

# Graph Convolutional Networks on User Mobility Heterogeneous Graphs for Social Relationship Inference

Yongji Wu, Defu Lian\*, Shuwei Jin and Enhong Chen

School of Computer Science and Technology, University of Science and Technology of China  
 {wuyongji317, dove.ustc}@gmail.com, jsw666@mail.ustc.edu.cn, cheneh@ustc.edu.cn

## Abstract

Inferring social relations from user trajectory data is of great value in real-world applications such as friend recommendation and ride-sharing. Most existing methods predict relationship based on a pairwise approach using some hand-crafted features or rely on a simple skip-gram based model to learn embeddings on graphs. Using hand-crafted features often fails to capture the complex dynamics in human social relations, while the graph embedding based methods only use random walks to propagate information and cannot incorporate external semantic data provided. We propose a novel model that utilizes Graph Convolutional Networks (GCNs) to learn user embeddings on the User Mobility Heterogeneous Graph in an unsupervised manner. This model is capable of propagating relation layer-wisely as well as combining both the rich structural information in the heterogeneous graph and predictive node features provided. Our method can also be extended to a semi-supervised setting if a part of the social network is available. The evaluation on three real-world datasets demonstrates that our method outperforms the state-of-the-art approaches.

## 1 Introduction

The widespread popularity of location-based social networks (LBSNs) such as Foursquare, location-based online services such as Uber has brought us a tremendous volume of human trajectory data. Understanding the underlying human mobility patterns proved to be of great value to various applications, e.g., POI recommendation, next visit-location prediction. In addition, social relations can be inferred from trajectory data since there are certain correlations between the two [Eagle *et al.*, 2009]. An intuitive thought is that two users who frequently meet are highly likely to be friends. Mining social ties from human mobility data is of vital importance since it provides us a new perspective on many real-world problems, such as friend recommendation, link prediction and transportation scheduling (e.g., ride-sharing).

Extensive research has been conducted on the social relationship inference using human mobility data. However, most existing works [Scellato *et al.*, 2011; Pham *et al.*, 2013; Pham *et al.*, 2016; Wang *et al.*, 2014] rely on simple hand-crafted features and predict pairwise relations individually, which does not take account of relation propagation. They fail to solve the data sparsity challenge of inactive users since almost all pairwise based methods require that two users share common locations in order to make predictions. Recently, a number of graph embedding based methods [Backes *et al.*, 2017; Yu *et al.*, 2018] have been proposed. They do not require hand-crafted features which may not accurately characterize various factors in human relations. However, instead of utilizing more powerful embedding approaches, these methods mainly use a simple skip-gram based model to learn node embeddings, which only computes the probability of a context extracted from random walks, and external semantic information cannot be well integrated. Besides, they are based on either a user meeting graph or a user-location bipartite graph, which may be combined to construct a heterogeneous graph that contains richer information for better performance.

Graph Convolutional Networks (GCNs) are developed in recent years [Defferrard *et al.*, 2016; Kipf and Welling, 2017] to enable convolutions on arbitrary structured graphs. The ability of GCNs to propagate information layer-wisely allows them to learn localized patterns at different scales, while the skip-gram model only exploits nodes co-occur in a random walk to minimize a carefully designed loss function. Moreover, GCNs can combine both the graph structure and external node features provided to learn predictive embeddings.

Due to the aforementioned advantages of GCNs and the rich information of a heterogeneous graph, we propose a novel method for unsupervised social relationship inference using human trajectories. Our method applies GCNs to learn user embeddings on the User Mobility Heterogeneous Graph, which incorporates user-user, user-location, location-location relations. To provide meaningful features for the GCN as input, inspired by [Gao *et al.*, 2017], we use a skip-gram based model to train location embeddings as features for location nodes, and trajectory embeddings as features for user nodes are extracted by training a RNN model to classify which user a trajectory belongs to. In addition, if a portion of the corresponding social network is known in advance, our model can also encompass it to obtain better performance.

\*Corresponding author

The major contributions of our work are three-fold:

1. To the best of our knowledge, this is the first model to infer social relations from human mobility data by mining a heterogeneous graph, which consists of user-user meeting graph, social graph, user-location bipartite graph and location-location co-occurrence graph.
2. We propose GCNs on heterogeneous graphs to integrate external semantic information (node features) and graph structure for user embedding learning in both unsupervised and semi-supervised way.
3. We conduct extensive experiments on three real-world datasets and obtain better performance than all compared baselines in the unsupervised setting, while the evaluation in the semi-supervised setting shows our model can achieve a tremendous performance boost given only a small portion of the ground truth social network.

## 2 Related Works

**Graph convolutional networks.** In recent years, great efforts have been made to extend traditional convolutional neural networks (CNNs) which operate on Euclidean structures to arbitrary graphs. Bruna et al. [2014] first proposed the convolution operation on graphs based on spectral graph theory, which is extended in [Defferrard et al., 2016] through using Chebyshev polynomials to approximate filters. Kipf and Welling [2017] further applied the first-order approximation to develop a layer-wise linear model, which can be used in fast and scalable semi-supervised node classification. This architecture is extended to operate on knowledge graphs in [Schlichtkrull et al., 2018]. An unsupervised inductive embedding learning framework is first introduced in [Hamilton et al., 2017], but is limited to homogeneous graphs.

**Mobility relationship inference.** There are a number of studies in using mobility data to infer social relations. Eagle et al. [2009] made the first attempt to discover the correlations between meeting events and social ties by using mobile phone data. Scellato et al. [2011] considered it as a link prediction problem and proposed a supervised learning framework which uses hand-crafted features to predict missing links. Pham et al. [2013] proposed an entropy-based model (EBM) that takes the diversity of meeting events and location entropy into consideration, which is further extended in [Pham et al., 2016] to incorporate location semantics. Wang et al. [2014] developed a unified framework that considers personal, global, and temporal factors. All methods above make predictions for each pair individually. Backes et al. [2017] first proposed a graph embedding based method on the user-location bipartite graph. Yu et al. [2018] were inspired by this mechanism and applied graph embedding on user meeting graph. Graph embedding is also used on the social network to predict missing links in [Zhou et al., 2018].

## 3 Graph Convolutional Networks for User Mobility Heterogeneous Graph

In this section, we elaborate our proposed method. First, we introduce the User Mobility Heterogeneous Graph. Second,

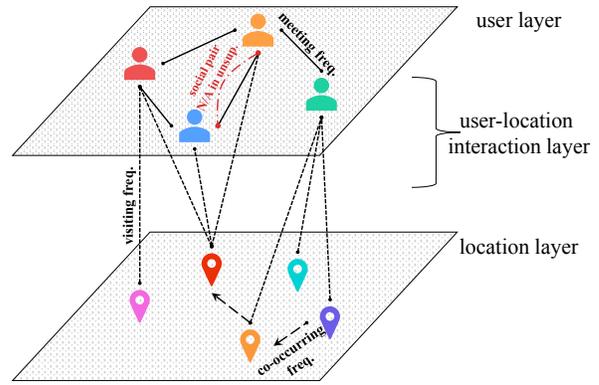


Figure 1: The User Mobility Heterogeneous Graph.

we present our GCN framework that operates on heterogeneous graphs to learn node representations unsupervisedly as well as the node features we feed to the GCN. Finally, we demonstrate that our method can also be used in a semi-supervised way by incorporating partial social networks. The workflow of our proposed method is illustrated in Figure 2.

### 3.1 Constructing User Mobility Heterogeneous Graph

In order to characterize user’s mobility profile and model relation propagation, we propose the User Mobility Heterogeneous Graph to incorporate various kinds of information that represent relations between user-user, user-location and location-location. The User Mobility Heterogeneous Graph can be divided into: user layer, location layer, and user-location interaction layer. We give an illustration in Figure 1.

Edges in the user layer connect user nodes. Each edge  $e_{i,j}$  represents the mobility relationship between user  $i$  and user  $j$  and its weight is defined by the meeting frequency between the two users. In the semi-supervised setting, we also connect an unweighted edge (as a second kind of relation in the user layer) for each social pair in the given partial social network. In user-location interaction layer, we connect an edge  $e_{u,l}$  from a user  $u$  to each of his check-in location  $l$ . The weight of the edge is the number of check-ins in this particular location (visiting frequency). For each edge  $e_{p,q}$  in the location layer, we connect the two locations  $p$  and  $q$  that co-occur most frequently in raw user trajectories and the weight is given by the frequency of co-occurrences (i.e., the number of times  $p$  and  $q$  appear consecutively in the trajectories of all users). Intuitively, a path along the edges in the location layer represents a popular travel sequence.

### 3.2 Unsupervised Node Representation Learning

We extend the R-GCN framework [Schlichtkrull et al., 2018], which is aimed for semi-supervised entity classification and being used as an entity encoder for existing link prediction models on knowledge graphs, to unsupervised node representation learning on heterogeneous graphs.

R-GCNs use the following propagation rule expressed in the message-passing architecture [Gilmer et al., 2017] that

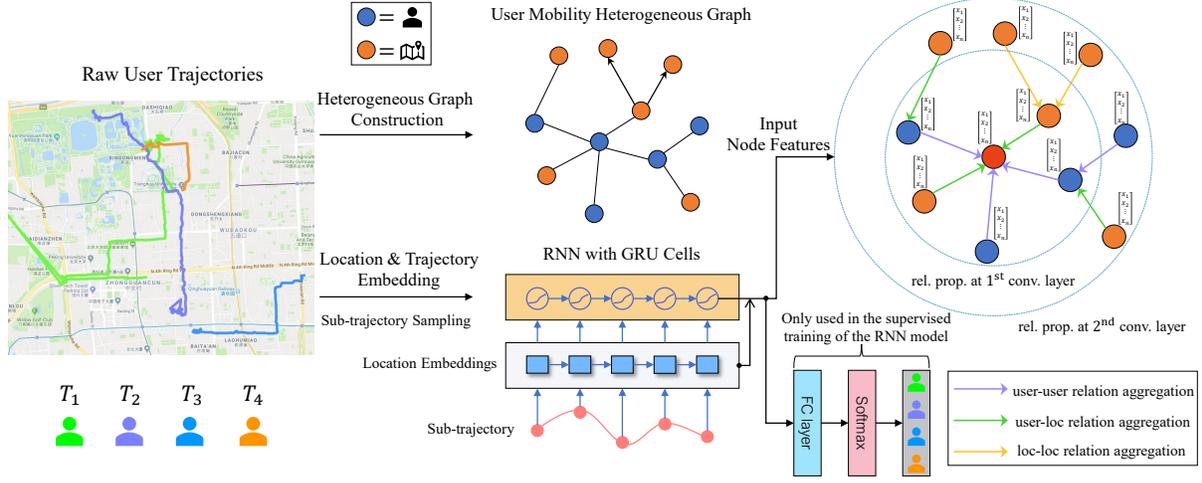


Figure 2: Workflow of our proposed method. Raw user trajectories are first extracted to construct the User Mobility Heterogeneous Graph. Meanwhile, sub-trajectories are sampled to learn location embeddings using a skip-gram based model, and trajectory embedding is formed by averaging the final hidden state of a RNN model over all sampled sub-trajectories of the user. The RNN model is trained in a supervised way under the Trajectory-User Linking problem. The location and trajectory embeddings are feed to the GCN as input node features.

aggregates information from a node’s local neighbors and forward the aggregated information to the next layer.

$$h_i^{(l+1)} = \sigma \left( \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \right) \quad (1)$$

where  $\mathcal{N}_i^r$  denotes the neighbors of node  $i$  under the relation  $r \in \mathcal{R}$ , and  $c_{i,r}$  is just a normalization constant.  $\sigma$  denotes an activation function.  $h_i^{(l)}$  is the hidden state of  $i$ -th node at the  $l$ -th layer. It assigns a different weight  $W_r^{(l)}$  for each different relation  $r \in \mathcal{R}$  and a self-connection is added. This layer-wise propagation model can be implemented in sparse-dense matrix multiplications and has a computational complexity linear in the number of edges.

There are three types of relations in our User Mobility Heterogeneous Graph: user-user meeting, user-location visiting and location-location co-occurrence (ground truth social pairs are not available in the unsupervised setting). All these relations must be explicitly modeled and have separate weights. We can express the propagation rule in the matrix form:

$$\begin{aligned} H^{(l+1)} &= \sigma \left( \tilde{D}_{UU}^{-\frac{1}{2}} \tilde{A}_{UU} \tilde{D}_{UU}^{-\frac{1}{2}} H^{(l)} W_{UU}^{(l)} \right. \\ &\quad + \tilde{D}_{LL}^{-\frac{1}{2}} \tilde{A}_{LL} \tilde{D}_{LL}^{-\frac{1}{2}} H^{(l)} W_{LL}^{(l)} \\ &\quad \left. + D_{UL}^{-\frac{1}{2}} A_{UL} D_{UL}^{-\frac{1}{2}} H^{(l)} W_{UL}^{(l)} \right) \quad (2) \end{aligned}$$

where  $\tilde{A}_{UU} = A_{UU} + I$  is the adjacency matrix of the heterogeneous graph contains only user-user edges with added self-connections (in the user nodes),  $\tilde{D}_{UU}$  is corresponding degree matrix and  $W_{UU}^{(l)}$  is the weight matrix for user-user relation in  $l$ -th layer.  $\tilde{A}_{LL}, \tilde{D}_{LL}, W_{LL}^{(l)}$  is defined in the same way for location-location relation. For  $A_{UL}, D_{UL}, W_{UL}^{(l)}$  that regard to user-location relation, we do not add self-connections in the adjacency matrix since it is a bipartite relation between two different kinds of nodes. All these adjacency matrices

have the same dimension: ( $\#users + \#locations, \#users + \#locations$ ).  $H^{(l)}$  is the activations in the  $l$ -th layer and  $H^{(0)}$  is the input node features.  $\sigma$  symbolizes an activation function (here we use ReLu activation).

Inspired by the heterogeneous skip-gram model [Dong *et al.*, 2017], we introduce a loss function designed to learn node embeddings in a fully unsupervised manner. We encourage neighbor nodes to have similar representations while distant nodes have distinct ones. The loss for a single node  $u$  is:

$$\begin{aligned} \mathcal{L}_{\text{unsup}}(u) &= -\log(\sigma(\Phi_u^\top \Phi_v)) - \lambda_{\text{heter}} \log(\sigma(\Phi_u^\top \Phi_w)) \\ &\quad - \lambda_{\text{neg}} \sum_{z \in \text{NEG}(u)} \log(-\Phi_u^\top \Phi_z) \quad (3) \end{aligned}$$

$\Phi(u)$  here denotes the output embedding for node  $u$ ,  $\sigma$  is the sigmoid function.  $v$  represents a homogeneous node (i.e., if  $u$  is an user node, then  $v$  is also an user node) in the local context of node  $u$ , and  $w$  represents a heterogeneous node in this local context.  $v$  and  $w$  may be chosen from nodes near  $u$  in a random walk.  $\text{NEG}(u)$  here is the negative sampling set of node  $u$  which contains nodes not in the local context of  $u$ .  $\lambda_{\text{heter}}$  and  $\lambda_{\text{neg}}$  control the weights of the heterogeneous context loss and the negative sampling loss, respectively. The loss of the entire graph is averaged over all nodes.

### 3.3 Extracting Node Features from Trajectories

Since GCNs leverage node features to learn node representations, we shall provide useful, predictive user and location embeddings as input. We adapt the method in [Gao *et al.*, 2017] to learn embeddings for users and locations.

First we divide each user’s trajectory into a series of sub-trajectories. After the splitting, location embeddings are learned using a skip-gram based model: for a given location  $l$ , we learn its embedding  $\Psi_l$  by maximizing the probability of predicting its context  $C(l, T)$  (nodes co-occurs with  $l$  in

the trajectory  $T$  where  $l$  lies):

$$\max \prod_{l' \in C(l, T)} \frac{\exp(\Psi_l \Psi_{l'})}{\sum_{l'' \in C(l, T)} \exp(\Psi_l \Psi_{l''})} \quad (4)$$

After embeddings are learned for each location, the sub-trajectories after location embedding are then feed into a RNN with GRU cells. To train the network in a supervised manner, the final hidden state then goes through a FC layer with softmax to link the sub-trajectories to their users. The final hidden state of the RNN is used as sub-trajectory embedding. A user’s embedding is given by averaging the embeddings of all sampled sub-trajectories of that user.

### 3.4 Semi-supervised Node Representation Learning

We can further improve the performance of our method if a portion of friend pairs from the social network is known. In this scenario, we use two ways to augment our model. First, we incorporate the neighbor information of nodes as local contexts into the loss function. We compute the following semi-supervised loss term  $\mathcal{L}_{\text{semi}} = -\log(\Phi_u^T \Phi_z)$ , where  $z$  is a neighbor of node  $u$  in the social network. The final loss function in the semi-supervised setting is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{unsup}} + \lambda_{\text{semi}} \mathcal{L}_{\text{semi}} \quad (5)$$

where  $\lambda_{\text{semi}}$  is used to control the contribution of the semi-supervised loss to the overall loss.

Second, we treat the partial social network as a new kind of user-user relation (as described in Section 3.1) and use a separate weight matrix for it in the convolution. Concretely, we add the term  $\tilde{D}_S^{-\frac{1}{2}} \tilde{A}_S \tilde{D}_S^{-\frac{1}{2}}$  to Equation (2), where  $\tilde{A}_S, \tilde{D}_S$  are the adjacency matrix and the degree matrix of the known partial social network with self-connections on user nodes.

## 4 Experiments

### 4.1 Experiment Setup

#### Datasets

We have conducted extensive experiments on three publicly available real-world datasets: Gowalla, Brightkite [Cho *et al.*, 2011] and Foursquare [Yang *et al.*, 2019], which contain user check-in records and social relations from three LBSNs. As in previous works [Yu *et al.*, 2018; Backes *et al.*, 2017; Zhou *et al.*, 2018], we focus on a specific city for each dataset. User check-ins within the selected cities are extracted by using a geographic bounding box. For each dataset, we obtain the ground truth friendship in social networks as positive instances. Then we substitute one of the users in every friend pair with a random non-friend user to build negative instances. Table 1 shows the statistics of the datasets.

For unsupervised learning, we split the friendship data into 20% validation set and 80% test set. For semi-supervised learning, we first divide the data into two parts: a 30% semi-supervised sampling set and a remaining part composed of the rest 70% data, which is further split into 10% validation set and 90% test set.

	Gowalla	Brightkite	Foursquare
City	Austin	Chicago	New York
#locations	18,382	9,239	43,226
#users	7,844	2,273	9,083
#check-ins	332,742	57,710	283,458
#friends	37,978	4,596	14,187

Table 1: Dataset statistics.

### Metrics

Each method learns a vector representation for each user; the pairwise cosine similarity of the embedding vectors are then computed to generate a score that indicates the probability of two users being friends. We use Area Under the ROC Curve (ROCAUC) to measure the performance of different methods in both unsupervised setting and semi-supervised setting.

### Training Configuration

We use 256-dimensional vectors for both location and trajectory embeddings. After embeddings are learned, we use zero padding to form 512-d vectors as GCN input features, where the first 256-dims are for users and the last 256-dims are for locations. Location pairs with a co-occurring frequency  $\geq 2$  are added to the location layer of the heterogeneous graph.

We follow the two-layer GCN setup in [Kipf and Welling, 2017]. For both unsupervised and semi-supervised settings, we optimize hyperparameters on the validation set. The settings after hyperparameter tuning are reported. We use an output embedding dimension of 128, and the number of units in the hidden layer is set to 256. We use Adam optimizer with a learning rate of 0.0001. L2 regularization is used for the first GCN layer with a weight decay coefficient of  $5 \cdot 10^{-4}$ . The number of negative samples is set to 20 and the weight of the negative sampling loss  $\lambda_{\text{neg}}$  is set to 1.0. Dropout is not used. The weight of the heterogeneous context loss  $\lambda_{\text{heter}}$  in the unsupervised setting for Gowalla, Brightkite and Foursquare is 0.9, 0.1, 0.3, respectively. For semi-supervised learning on Gowalla, Brightkite and Foursquare, the weight is 0.1, 1.0 and 0.4. The weight of the semi-supervised loss  $\lambda_{\text{semi}}$  is set to 2.0 for Gowalla, 5.0 for Brightkite and 2.0 for Foursquare. We train a maximum of 5000 epochs in each case. Our source code is available at [https://github.com/libertyeagle/gcn\\_mobility\\_relationship](https://github.com/libertyeagle/gcn_mobility_relationship).

### Baselines

In the unsupervised setting, we compare our method with five state-of-the-art methods. All compared methods also use an embedding dimension of 128. We also compare the performance of raw features used in the GCN to investigate the role they play. Our method is denoted as Heter-GCN.

- **DeepWalk:** It performs random walks on graphs and uses the skip-gram model to learn node embeddings. Since it is designed for homogeneous graphs, we apply DeepWalk on the user layer of the User Mobility Heterogeneous Graph. [Perozzi *et al.*, 2014].

	Gowalla	Brightkite	Foursquare
DeepWalk	0.6340	0.5364	0.5516
LINE	0.6268	0.5372	0.5565
emb	0.6280	0.5397	0.5475
walk2friends	0.6824	0.5287	0.5956
metapath2vec	0.6571	0.5547	0.6115
Raw features	0.5532	0.5565	0.6198
Heter-GCN	<b>0.7280</b>	<b>0.5762</b>	<b>0.6445</b>

Table 2: Performance comparison for unsupervised learning.

- **LINE**: LINE is a network embedding method designed to preserve the first-order and second-order proximity. We also apply LINE on the user graph. [Tang *et al.*, 2015].
- **emb**: emb applies skip-gram based model on user graph to learn embeddings. To better utilize edge weights, hierarchical sampling is proposed. [Yu *et al.*, 2018].
- **walk2friends**: walk2friends applies a skip-gram based graph embedding model to learn user embeddings on the user-location bipartite graph for social link inference. [Backes *et al.*, 2017].
- **metapath2vec**: This is a state-of-the-art representation learning model for heterogeneous networks which utilizes meta-path based random walks and heterogeneous skip-gram model to learn node embeddings. [Dong *et al.*, 2017].
- **Raw features**: This approach only uses user embeddings and location embeddings, which are fed into the Heter-GCN as input node features.

#### 4.2 Findings and Analysis in Unsupervised Setting Findings - Our Method Outperforms All Baselines

The results obtained on the three datasets in the unsupervised setting are reported in Table 2. We find out that our method performs significantly better than other approaches on all of the three datasets, since our method not only uses external node features but also exploits the heterogeneous graph structure to capture multidimensional relations. The convolutional nature of GCN allows the model to extract the local stationary structures and propagate information at different scales, which also contributes to the performance gain.

Besides, using raw features alone already makes an adequate classifier. But the User Mobility Heterogeneous Graph contains rich structural information of relations which cannot simply be ignored, only through GCN’s combination of both semantic and structural information attains the best performance. We notice that raw features perform relatively worse on Gowalla. That is because the graph constructed on Gowalla is far denser than the other two datasets, the graph-based methods can perform well using rich information the graph provides. Concretely, there are 26,226 nodes and 516,842 edges (272,839 user-user edges) in the heterogeneous graph on Gowalla. While on Brightkite, there are 11,512 nodes and 16,760 edges (1,156 user-user edges). On Foursquare, there are 52,309 nodes and 194,529 edges (13,937 user-user edges).

We observe that walk2friends achieves great performance on Gowalla and Foursquare, because it characterizes user mobility profiles by learning embeddings on the user-location bipartite graph. However, walk2friends performs the worst on Brightkite. This is most likely caused by the sparsity of check-in data on Brightkite.

On Brightkite, emb performs slightly better than DeepWalk and LINE since it uses hierarchical sampling to enhance the contributions of edges with larger weights in user graph. Yu *et al.* [2018] suppose that edges with larger weights indicate strong relations, while low-weight edges are more likely noisy, though this is not always the case. For instance, two individuals who work in the same office building may frequently meet, but they may not necessarily be friends. This leads to its relatively poor performance on Gowalla and Foursquare. Yu *et al.* [2018] mitigate this problem by considering POI’s categorical information to weight meeting frequency. Nevertheless, POI information is not easy to obtain; therefore this method is not generally applicable.

Not surprisingly, metapath2vec demonstrates decent performance on all three datasets, indicates that the User Mobility Heterogeneous Graph which combines relational information both within as well as between users and locations does help improve the overall performance. Still, because it does not explicitly model different kinds of relations, but only rely on the heterogeneous skip-gram loss function to learn network embeddings, its performance is worse than Heter-GCN.

#### Findings - Our Method Also Shows Superiority in Relationship Propagation

As suggested in [Yu *et al.*, 2018], taking account of relationship propagation is an indispensable factor in making precise predictions. For instance, user *A* and *B* have a strong social relation, so does user *B* and *C*. Then we can infer that user *A* and *C* may also have a strong relation even if we have not directly observed any meeting events between them. From Table 3 we can observe that the majority of friend pairs in all three datasets have a meeting frequency less than 2. Hence we must handle these low-frequency friend pairs properly to achieve high performance. Using meeting frequency alone is not able to identify these pairs, since low meeting frequency indicates that the two users are unlikely to be friends. For example, in Gowalla, among all user pairs who have zero meeting frequency, only 0.10% of them are friends; the percentage increases to 1.61% for user pairs with one meeting and 2.90% with two meetings. For pairs with more than 10 meetings, 56.47% of them are friends. Only through modeling relationship propagation, we can solve this issue.

Hence we compare the performance of our approach with one of the best performing user graph based embedding method: emb, in order to demonstrate the GCN framework’s superiority in layer-wise relation propagation. Since our method also takes advantage of node features and utilizes user-location interactions in the heterogeneous graph which enables relationship propagation between users and locations, we also add walk2friends and raw features to our comparison. Figure 3 shows the results. Here we use Area Under the Precision-Recall Curve (PRAUC) instead of ROCAUC because precision-recall curves are better in evaluating per-

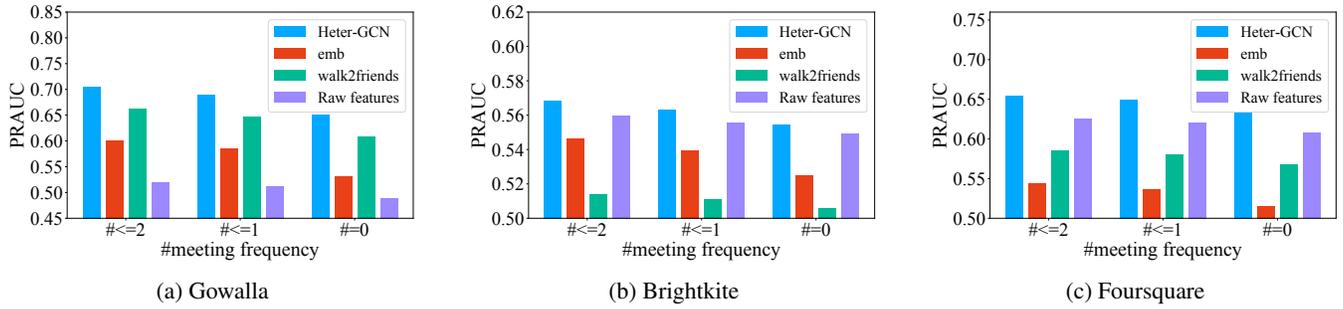


Figure 3: Performance of different methods on user pairs whose meeting frequency is lower than certain threshold.

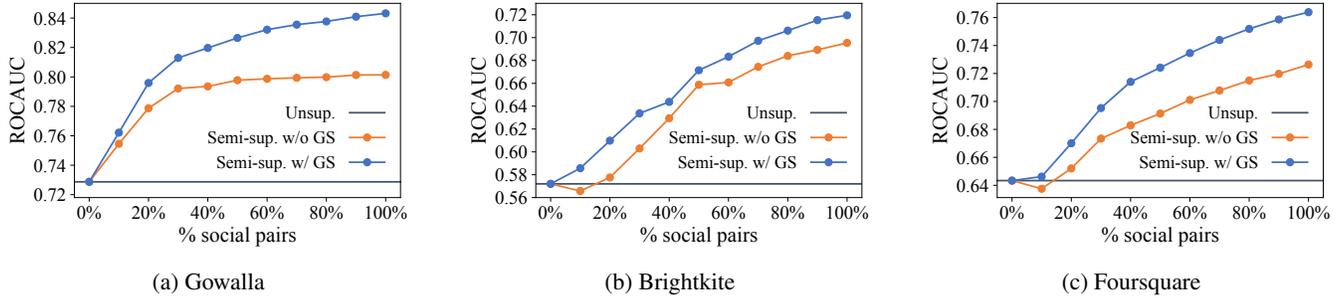


Figure 4: Performance of our method in the semi-supervised setting. 0% social pairs refers to the unsupervised model.

	Gowalla	Brightkite	Foursquare
#friends	37,978	4,596	14,187
#meeting = 0	82.26%	97.33%	96.67%
#meeting = 1	8.85%	1.16%	1.92%
#meeting = 2	3.25%	0.57%	0.61%
#meeting > 2	5.64%	0.94%	0.80%

Table 3: Friendship statistics under different meeting frequency. The percentages here are relative to #friends of the dataset in the first row.

formance with class imbalance, while ROC curves can be deceptive in this circumstance [Davis and Goadrich, 2006].

We observe from Figure 3 that our method yields substantially better performance than other methods. It is apparent that our method can better model relationship propagation, owing to the convolutional property of GCN. In addition, assimilating user-location interactions into the graph structure and exploiting the semantic information that input node features encompass also contributes to the performance gap.

### 4.3 Finding and Analysis in Semi-supervised Setting

#### Finding - Our Method Achieves a Huge Performance Boost given Only a Small Portion of the Social Network

We assume that only a small portion of edges in the social network are available as ground truth. We randomly sample 10% to 100% edges from the 30% semi-supervised sample set we split to construct the partial social network, then feed it to GCN for semi-supervised learning. As stated in Section 4.1, the evaluation is performed on the split test set which does not include the sampled edges. To show the effective-

ness of treating the partial social network as a new kind of relation in the , we also evaluate the performance of semi-supervised learning using only ground truth neighbor information in skip-gram loss, namely Semi-sup. w/o GS. The method which also utilizes the structure of the partial social network is designated as Semi-sup. w/ GS. The results on the three datasets are presented in Figure 4.

We discover that we obtain much better performance compared to the unsupervised training by using only a small portion of ground truth social pairs. Though the performance of Semi-sup. w/o GS drops a bit compared to the unsupervised model on Brightkite and Foursquare when using 10% social pairs in the sample set (which is only 3% in the complete social network), this is probably because we fail to capture the global structure of the social network given limited data observed. Still, as the number of social pairs used in training increases, so does the performance leap. Moreover, by examining the results of Semi-sup. w/o GS and Semi-sup. w/ GS, we find that letting GCN learn from the structure of the partial social network rather than rely only on neighbor information it provides, further performance improvements are realized.

## 5 Conclusion

In this paper, we developed GCNs on User Mobility Heterogeneous Graphs for social relationship inference from human trajectories. Extensive experiments on three datasets demonstrate the effectiveness of our method in both unsupervised and semi-supervised settings.

## Acknowledgments

The work was supported in part by grants from the National Natural Science Foundation of China (Grant No. U1605251, 61832017, 61631005 and 61502077).

## References

- [Backes *et al.*, 2017] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. Walk2friends: Inferring social links from mobility profiles. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17*, 2017.
- [Bruna *et al.*, 2014] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR)*, 2014.
- [Cho *et al.*, 2011] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 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, KDD '11*, 2011.
- [Davis and Goadrich, 2006] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, 2006.
- [Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29*, pages 3844–3852. Curran Associates, Inc., 2016.
- [Dong *et al.*, 2017] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17*, 2017.
- [Eagle *et al.*, 2009] Nathan Eagle, Alex (Sandy) Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.
- [Gao *et al.*, 2017] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. Identifying human mobility via trajectory embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017.
- [Gilmer *et al.*, 2017] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning, ICML '17*, 2017.
- [Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, pages 1024–1034. Curran Associates, Inc., 2017.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
- [Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, 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 *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13*, 2013.
- [Pham *et al.*, 2016] Huy Pham, Cyrus Shahabi, and Yan Liu. Inferring social strength from spatiotemporal data. *ACM Trans. Database Syst.*, 41(1):7:1–7:47, 2016.
- [Scellato *et al.*, 2011] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, 2011.
- [Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)*, 2018.
- [Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, 2015.
- [Wang *et al.*, 2014] H. Wang, Z. Li, and W. Lee. Pgt: Measuring mobility relationship using personal, global and temporal factors. In *2014 IEEE International Conference on Data Mining, ICDM '14*, 2014.
- [Yang *et al.*, 2019] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In *Proceedings of the 2019 World Wide Web Conference, WWW '19*, 2019.
- [Yu *et al.*, 2018] Yanwei Yu, Hongjian Wang, and Zhenhui Li. Inferring mobility relationship via graph embedding. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(3):147:1–147:21, 2018.
- [Zhou *et al.*, 2018] Fan Zhou, Bangying Wu, Yi Yang, Goce Trajcevski, Kunpeng Zhang, and Ting Zhong. Vec2link: Unifying heterogeneous data for social link prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, 2018.