important time information in the user-location bipartite graph, which results in a mass of redundancy edges connected users and their visited locations. This actually causes lots of noise for relationship inference of both inactive users and active users. Also, none of existing work utilizes external contextual data to measure the relationship strength.*

3 PROBLEM DEFINITION

We first define the key data structures and notations used in the paper. Let $U = \{u_1, u_2, \ldots, u_m\}$ denote the set of all users, we define a check-in record as follows:

Definition 1 (Check-in record). *A check-in record is a triple $\langle u, t, \ell \rangle$ that represents user u visiting location ℓ at time t, where ℓ denotes a place with the geographical coordinates (i.e., longitude and latitude).*

Definition 2 (Trajectory). *The trajectory of a user u is a sequence $(\langle u, t_1, \ell_1 \rangle, \langle u, t_2, \ell_2 \rangle, \ldots, \langle u, t_n, \ell_n \rangle)$ of all check-in records made by user u where $t_i < t_{i+1}$, for all $1 \leq i < n$. We denote it as Tr_u.*

To understand the semantics of the user mobility data, in this paper we use the external point of interest (POIs) information to annotate the check-in records. We next define the POI and the semantic check-in record as follows:

Definition 3 (POI). *A POI is a uniquely identified venue in the form of $\langle p, name, category, \ell \rangle$, where p is the POI identifier, name is the name of the POI, category denotes its category, and ℓ represents the geographical location of the POI (i.e., longitude and latitude).*

Definition 4 (Semantic check-in record). *A semantic check-in record is a triple $\langle u, t, p \rangle$ that represents user u visiting POI p at time t.*

In this paper, a semantic check-in record is obtained by associating a check-in record $\langle u, t, \ell \rangle$ with the closest POI $\langle p, name, category, \ell' \rangle$, subject to the condition that the distance between ℓ and ℓ' is less than certain threshold ε_d. In real-world applications, different types of location-aware devices with varying positioning accuracy may be used. To obtain meaningful semantic check-in records, we fix $\varepsilon_d = 20$ meters in this paper. Specifically, if the distance between the check-in location and its closest POI is less than 20 meters, we keep this semantic check-in record. Note that other methods for annotating POI with mobility data could be plugged in and would be orthogonal to our method.

Definition 5 (Semantic trajectory). *The semantic trajectory of a user u is a sequence of all semantic check-in records $(\langle u, t_1, p_1 \rangle, \langle u, t_2, p_2 \rangle, \ldots, \langle u, t_n, p_n \rangle)$ made by user u, where $t_i < t_{i+1}, \forall\ 1 \leq i < n$. We denote it as STr_u.*

Finally, we formally define our problem as follows:

Problem (Mobility Relationship Strength Inference). *Given a set of users $U = \{u_1, u_2, \ldots, u_m\}$ and their semantic trajectories, we wish to infer the mobility relationship strength score $S(u_i, u_j) \in [0, 1]$ for each pair of users.*

The mobility relationship strength between two mobile users is commonly learned from their spatial-temporal interactions. In the literature, the interaction behavior, i.e., a *meeting event*, is usually defined as follows:

Definition 6 (Meeting event). *Given two users u and u' and their semantic trajectories, and a time threshold τ, we say that u and u' have a meeting event if $\exists \langle u, t, p \rangle \in STr_u, \langle u', t', p' \rangle \in STr_{u'}$ such that $p = p'$ and $|t - t'| \leq \tau$.*

One simple but intuitive measure for relationship strength is the *meeting frequency*, which is frequently used in the literature [7, 8, 12, 15, 29, 30, 36]. Specifically, let $M = \{m_1, m_2, \ldots\}$ denote the set of all meeting events between users u_i and u_j. Then the meeting frequency of user pair (u_i, u_j) is the cardinality of M, i.e. $|M|$.

A key limitation of the meeting frequency-based method and its extensions is that they consider each user pair independently. Hence, these methods perform poorly on user pairs with few meeting events. To address this problem, our goal is to infer mobility relationship strength between two users by taking account of relationship propagation and further incorporating external contextual data.

4 USER MOBILITY INTERACTION GRAPH

In this section, we first introduce our user mobility interaction graph (or simply user graph) to model the meeting events among all users. Then we introduce a graph embedding framework for relationship strength inference. Finally, we describe how to integrate external semantic information and partial labels into embedding learning.

Definition 7 (User Graph). *A User Graph is defined as an undirected graph $\mathcal{G} = (U, E)$, where U is the set of vertices, each representing a user, and E is the set of edges between the vertices. Each edge $e_{ij} \in E$ represents the relationship between users u_i and u_j and is associated with a weight $w_{ij} > 0$, which indicates their interaction behavior (i.e., the meeting frequency).*

The user graph is designed to capture all the meeting events among users. Inspired by recent graph embedding techniques [14, 28, 32], we propose to learn embeddings from the user graph to estimate the user similarity. The graph embedding can capture not only the explicit meeting information, but also the implicit similarity propagation among users – two users could be similar through other common friends even if these two users have low meeting frequency. To measure the relationship strength between two users, we use the cosine similarity between the corresponding embedding vectors.

4.1 Preliminary: User Graph Embedding Learning

Given a user graph $\mathcal{G} = (U, E)$, we let $\Phi : U \rightarrow \mathbb{R}^d$ be the mapping function from users to vector representations, where d is the dimension of vector representation. For a user $u \in U$, we define set $\mathcal{N}(u) \subset U$ as a local context of vertex u. One example of the local context could be all the one-hop neighboring vertices of u in the user graph $\mathcal{G}$.

We adopt the Skip-gram language model [14, 24, 28] to calculate user graph embedding, that is, use the nodes in the network to predict the probability of their context. The learning objective is to maximize the log-probability of observing the context $\mathcal{N}$ for each user u conditioned on its vector representation Φ:

$$\begin{aligned} O &= \max \sum_{u \in U} \log P(\mathcal{N}(u)|u) \\ &= \max \sum_{u \in U} \log \prod_{v \in \mathcal{N}(u)} P(v|u) \\ &= \max \sum_{u \in U} \sum_{v \in \mathcal{N}(u)} \log P(v|u). \end{aligned} \tag{1}$$

To make the optimization problem tractable, we approximate the conditional probability $P(\mathcal{N}(u)|u)$ using the standard conditional independence assumption in Eq. (1).

The probability $P(v|u)$ is estimated by the embeddings of vertices through

$$P(v|u) = \sigma(\Phi(u)^T \Psi(v)), \tag{2}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function, and $\Psi(v) \in \mathbb{R}^d$ is an auxiliary vector of v when v is treated as "context" of other vertices. Namely, the conditional probability $P(v|u)$ is correlated to the similarity of representation vector of u and auxiliary vector of its context v.

In order to speed up the training and improve the quality of the vector representation, the negative sampling technique [24] is used. Specifically, we sample negative users $NEG(u) = \{z|z \notin \mathcal{N}(u)\}$ with respect to each user

u, and we try to minimize the probability $P(z|u)$ for each user $z \in NEG(u)$. This is equivalent to maximize the probability $1 - P(z|u)$, i.e., the objective of negative sampling follows:

$$O_{neg} = \max \sum_{u \in U} \sum_{z \in NEG(u)} \log(1 - P(z|u)). \tag{3}$$

Combine Eq. (1) and Eq. (3), our optimization problem becomes:

$$\begin{aligned} O &= \max \sum_{u \in U} \sum_{v \in \mathcal{N}(u)} \left\{ \log P(v|u) + \sum_{z \in NEG(u)} \log(1 - P(z|u)) \right\} \\ &= \max \sum_{u \in U} \sum_{v \in \mathcal{N}(u)} \left\{ \log \left[\sigma(\Phi(u)^T \Psi(v))\right] + \sum_{z \in NEG(u)} \log \left[1 - \sigma(\Phi(u)^T \Psi(z))\right] \right\}. \end{aligned} \tag{4}$$

That is, we aim to maximize the probability of each node in $\mathcal{N}(u)$ being a neighbor of vertex u and minimize the probability of each node in $NEG(u)$ being a neighbor of vertex u in the embedding space.

Given $\mathcal{N}(u)$ for each user, we can solve Eq. (4) using asynchronous stochastic gradient descent (ASDG) [31] over the two parameters Φ and Ψ. So the remaining issue is how to choose the local context $\mathcal{N}(u)$, which we discuss in next section.

4.2 Local Context Selection

The choice of local context $\mathcal{N}(u)$ plays a critical role in the representation learning on any graph. In this paper, we first adopt a second-order random walk [14] to obtain $\mathcal{N}(u)$. Then, we extend it with a hierarchical framework to discover more meaningful neighbors in the mobility data.

4.2.1 Second-Order Random Walks. In embedding learning, random walk is an efficient method to sample the local context of vertices on graph $\mathcal{G}$. However, the search for the original random walk is restricted to the neighborhood of each node, and is unable to take latent local communities and the roles of each node in the communities into account. To address this issue, we adopt the second-order random walk procedure proposed in [14] to guide a biased neighborhood sampling. Consider a random walk that just stopped by node t and now arrives at node v. Now the walk need to decide on the next stop by setting biased weights on edges e_{vx} with two parameters. In particular, they use a in-out parameter q to control the random walk's preference towards Breadth-first Sampling (BFS) or Depth-first Sampling (DFS) by scaling the weights of the second-order edges. On other hand, they use a return parameter p on edge e_{tv} to encourage the walk to backtrack a step by setting parameter $p < 1$ or to avoid it by setting $p > 1$.

As validated in [14], the performance of embedding improves as the parameters p and q decrease on social networks. This is because q with a low value encourages outward exploration in neighborhood searching, at the same time it is balanced by a low-valued p which ensures that the walk does not go too far from the start node. Therefore, we choose the best values for p and q studied in [14] in the subsequent experiments.

4.2.2 Hierarchical Sampling. Although the second-order random walk already considers the edges' weights in sampling procedure, the sampled neighbors in a walk for user u are regarded as equivalent. However, in real world scenarios the user graph is always noisy, especially for the low-weight edges. These edges often correspond to coincidences (e.g., two strangers happen to co-locate at a public place) rather than real meeting events. On the other hand, existing works [29, 36] have shown that user pairs with high meeting frequencies are much more likely to be friends. These phenomenons pose following two challenges in the random walk-based sampling schemes:

First, while the active users frequently meet with their friends, they may also have many co-locating events with non-friends by accident. For an inactive user, if he/she happens to connect to an active users by low-weight

edges, the walks would easily move towards the connected active users and get trapped in the communities of the active users, resulting in unreasonable similarity between the inactive user and the active user's communities.

Second, the second-order random walk may not be able to differentiate the meeting events during relationship propagation. For example, in a network $\{u_1, u_2, u_3\}$ with $w_{u_1u_2} = 10$ and $w_{u_2u_3} = 10$, one has high confidence to infer that there is strong relationship between u_1 and u_3, even if they do not meet each other. In contrast, in another network $\{v_1, v_2, v_3\}$ with $w_{v_1v_2} = 1$ and $w_{v_2v_3} = 1$, one has less confidence to infer the relationship between v_1 and v_3.

To filter the noisy edges and strengthen the relationship of frequent meeting pairs, we design a hierarchical sampling scheme based on second-order random walk. Specifically, consider a given minimum edge weight min_w, we omit the edges whose weights are less than min_w, and then perform second-order random walk sampling on the remaining graph. Note that the isolated nodes are also eliminated in sampling process. In this way, we eliminate many accidental meeting events and further highlight the dependences of the user pairs that meet frequently. By gradually changing the parameter min_w, we are implementing a hierarchical sampling scheme by learning embedding at different frequency levels. The multiple gradually incremental edge weights used in the hierarchical sampling process form the minimum weight set, denoted by Min_w.

Fig. 1 shows an example of a user graph including 12 users (nodes). Black edges indicate that the edges with weights between 1 and 2, blue edges have weights between 3 and 4, and red edges have weights are larger than 4. Given a minimum weight set $Min_w = \{0, 3, 5\}$, we first perform second-order random walks on the entire graph when $min_w = 0$. Then, for $min_w = 3$, we omit the edges whose weights are less than 3, and do the walks on the remaining graph, i.e. sampling the nodes connected by blue and red edges. Finally, we repeat the step for $min_w = 5$, namely, performing random walks on the sub-graphs with a higher meeting frequency level, i.e., sub-graphs connected by the red edges in Fig. 1.

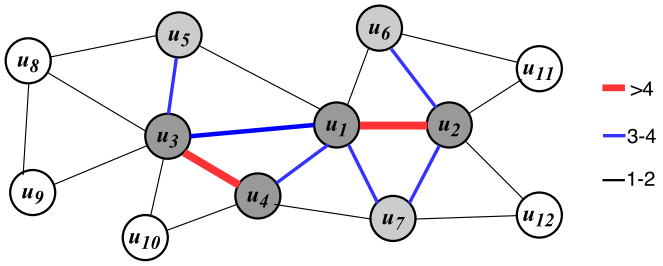


Fig. 1. Our hierarchical sampling strategy.

Using random walks on the entire graph alone would pull up the similarities between active users' communities with their non-friend neighbors in embedding space. By strengthening the relationship of frequent meeting pairs gradually at different frequency levels in our hierarchical sample, the user pairs with higher meeting frequency would be more similar in embedding space, which resolves the challenge to differentiate meeting events during relationship propagation. On the another hand, the similarities between active users and their non-friend inactive neighbors are weakened equivalently by removing the connected edges in high level sample. The similarities between more active users and their non-friend inactive neighbors may be reduced more, because the relationship between frequent user pairs may be strengthened multiple times in hierarchical sampling. In other words, the similarities for inactive user pairs with low weighted edges in embedding space can be highlighted gradually.

4.2.3 The emb Algorithm. The pseudo-code for our user graph embedding (denoted by emb) is given in Algorithm 1. First, we use the hierarchical sampling strategy to generate the biased random walks on different levels of meeting frequency (lines 1-8). In practice, we sample W_n random walks of fixed length W_l for every source node

in user graph (lines 2-7). Note that, for a non-zero min_w, we simulate the random walks on the sub-graphs rather than all nodes, where U_{min_w} denotes the filtered nodes which have at least one edge satisfying the minimum weight min_w (line 4). We use the alias table method [20] to draw a node sample in the random walk procedure, which only takes $O(1)$ time.

Second, we use negative sampling to implement the Skip-gram language model in accordance with our objective function Eq. (4) for each user u. To optimize Eq. (4), we employ the asynchronous stochastic gradient descent algorithm (ASGD) proposed in [31] (lines 3-7 in Algorithm 2). Specifically, for each context v_j, we sample a set of negative users for u_i, denoted $NEG(v_j)$ instead of $NEG(u_i)$ (lines 3-4). The learning rate η for ASGD is initially set to a starting value and then decreased linearly with the number of vertices that have been trained so far.

Algorithm 1 Graph Embedding Algorithm

Input: User Graph $\mathcal{G} = (U, E)$, dimension d, window size w, walks per user W_n, walk length W_l, and minimum edge weight set Min_w.
Output: Matrix of representation vectors Φ

1: **for** each $min_w \in Min_w$ **do**
2: **for** $i = 0$ to W_n **do**
3: **for** each user u *in* U_{min_w} **do**
4: $walk \leftarrow$ 2ndOrderWalk$(\mathcal{G}, u, W_l, min_w)$;
5: $Walks \leftarrow Walks \cup \{walk\}$;
6: **end for**
7: **end for**
8: **end for**
9: **for** each $walk \in Walks$ **do**
10: $SkipGramNeg(\Phi, \Psi, walk, w)$
11: **end for**

Algorithm 2 SkipGramNeg(Φ, Ψ, $walk$, w)

1: **for** each $u_i \in walk$ **do**
2: **for** each $v_j \in walk[i - w : i + w]$ **do**
3: $O = \sum_{z \in \{v_j\} \cup NEG(v_j)} \log Pr(z|u_i)$;
4: **for** each $z \in \{v_j\} \cup NEG(v_j)$ **do**
5: $\Psi(z) = \Psi(z) + \eta \frac{\partial O}{\partial \Psi(z)}$
6: **end for**
7: $\Phi(u_i) = \Phi(u_i) + \eta \frac{\partial O}{\partial \Phi(u_i)}$;
8: **end for**
9: **end for**

4.3 Incorporating Semantics and Partial Labels

Our method provides a flexible framework that can easily incorporate additional information. Next, we will introduce two kinds of data to enhance the relationship inference.

4.3.1 User Graph with Semantics. First, we incorporate the POI data to differentiate various meeting events by their semantics. Here, we use the POI's categorical information to annotate the check-in records, in order to characterize the type of meeting place. We collect POIs from FourSquare via its public API [13]. The crawled POI dataset contains specific place, category, popularity and GPS coordinates. We mainly use the major category information to annotate the check-in records, in order to characterize the type of meeting place between users. There are 10 major categories, as shown in first column of Table 1.

Take Austin city as an example, we crawl 27,247 POIs from FourSquare API. By matching POIs with check-in data from Gowalla dataset, we obtain 207,278 semantic check-in records for 7,355 users. Further, we generate 176,083 pairs of users that have at least one meeting event according to Definition 6 under the setting of $\tau = 0.5$ hour. In Table 1, we show the probability of a meeting event being generated by a friend pair for each POI category. As we can see that, two users meeting in a professional venue only has a probability of 0.0279 to be a friend pair. This is much lower compared with meeting in a residence venue with 0.4509 probability to be a friend pair.

Table 1. Statistics of meeting events w.r.t. POI categories.

POI category	# meetings	# friends	ratio
Professional	126,944	3,541	0.0279
Shops	20,573	2,661	0.1293
Arts & Entertainment	9,923	792	0.0798
Outdoors & Recreation	7,869	542	0.0689
Nightlife	15,356	1,846	0.1202
Travel	20,101	1,643	0.0817
Food	16,192	3,960	0.2445
College & Education	954	198	0.2075
Residence	173	78	0.4509
Event	13	6	0.4615

Given the probability distribution of each major category, a weight is computed for each meeting event based on the category of its corresponding POI. Then, we use the weighted meeting frequency as the weight w_{ij} of edge e_{ij} in the user graph instead of meeting frequency and apply graph embedding for relationship propagation. In this way, we combine the meeting frequency and the semantic context of the meeting events into the representations of the users via a graph-based embedding learning approach.

One may note that, to calculate such category weights, we need knowledge about the friendships. However, we argue that such weights can be treated as a general statistics. The numbers remain constant regardless of the training data we use. We have conducted an experiment by using only 10% meeting events to obtain such weights and repeat the random sampling of 10% data for 100 times. Such 10% data are removed in method evaluation. We get very similar ratio values as the last column in Table 1. For example, the variance of ratio for professional category is only 0.0013 on Gowalla dataset. We have conducted similar experiments on relationship inference and the results by using different 10% training datasets have at most 0.001 in difference (measured in PRAUC), which suggests that the category weights can be treated as a general statistics.

4.3.2 Learning with Partial Labels. Second, we incorporate available online social network information to learn the embedding.

Our method can also be easily extended to a semi-supervised setting if some of the social network information is given. We denote the given partial social graph as $\mathcal{G}_s = (U_s, E_s)$, where U_s is a subset of U, and $e_{ij} \in E_s$ is a binary edge indicating the friendship.

For each pair of users u_i and u_j in $\mathcal{G}_s$, we want their embeddings $\Phi(u_i)$ and $\Phi(u_j)$ to comply with the structure of $\mathcal{G}_s$ as well. Following the objective in Eq. (4), we have

$$O_s = \sum_{u \in U_s} \sum_{v \in \mathcal{N}_{\mathcal{G}_s}(u)} \sum_{z \in \{v\} \cup NEG_{\mathcal{G}_s}(u)} \log Pr(z|u), \tag{5}$$

where $\mathcal{N}_{\mathcal{G}_s}(u)$ denotes the neighborhood of user u in social graph $\mathcal{G}_s$, and $NEG_{\mathcal{G}_s}(u)$ denotes negative sampling for u in social graph $\mathcal{G}_s$. The social graph $\mathcal{G}_s$ complements the co-location user graph $\mathcal{G}$ and guides the learning of the vector representations of users. To encode both graphs in our embeddings, the final objective is:

$$O_j = O + O_s. \tag{6}$$

To learn the parameter set $\{\Phi, \Psi\}$ in Eq. (6), we again generate second-order random walks in social graph $\mathcal{G}_s$ and then use ASGD to solve the optimization problem.

5 EXPERIMENTS

5.1 Data Description

We use two real datasets containing user check-ins from Gowalla and Brightkite[1] [6], which are two location-based social network services. In Gowalla dataset, the friendship network consists of 196,591 nodes and 950,327 edges, and a total of 6,442,890 check-in records of these users are collected over the period of Feb. 2009 - Oct. 2010. Brightkite dataset consists of 58,228 nodes and 214,078 edges in the friendship network, and has 4,491,143 check-in records over the period of Apr. 2008 - Oct. 2010. The format of one check-in is as follows: ⟨user ID, time, latitude, longitude, location ID⟩. The social network of friendships servers as the ground truth in our evaluation.

We focus on three cities with most check-ins in our experiments: Austin in Texas for Gowalla dataset, San Francisco (SF) and Los Angeles (LA) in California for Brightkite dataset. The reason that we select these three cities is that our method makes use of the venue information which is not available in the datasets themselves. Instead, we collected such information using FourSquare API [13]. Foursquare API has a query limit (up to 500 requests per hour), so it is not feasible to collect POIs for all the cities. For each check-in record in the datasets, we match it with the closest POI collected from FourSquare. Table 2 shows the basic statistics of our datasets after pre-processing.

Table 2. Data statistics.

Dataset	Top Cities	# POIs	# users	# check-ins
Gowalla	Austin	27,247	7,355	207,278
Brightkite	SF and LA	261,973	6,393	223,549

In Fig. 2, we plot the sorted number of check-ins for both datasets. It is clear that the number of check-ins follows a long-tail distribution. Specifically, the number of users with more than 50 check-ins is only 960 for Gowalla and 757 for Brightkite.

Furthermore, the distribution of pairwise meeting frequency over all user pairs with non-zero meeting frequencies is also highly skewed as shown in Table 3. For example, 84.92% of user pairs have meeting frequency equal to 1 and 11.07% user pairs have meeting frequency equal to 2 on Gowalla. Additionally, as shown in Table 4, more than 70% of friendships are implied in the pairs whose meeting frequencies are less than 3 for both datasets. In our experiments, we evaluate the methods on all the user pairs with non-zero meeting frequency.

[1] Available at http://snap.stanford.edu/data/

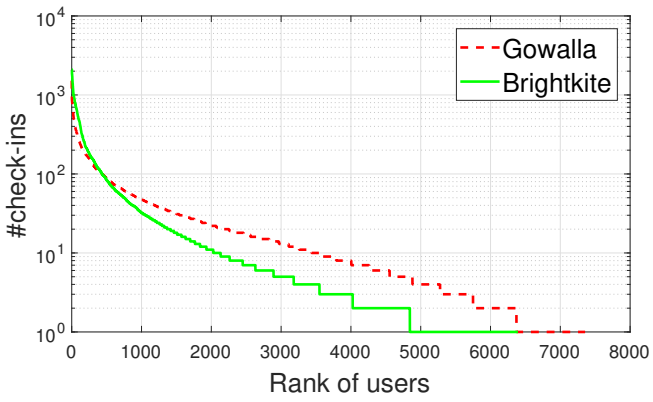


Fig. 2. Statistics of check-in records.

It is worth noting that previous work [15, 29, 30, 36] all discard the pairs with meeting frequency less than 2 in their experiments. In other words, they discard 84.92% of the user pairs on Gowalla and 72.13% user pairs on Brightkite in the experiments. These frequency-1 pairs are actually the challenging cases in real-world scenarios.

Table 3. Meeting frequency statistics.

Dataset	frequency≥1	frequency=1	frequency=2	frequency>2
Gowalla	176,083	149,528 (84.92%)	19,487 (11.07%)	7,068 (4.01%)
Brightkite	33,605	24,238 (72.13%)	4,952 (14.74%)	4,415 (13.14%)

Table 4. Friendship statistics.

Dataset	# friends	frequency = 1	frequency = 2	frequency > 2
Gowalla	5,548	3,119 (56.22%)	1,013 (18.26%)	1,416 (25.52%)
Brightkite	1,639	841 (51.32%)	310 (18.91%)	488 (29.77%)

5.2 Evaluation Settings

We compare our method with the following methods:

- freq. This is the baseline method that uses meeting frequency as the relationship measure.
- freq-cat. freq-cat uses meeting frequency weighted by the category of POIs as relationship measure.
- EBM (entropy-based method). EBM computes a weighted frequency by considering the location diversity using entropy [29].

- PGT. PGT is a state-of-the-art method which further extends EBM by considering the personal factor and temporal factor [36].
- node2vec. Graph embedding uses the meeting frequency as the edge weight and directly applies the state-of-the-art embedding method node2vec [14].
- node2vec-cat. The baseline applies the node2vec embedding method on user graph and also incorporates the POI category information.
- DeepWalk. This approach learns low-dimensional feature representations for each user in the user graph by simulating truncated random walks [28]. Note that the approach only supports networks with binary edges.
- walk2friends. walk2friends [1] is the state-of-the-art method which applies random walk based graph embedding on a user-location bipartite graph to infer social links.

Our embedding method has three variations:

- emb. Graph embedding method which uses the meeting frequency as the weight of the edge.
- emb-cat. emb-cat applies our proposed graph embedding method on user graph with additional POI category information.
- emb-cat-social. This is the joint embedding method in a semi-supervised setting that incorporates both social graph information and POI semantics into emb.

For each mobility relationship measure, we obtain a ranked list of pairs with higher scores indicating higher likelihood to be friends. We use the online friendship as the ground truth for evaluation. We use precision, recall, precision-recall curve and *the area under the precision-recall curve (PRAUC)* to quantify the results. The precision and recall are defined as follows: $Precision(Q) = \frac{|G \cap Q|}{|Q|}$ and $Recall(Q) = \frac{|G \cap Q|}{|G|}$, where G denotes the set of ground truth friend pairs in the dataset, and Q denotes the set of friend pairs reported by the method under a particular experiment setting. Note that our PRAUC is based on the precision-recall curve that is different from "AUC (Area Under roc Curve)". This is because precision-recall curves yield better precision in evaluating the performance of link prediction with class imbalance, while roc curve and AUC can be deceptive in evaluation [9, 38]. As shown in Table 3 and Table 4, the high ratio of negative instances (non-friend pairs) to positive instances (friend pairs) occurs in both Gowalla and Brightkite datasets, which exhibits extreme class imbalance.

We define meeting events as two users visiting same POI within 30 minutes. For our method, we set the embedding dimension $d = 128$ by default and minimum weight set $Min_w = \{0, 3, 7, 15\}$. Similar to [28, 32], the starting value of learning rate η is set to 0.025. As studied in [14], we set $p = 0.5$ and $q = 0.25$ for second-order random walk. emb-cat selects the same number of pairs at each level in hierarchical sampling as emb. We randomly selects 10% meeting events to calculate the friendship ratio for each venue category in emb-cat and freq-cat, and those 10% data are removed from the method evaluation. For semi-supervised learning, the sampled social graph is also removed from the method evaluation.

5.3 Overall Performance

We first evaluate the overall performance on both Gowalla and Brightkite datasets. Fig. 3 and Fig. 4 show the PRAUC values of all methods in each dataset on the three cases: frequency = 1, frequency $\leq$ 2 and all pairs. Fig. 3(a) and Fig. 3(b) show variations of our method compared with baseline freq, Fig. 3(c) and Fig. 3(d) show the comparison between our method and recent methods PGT and EBM, Fig. 4(a) and Fig. 4(b) show the comparison between our emb and emb-cat with the state-of-the-art embedding method node2vec and node2vec-cat, and Fig. 4(c) and Fig. 4(d) show the comparison between our method with DeepWalk and the state-of-art social link inference method walk2friends. In addition, Fig. 5 further shows the precision-recall curves of all methods on all pairs for both Gowalla and Brightkite datasets. Based on these results, we make the following observations:

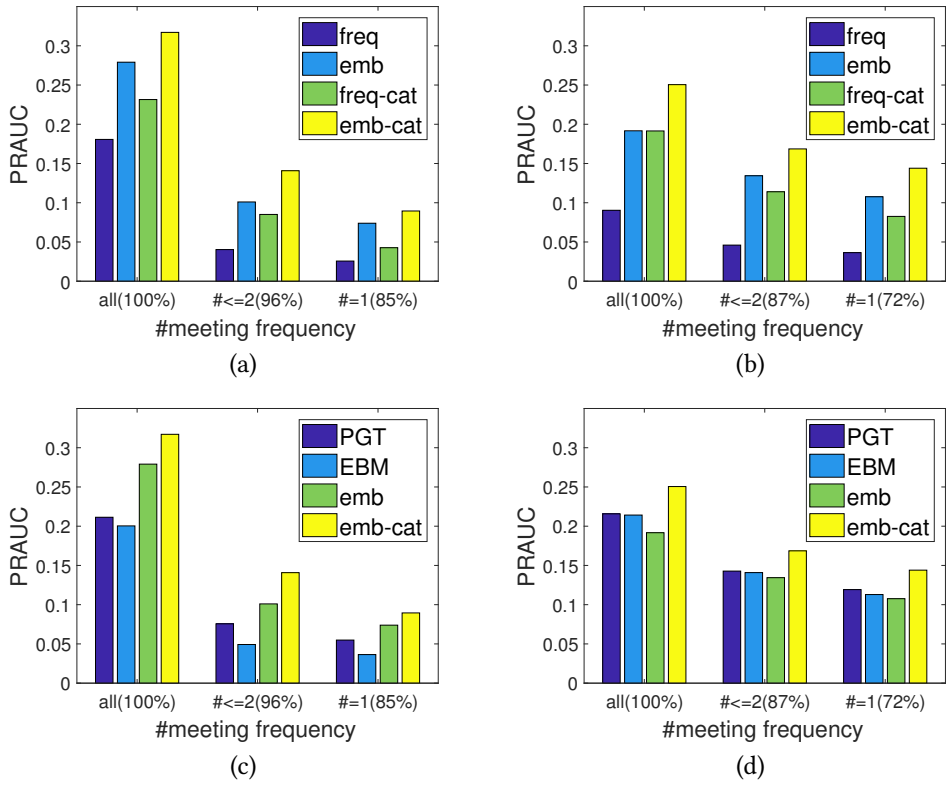


Fig. 3. Overall performance comparison on Gowalla (**first column**) and Brightkite (**second column**) datasets. **(a) and (b):** Variation of frequency-based baselines. **(c) and (d):** Comparison to state-of-the-art relationship inference methods.

(1) Embedding methods outperform frequency-based methods. Although meeting frequency is an important indicator of relationships, from Fig. 3(a) and Fig. 3(b), we can see that embedding methods consistently outperform the corresponding frequency methods, i.e., emb outperforms freq and emb-cat outperforms freq-cat. Note that it is much more difficult to predict the relationships for low meeting frequency pairs (i.e., frequency = 1 and frequency $\leq$ 2) using the meeting frequency alone. The embedding methods are particularly effective in such cases because they enable relationship propagation.

(2) Semantic category information helps relationship inference. The methods using category information consistently outperform the methods without such information, i.e., freq-cat outperforms freq and emb-cat outperforms emb, as shown in Fig. 3(a) and Fig. 3(b). This result indicates that utilizing external semantic venue information can help differentiate the meeting events.

(3) Our proposed method outperforms the state-of-the-art pairwise relationship inference methods. As shown in Fig. 3(c) and Fig. 3(d), emb-cat outperforms both EBM and PGT. Even though EBM and PGT consider various factors to weight meeting events, none of these methods utilize relationship propagation and semantic venue information.

(4) Hierarchical sampling improves inference performance. As shown in Fig. 4, the proposed embedding method outperforms DeepWalk and node2vec on all cases (emb vs. node2vec and emb-cat vs. node2vec-cat in Fig. 4(a) and

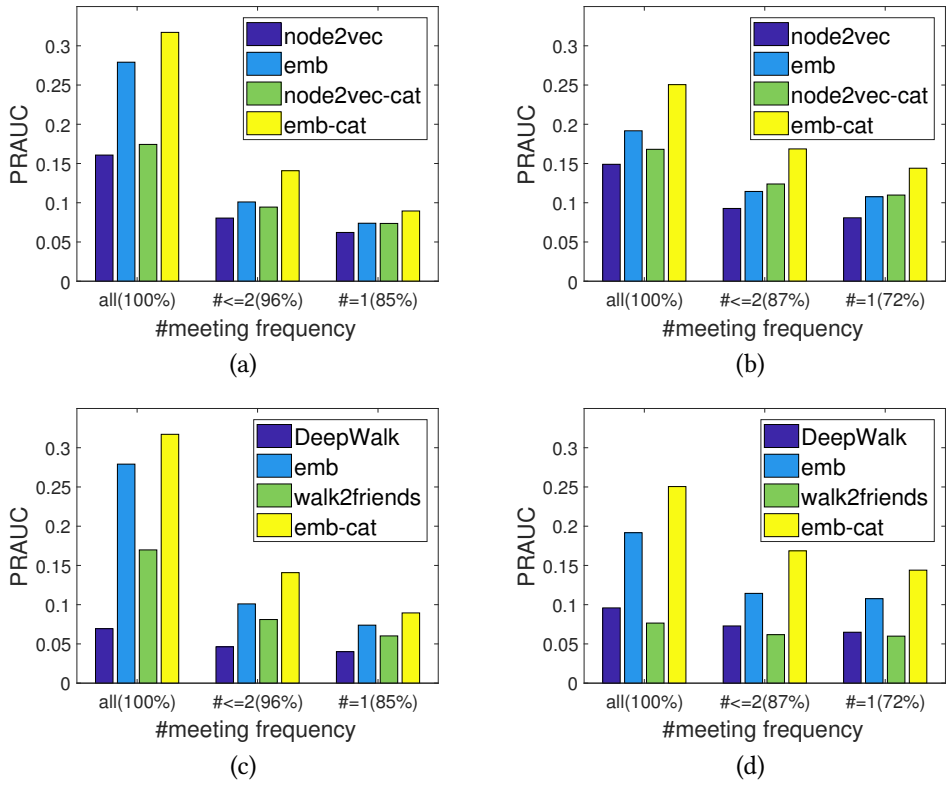
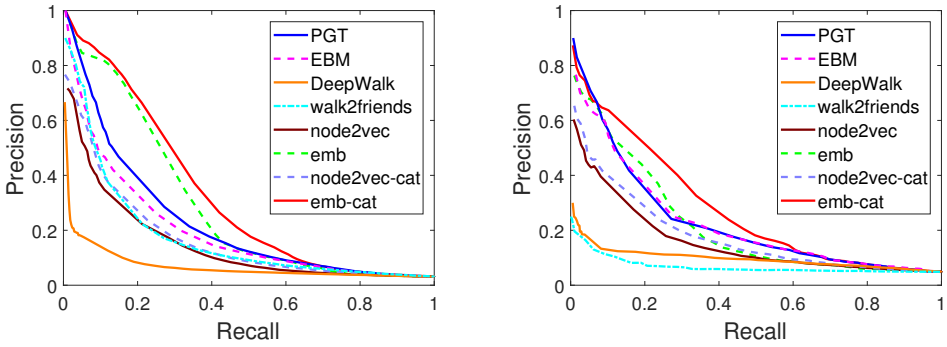


Fig. 4. Overall performance comparison on Gowalla (**first column**) and Brightkite (**second column**) datasets. **(a) and (b):** Comparison to node2vec and node2vec-cat methods. **(c) and (d):** Comparison to DeepWalk and walk2friends.

Fig. 4(b), emb vs. DeepWalk in Fig. 4(c) and Fig. 4(d)). The reason is that our emb and emb-cat use the hierarchical sampling to implement relationship propagation at different levels. Our method can strengthen the relationship of frequent meeting pairs and simultaneously suppress noises in the meeting events. Fig. 5 also demonstrates emb and emb-cat outperform embedding baselines in precision-recall curves. DeepWalk has the worst performance, because it only supports binary edges.

(5) Our proposed method outperforms the state-of-the-art graph embedding based inference methods. As we can see, in Fig. 4(c) and Fig. 4(d), our method significantly outperforms the state-of-the-art walk2friends on both datasets. The reason is that our method constructs the user graph based on meeting events that incorporate users' spatial-temporal interaction, while walk2friends ignores the important time constrains in its user-location bipartite graph, resulting much more noisy. Specifically, walk2friends is even worse than DeepWalk on Brightkite dataset, this is because there are more than 260 thousand locations in Brightkite, which causes the noisy edges in the user-location graph being far more than those in our user graph. In addition, our method further uses the hierarchical sampling to improve inference performance.

(6) The relative performance varies among different user pairs. The performance of all methods increases from frequency = 1 to frequency $\leq$ 2 and to all pairs. This is because for user pairs with higher meeting frequency, the

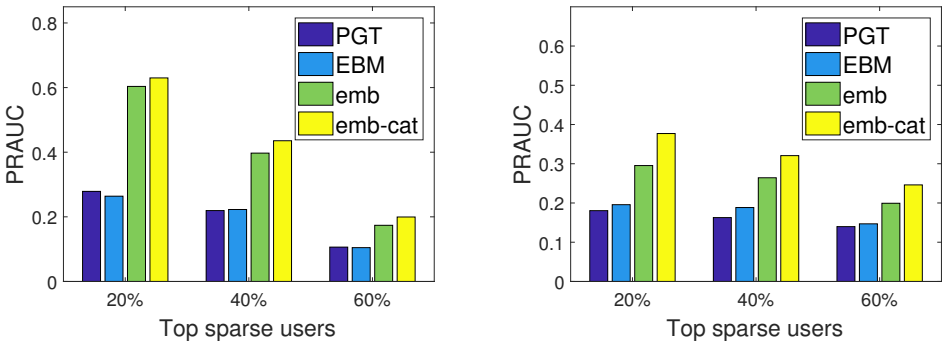


(a) Gowalla (b) Brightkite

Fig. 5. Precision-recall curve on all pairs.

ratio of friendship pairs increases. With the higher friendship ratio, the retrieval task becomes easier, leading to higher PRAUC scores.

5.4 Performance w.r.t Data Sparsity



(a) Gowalla (b) Brightkite

Fig. 6. Performance w.r.t user sparsity.

Next we demonstrate that our method is particularly useful when the data is sparse. Recall that the check-in data follow a long-tail distribution, and the majority of the users have very few check-ins (Fig. 2). The ability to handle data sparsity is therefore of great importance in practice.

The performance of our method w.r.t. user check-ins on two datasets is shown in Fig. 6. We rank the users based on their numbers of check-ins in an ascending order and select the top-k% users (i.e., users with the least check-ins, or top sparse users). Note that all methods are performed on whole user graph and only evaluated on top sparse users in this experiment. As we can see, our method is consistently better than state-of-the-art methods with such sparse users. The top-20% sparse users are the most difficult cases, where most of them have only one check-in and almost all pairs meet only one time. PGT performs nearly the same as EBM in this case because

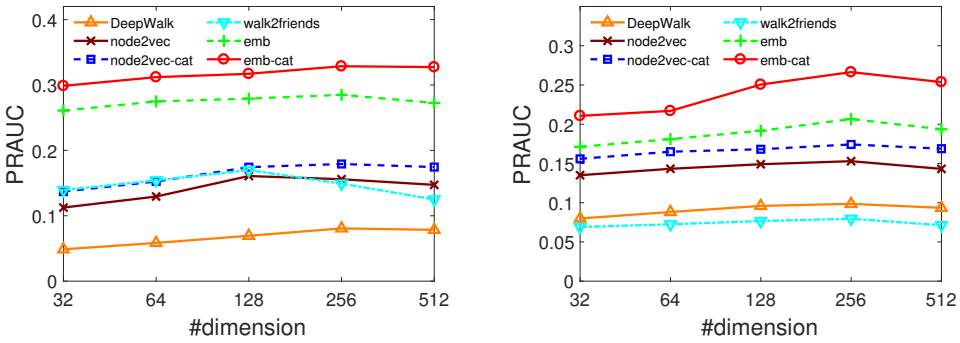
personal factor and temper factor are no longer useful. However, our methods are much more effective for such challenging cases. In particular, our emb method improves by 118% in PRAUC compared to the state-of-the-art PGT for the top-20% sparse users on Gowalla dataset. In addition, emb-cat can further outperform emb by using extra venue information.

We also observe that our method performs much better on top-20% sparse users than on all data (Fig. 6 vs. Fig. 3). This is mainly because the percentage of friend pairs is different. The friendship ratio of top-20% sparse users is higher than that of all user pairs. For example, in Gowalla, the friendship ratio is 17.97% for top-20% sparse users vs. 3.15% for all users. With a higher friendship ratio, the prediction job tends to be easier and that is why we observe higher PRAUC for all the methods on the sparse users. Nevertheless, it might still be a little counter-intuitive that friendship ratio of sparse users is even higher than that of all users. This is because we only use the user pairs that meet at least 1 time (suppose there are N such pairs) for the purpose to evaluate the frequency-based methods. Among these N pairs of users, suppose there are M friend pairs. In Gowalla, for top-20% sparse users, M/N=39/217=17.97%, for all the users, M/N=5548/176083=3.15%. Therefore, top-20% sparse users actually have a higher friendship ratio due to the filtering of user pairs with zero meeting frequencies. However, PGT and EBM do not improve the performance of inference due to the very low meeting frequency among top sparse users, while our method significantly improves the performance on top-20% sparse users.

The results again demonstrate that user propagation through graph embedding learning and the external contextual information are helpful for mobility relationship inference, especially in the case of sparse user data.

5.5 Performance w.r.t Embedding Dimension

We now investigate the performance of our method w.r.t. the embedding dimension compared with embedding-based baselines by varying the number of dimension from 32 to 512.



(a) Gowalla (b) Brightkite

Fig. 7. Performance w.r.t. the dimension d.

The comparison results on two datasets are shown in Fig. 7. As we see, our method consistently outperforms all embedding-based baselines in PRAUC on all tested cases for both datasets. Specifically, our emb-cat method improves by an average of 171% in PRAUC compared to the state-of-the-art walk2friends on two datasets. The reason is same as explained above. As expected, the performance of each method first increases slightly as embedding dimension increases, and then drops when the dimension becomes too large. But in general, the performance of each method is quite consistent with respect to the dimension. And the relative performance between methods is relatively stable across different values of the embedding dimension. The experiment also demonstrates the robustness of our method within a large range of embedding dimension.

5.6 Evaluation on Semi-Supervised Learning

Table 5. Semi-supervised learning by using a small portion of the social graph (Gowalla dataset).

Case	% social pairs	emb-cat	emb-cat-social	% improve
# = 1	0	0.08952	0.08952	0
# = 1	10%	0.08475	0.11032	34.21%
# = 1	20%	0.08057	0.14793	83.60%
# = 1	30%	0.07336	0.19063	159.86%
# ≤ 2	0	0.14081	0.14081	0
# ≤ 2	10%	0.13421	0.14751	9.91%
# ≤ 2	20%	0.12813	0.18705	45.98%
# ≤ 2	30%	0.12003	0.23117	92.59%
all pairs	0	0.30714	0.30714	0
all pairs	10%	0.28493	0.31229	9.60%
all pairs	20%	0.27917	0.32084	14.93%
all pairs	30%	0.26654	0.33548	25.86%

Table 6. Semi-supervised learning by using a small portion of the social graph (Brightkite dataset).

Case	% social pairs	emb-cat	emb-cat-social	% improve
# = 1	0	0.14412	0.14412	0
# = 1	10%	0.13923	0.14985	7.19%
# = 1	20%	0.11294	0.16358	44.84%
# = 1	30%	0.10151	0.19946	96.49%
# ≤ 2	0	0.16851	0.16851	0
# ≤ 2	10%	0.16053	0.16909	5.33%
# ≤ 2	20%	0.15222	0.19023	24.97%
# ≤ 2	30%	0.14483	0.22767	57.20%
all pairs	0	0.25056	0.25056	0
all pairs	10%	0.24975	0.25385	1.64%
all pairs	20%	0.24527	0.27179	10.80%
all pairs	30%	0.23192	0.31294	34.93%

As described in Section 4.3.2, our embedding method can naturally be extended to incorporate social network information, if available. In this experiment, we evaluate the proposed joint embedding method in a semi-supervised setting, where a small portion of social network information (i.e., ground truth friendship) is available. We sample a subgraph from the social network as training data. More specifically, the subgraph consists of 10% to 30% of edges from the complete social graph. We use the remaining unknown graph edges as the testing samples. We compare the joint embedding method emb-cat-social with emb-cat, which does not utilize the social graph.

The comparison results are shown in Table 5 for Gowalla dataset and Table 6 for Brightkite dataset on the three cases: frequency = 1, frequency ≤ 2 and all pairs. emb-cat-social is superior to emb-cat consistently on all cases, especially on the challenging sparse cases. The PRAUC of emb-cat is decreasing as the number of training friend

pairs increase, because the number of positive friend pairs remained in the testing set is decreasing. The PRAUC of emb-cat-social increases dramatically as the training set increases. The reason is that as more information of the social network is given, it is easier to infer the rest of the graph structure.

6 CONCLUSION

The user check-in data in the physical world correlate with the users' relationship strength. However, the commonly-used meeting frequency-based methods for relationship inference often suffer from the data sparsity issue. In this paper, we propose an embedding learning method on the user graph to address this issue. An embedding method has the advantage to account for the relationship propagation, while a frequency-based method considers the pairwise relationships independently. To better capture the user interaction in the graph and to reduce noises, we develop a hierarchical sampling strategy with second-order random walks to select the neighborhood for each user. We further propose to encode the external venue information of meeting venues in the user graph. Finally, when part of the social graph information is available, we can easily encode such information through joint graph embedding. Extensive evaluations on two real world datasets show that our embedding method consistently outperforms the state-of-the-art methods. It is worthy to mention that our method is particularly effective for user pairs with low meeting frequencies compared to most existing methods.

We shall mention that our method does not account for the personal, global, and temporal factors proposed in [29, 36]. As future work, we plan to model all these factors in a heterogeneous user graph, so that the embedding method can incorporate these factors as well.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is partially supported by the National Natural Science Foundation of China under Grant No.: 61773331 and 61403328, the National Science Foundation under Grant No.: 1544455, 1652525 and 1618448, and the China Scholarship Council under Grant No.: 201608370018. Zhenhui Li would like to acknowledge the support from Haile Family Early Career Professorship. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. 2017. walk2friends: Inferring Social Links from Mobility Profiles. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1943–1957.

[2] Chloë Brown, Neal Lathia, Cecilia Mascolo, Anastasios Noulas, and Vincent Blondel. 2014. Group colocation behavior in technological social networks. *PloS one* 9, 8 (2014), e105816.

[3] Lei Chen and Raymond Ng. 2004. On the marriage of lp-norms and edit distance. In *Proceedings of the VLDB Endowment*. VLDB Endowment, 792–803.

[4] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD*. ACM, 491–502.

[5] Ran Cheng, Jun Pang, and Yang Zhang. 2015. Inferring friendship from check-in data of location-based social networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ACM, 1284–1291.

[6] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1082–1090.

[7] David J Crandall, Lars Backstrom, Dan Cosley, Siddharth Suri, Daniel Huttenlocher, and Jon Kleinberg. 2010. Inferring social ties from geographic coincidences. *PNAS* 107, 52 (2010), 22436–22441.

[8] Justin Cranshaw, Eran Toch, Jason Hong, Aniket Kittur, and Norman Sadeh. 2010. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 119–128.

[9] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 233–240.

[10] Manlio De Domenico, Antonio Lima, and Mirco Musolesi. 2013. Interdependence and predictability of human mobility and social interactions. *Pervasive and Mobile Computing* 9, 6 (2013), 798–807.

[11] Wen Dong, Bruno Lepri, and Alex Sandy Pentland. 2011. Modeling the co-evolution of behaviors and social relationships using mobile phone data. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 134–143.
[12] Nathan Eagle, Alex Sandy Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *PNAS* 106, 36 (2009), 15274–15278.
[13] Foursquare Venues Service. 2015. https://developer.foursquare.com/overview/venues.html. (2015).
[14] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.
[15] Hsun-Ping Hsieh, Rui Yan, and Cheng-Te Li. 2015. Where you go reveals who you know: analyzing social ties from millions of footprints. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1839–1842.
[16] Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S Jensen, and Heng Tao Shen. 2008. Discovery of convoys in trajectory databases. *Proceedings of the VLDB Endowment* 1, 1 (2008), 1068–1080.
[17] Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. 2005. On discovering moving clusters in spatio-temporal data. In *International Symposium on Spatial and Temporal Databases*. Springer, 364–381.
[18] Renaud Lambiotte, Vincent D Blondel, Cristobald De Kerchove, Etienne Huens, Christophe Prieur, Zbigniew Smoreda, and Paul Van Dooren. 2008. Geographical dispersal of mobile communication networks. *Physica A: Statistical Mechanics and its Applications* 387, 21 (2008), 5317–5325.
[19] Ruoshan Lan, Yanwei Yu, Lei Cao, Peng Song, and Yingjie Wang. 2017. Discovering Evolving Moving Object Groups from Massive-Scale Trajectory Streams. In *2017 18th IEEE International Conference on Mobile Data Management (MDM),*. IEEE, 256–265.
[20] Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 891–900.
[21] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. 2008. Mining user similarity based on location history. In *Proceedings of the 16th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 34.
[22] Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kays. 2010. Swarm: Mining relaxed temporal moving object clusters. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 723–734.
[23] Zhenhui Li, Cindy Xide Lin, Bolin Ding, and Jiawei Han. 2011. Mining significant time intervals for relationship detection. In *International Symposium on Spatial and Temporal Databases*. Springer, 386–403.
[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of Advances in neural information processing systems*. 3111–3119.
[25] Thuong Nguyen, Vu Nguyen, Flora D Salim, and Dinh Phung. 2016. SECC: Simultaneous extraction of context and community from pervasive signals. In *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*. IEEE, 1–9.
[26] Thuong Nguyen, Dinh Phung, Sunil Gupta, and Svetha Venkatesh. 2013. Extraction of latent patterns and contexts from social honest signals using hierarchical Dirichlet processes. In *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*. IEEE, 47–55.
[27] Jukka-Pekka Onnela, Samuel Arbesman, Marta C González, Albert-László Barabási, and Nicholas A Christakis. 2011. Geographic constraints on social network groups. *PLoS one* 6, 4 (2011), e16939.
[28] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.
[29] Huy Pham, Cyrus Shahabi, and Yan Liu. 2013. Ebm: an entropy-based model to infer social strength from spatiotemporal data. In *Proceedings of the 2013 ACM SIGMOD*. ACM, 265–276.
[30] Huy Pham, Cyrus Shahabi, and Yan Liu. 2016. Inferring social strength from spatiotemporal data. *ACM Transactions on Database Systems (TODS)* 41, 1 (2016), 7.
[31] Benjamin Recht, Christopher Ré, Stephen J. Wright, and Feng Niu. 2011. Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *Proceedings of Advances in neural information processing systems*. 693–701.
[32] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 1067–1077.
[33] Lu-An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Chih-Chieh Hung, and Wen-Chih Peng. 2012. On discovery of traveling companions from streaming trajectories. In *Proceedings of International Conference on Data Engineering*. IEEE, 186–197.
[34] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings of International Conference on Data Engineering*. IEEE, 673–684.
[35] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. 2011. Human mobility, social ties, and link prediction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1100–1108.
[36] Hongjian Wang, Zhenhui Li, and Wang-Chien Lee. 2014. PGT: Measuring mobility relationship using personal, global and temporal factors. In *Proceedings of 2014 IEEE International Conference on Data Mining (ICDM)*. IEEE, 570–579.
[37] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. 2010. Finding similar users using category-based location history. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 442–445.

[38] Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. 2015. Evaluating link prediction methods. *Knowledge and Information Systems* 45, 3 (2015), 751–782.
[39] Byoung-Kee Yi, HV Jagadish, and Christos Faloutsos. 1998. Efficient retrieval of similar time sequences under time warping. In *Proceedings of International Conference on Data Engineering*. IEEE, 201–208.
[40] Wayne Xin Zhao, Ningnan Zhou, Wenhui Zhang, Ji-Rong Wen, Shan Wang, and Edward Y Chang. 2016. A Probabilistic Lifestyle-Based Trajectory Model for Social Strength Inference from Human Trajectory Data. *ACM Transactions on Information Systems (TOIS)* 35, 1 (2016), 8.
[41] Kai Zheng, Yu Zheng, Nicholas Jing Yuan, and Shuo Shang. 2013. On discovery of gathering patterns from trajectories. In *Proceedings of International Conference on Data Engineering*. IEEE, 242–253.
[42] Kai Zheng, Yu Zheng, Nicholas J Yuan, Shuo Shang, and Xiaofang Zhou. 2014. Online discovery of gathering patterns over trajectories. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1974–1988.
[43] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 791–800.

Received November 2017; revised May 2018; accepted September 2018