Spatio-Temporal Prediction of Social Connections

Guolei Yang
Iowa State University
Ames, Iowa, USA 50011
yanggl@iastate.edu

Andreas Züfle
George Mason University
Fairfax, Virginia, USA 22030
azufle@gmu.edu

ABSTRACT

It is long known that a user's mobility pattern can be affected by his social connections. Users tend to visit same locations visited by their friends. In this paper we investigate the inverse problem: How does a set of user trajectories reflect their social connections. To this end, we define the social connection prediction problem. Given two users, predict the probability that they are friends by mining their historical trajectories. A first approach to do so is to exam how often the two users visit the same location at the same time, which suffers from the problem that different locations/times may have different predictive power. We propose a comprehensive prediction model that is able to capture this difference between locations and time slots. To demonstrate its effectiveness, we trained the proposed model using the publicly available Foursquare dataset. The result shows the proposed model is able to predict existence of social connections between randomly selected users significantly more accurate comparing with the naive method.

CCS CONCEPTS

•Information systems → Location based services; Data mining; •Computing methodologies → *Machine learning;*

KEYWORDS

Location-Based Social Network, social connection prediction, feature selection, spatio-temporal data

ACM Reference format:
Guolei Yang and Andreas Züfle. 2017. Spatio-Temporal Prediction of Social Connections. In *Proceedings of GeoRich'17 , Chicago, IL, USA, May 14, 2017,* 6 pages.
DOI: http://dx.doi.org/10.1145/3080546.3080551

1 INTRODUCTION

In the past decade, with the rise of Location-Based Social Networks (LBSN), huge amount of geo-spatial data is collected on a daily basis. For example, the Foursquare[15] dataset contains more than 30 millions of self-reported check-ins from thousands of users. As a result, it becomes possible to mine spatio-temporal data and study human mobility pattern at unprecedented large scale.

Studies on human mobility patterns reveal that a user's movement can be affected to certain extent by his social connections [2, 17]. The underlying hypothesis here is that friends are likely to visit the same locations during the same time periods, which could be the result of attending the same social events. As such, it is possible to predict a user's future movement by mining the historical trajectory of his friends on a LBSN. These studies have since inspired a series of research efforts towards the prediction of future individual movements (e.g., [4, 7, 10, 12]). Another research direction (e.g., [14]) focuses on exploring historical trajectories to identify users who share similar interests for locations.

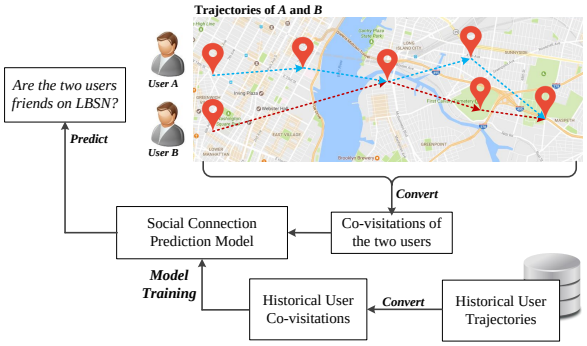

Figure 1: An illustration of spatio-temporal social connection prediction.

Towards the goal of a more thorough understanding of human mobility patterns, we propose to investigate the inverse problem: How does a set of users' trajectory reflect their social connections. We define the social connection prediction problem: Given the trajectories of two LBSN users u_i and u_j, we aim to model the probability that u_i and u_j are friends on the LBSN using their trajectories. Social connection prediction is a long standing research topic. To the best of our knowledge, none of the existing approaches exploit the users' trajectories for link prediction. The focus of this paper is not to compete with, but to supplement existing methods by exploring a new dimension of data source.

A straightforward way to predict the social connection, or the lack thereof, between two users is to examine the *spatio-temporal overlap* of their trajectories, i.e., find events where the two users visit the same location at the same time on their trajectories. We define such an event as a *co-visitation* of the two users. The assumption is, if two users frequently visit the same location during the same time peroid, their likelihood of being friends increases. Thus the occurance of co-visitations could reflect when and where they were meeting. The same assumption is used to identify similar users in [14]. Algorithms such as co-location mining [13] can be used to discover co-visitations among users.

Although a solution for link prediction based on just the above assumption is reasonable, it suffers from two problems. First, it treats all locations equally in predicting social connections, which

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GeoRich'17 , Chicago, IL, USA
© 2017 ACM. 978-1-4503-5047-1/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3080546.3080551

is not realistic. For example, if two users frequently meet at private locations like someone's house, or a small coffee shop, it is very likely that they know each other. However, if they both check-in to the same Walmart supermarket after work, it might be just an coincidence simply because it is the only supermarket near their home. Second, this method ignores the time difference of check-in behaviours. If two users both check-in to a restaurant at 6:00pm, it is not as significant as two users visit the same location at 10:00pm. This is because most customer of the restaurant may choose to dine there around 6:00pm, but if two users both decide to check-in there at 10:00pm, the chances are higher that they may be attending the same social event. Although the technique proposed in [14] considered the impact of different granularity of locations (e.g., the same state v.s. the same city), it does not explicitly distinguish the predictive power of different locations/time for different users.

We propose a more comprehensive methodology to study the social connection prediction problem. Unlike the naive solution, we assume different locations and different time slots have different predictive power. As such, we propose a social connection prediction model in which the predictive power of each location and time slot are modelled by latent variables. The proposed model is based on a novel data structure termed *Spatio-Temporal Co-visitation Matrix*. Additionally, our model also takes into consideration the geographic distance between the user's home/work location to the co-visitation locations. Using the users' social connections on Foursquare as ground truth, we show that the proposed model outperforms the naive algorithm that counts only the number of co-visitations. We summarize our contributions as follows:

- We study how the trajectories of a set of users reflect their social connections. To this end, we define the social connection prediction problem: Given the trajectories of two LBSN users u and v, model the probability that u and v are friends on the LBSN.
- Our key observation is: different locations and time may have different predictive power, which is in accordance with common sense. As such, we propose a social connection prediction model that is able to capture this difference among locations and times using latent variables.
- We evaluate effectiveness of the proposed model using the Foursquare dataset. The result shows the proposed method outperforms the naive trajectory overlap based solution.

The rest of the paper is organized as follows: Related works are summarized in Section 2. We formally define the problem and give an overview of our solution in Section 3. Section 4 presents the proposed model in detail. Experiment results are showed in Section 5. And finally, Section 6 concludes the paper.

2 RELATED WORK

The spatio-temporal social connection prediction problem we study in this paper is directly related to link prediction problem on social networks. Given the snapshot of a social network at time t, the goal of link prediction is to predict links, i.e., social connections, that will emerge at a later time, or to identify missing links at t. Such missing links could be the result of privacy settings, e.g., a user may want to hide his friend list from the general public.

Existing works in the field mainly explore two types of information in predicting links: 1) Network structure, i.e., existing social connections, and 2) node attributes such as user profiles. We briefly summarize some representative works. The relational learning [9, 11, 20] and matrix factorization-based [8] techniques both leverage attribute information for link prediction. The Supervised Random Walk (SRW) technique proposed in [1] combines networks structure and edge attributes to improve prediction accuracy, but does not fully explore node attributes. In [19], network structure and node attributes are integrated with a Social Attribute Network (SAN) model, which is later generalized in [5] to both predict links and infer missing attributes.

Our problem is also closely related to [14], which proposes to explore trajectory data to identify users who share similar interests in locations, in order to make friend recommendations on LBSNs, which is not the focus of our study. In a recent work [18], the authors proposed a community discover method that leverage spatio temporal co-occurrences. In their work, spatio temporal data is used in complementary to other information, e.g., network proximity, to discover how likely certain users belong to the same community. In contrast, our work focus on answering the following fundamental question: Is it possible (and to what extent) to predict the existence of social connections among two users by *only* looking at their spatio temporal information? Form this perspective, our work intends to complement existing studies on human mobility patterns.

3 OVERVIEW

3.1 Problem Statement

We define a user's trajectory as a series of timestamped check-ins, where each check-in indicates the exact place (i.e., a restaurant, a coffee shop, etc.) the user visits, instead of a geo-graphical coordinate. The Foursquare dataset is an example of such trajectory that consists of self-reported check-ins. Note that coordinate-based trajectory can be converted into such check-ins by joining the coordinates with a database of Point-of-Interests (PoI), such as provided by Open-Street Map. For simplicity, we consider only check-in-based trajectory in this paper. We formally define the notion of *Check-in* and *User Trajectory* as follows.

Definition 3.1 (Check-in). Let $\mathbb{U}$ denote a set of unique user identifiers, $\mathbb{L}$ denote a set of locations, and $\mathbb{T}$ denote the time domain. A check-in c is a triple $(u, l, t) \in \mathbb{U} \times \mathbb{L} \times \mathbb{T}$, which indicates the user u has visited location l at time t.

Definition 3.2 (User-trajectory). Let $\mathbb{C}$ be a collection of check-ins and let $u \in \mathbb{U}$ denote a user, then the set $C_u := \{(u, l, t) \in \mathbb{C}\}$ is the user-trajectory (or simply trajectory) of u.

The proposed social connection prediction model is based on the concept of *Co-visitation*, which is defined as follows:

Definition 3.3 (Co-visitation). A co-visitation of two users u_i and u_j to a location l is defined as the event that u_i and u_j report two check-ins (u_i, l, t_i) and (u_j, l, t_j) respectively, where $|t_i - t_j| \leq \tau$.

Here τ is an experience-based parameter called the co-visitation time window. We formulate the social connection prediction problem as a classification problem. Given the trajectory of two users

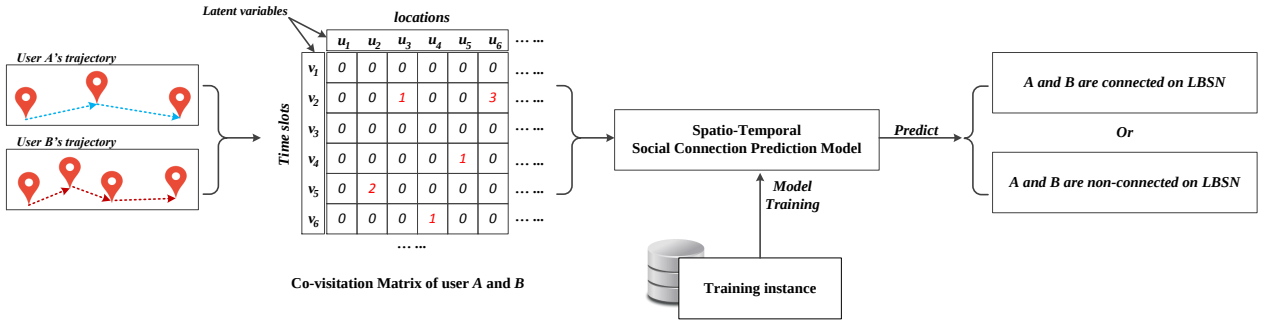

Figure 2: General steps of the proposed method.

u_i and u_j, the goal is to assign the pair of users (u_i, u_j) into one of the two classes: *Connected* or *Not-connected*.

Note that the above problem formulation implicitly assumes the social connection is symmetric, i.e., if u_i is a friend of u_j, then u_j is also a friend of u_i. This type of social connection is common on most social networks like Facebook. However, on some social networks, the connection can be one-way. For example, on Foursquare, a user can choose to "follow" other users and thus formulates a one-way social connection, where one user is the *follower* and the other being the *followee*. The one-way social connection can be model by treating (u_i, u_j) and (u_j, u_i) as different instance which can be assigned into different classes.

3.2 Methodology

We model the probability of the existence of social connection between two users based on the hypothesis that socially connected users tend to visit same locations at same time periods, which is defined as co-visitations. However, we observe that in reality not all co-visitations are equally important in terms of predicting user's social connections. We propose a three step model learning process to capture this difference (Figure 2).

- **Co-visitation Matrix formulation** Given the trajectories of two users u_i and u_j, we first convert their trajectories into a spatio-temporal co-visitation matrix that records the time and location of their co-visitations.
- **Probability estimation** The probability that u_i and u_j are socially connected is computed based on their co-visitation matrix.
- **Model learning** The latent variables in the model are estimated by optimizing a loss function, which measures the prediction error between actual class and predicted class for each pair of users in the training set.

We present details of these three steps in the next section.

4 PROPOSED SOLUTION

4.1 Spatio-Temporal Co-visitation Matrix

Given the trajectories C_{u_a} and C_{u_b} of two users u_a and u_b, respectively, their spatio-temporal co-visitation matrix $M(a, b)$ is a $m \times n$ matrix where m is the total number of locations in $\mathbb{L}$ and n is the number of time slots. The i-th row in $M(a, b)$ corresponds to the i-th location while the j-th column corresponds to the j-th time slot. As such, if the two users had $x \in \mathbb{N}$ co-visitation to the i-th location that occurred within the j-th time slot, $M(a, b)_{i,j}$ equals x. Figure 2 illustrates a co-visitation matrix generated for two users. Note that the co-visitation graph is usually highly sparse.

The granularity of locations and time slots used to build the co-visitation matrix can be adjusted as needed, i.e., each location can be an exact PoI or a geographic region with arbitrary size. The purpose of this mechanism is to provide the users with the flexibility to control the number of model parameters need to be learnt from training data. If a large and diverse set of check-in is available, more locations and time slots can be used. However, if the number of labelled instance is limited, using a large number of parameters may risk over-fitting the model. The appropriate location granularity and partition of time slots can be empirically selected based on experiment results on the specific dataset, i.e., starts from a large location and time granularity, then gradually increase the granularity settings in a cross-validation process until the desired prediction accuracy is achieved. The regularization terms introduced later in the proposed model is in place to assure the appropriate setting parameters can be reached in this process.

Efficient co-visitation detection: The co-visitations of two given users can be detected by exhaustive searching, i.e., for each check-in of a user u_a, exams every check-in of u_b to see if they formulate a co-visitation. This is obviously inefficient especially when the number of users are large in generating the training instances. Here we design a more efficient co-visitation detection algorithm. Due to limited space, we give a high-level description of the algorithm:

(1) Each check-in $c = (u, l, t)$ is converted into two tuples $c_s = (u, l, t - \frac{\tau}{2})$ and $c_e = (u, l, t + \frac{\tau}{2})$, where τ is the co-visitation time window. As such, the time of each check-in is seen as a time period $[t - \frac{\tau}{2}, t + \frac{\tau}{2}]$. Two check-ins to the same location l is a co-visitation if and only if their time periods overlap.
(2) All tuples are sorted by the time stamp in ascending order.
(3) For each possible co-visitation location, initialize an empty list. Initialize the co-visitation matrix to be all zeros.
(4) The algorithm then performs a running count by scanning through the sorted tuples one by one.
(5) When c_s is encountered for some check-in c, it is is added to the list of location l. If l is not empty, a co-visitation is detected. Update the co-visitation matrix accordingly.
(6) When c_e is encountered for some check-in c, remove c_s of the same check-in from the list of location l.

Note that the proposed algorithm is a time-sweep-algorithm which has log-linear complexity to the total number of check-ins ($O(n\log(n))$ to sort the check-ins and $O(n)$ to sweep, where n is the total number of check-ins). More importantly, it is able to detect co-visitation for a set of users in a parallel manner. Let C_{max} denote the maximal number of check-ins reported by a single user. In order to find all co-visitations among a set of users, the exhaustive search algorithm needs to process each pair of users one by one, thus have an overall complexity of $O(|\mathbb{U}|^2 * C_{max}^2)$. In contrast, the proposed algorithm has complexity $O(|\mathbb{U}| * C_{max} * \log(|\mathbb{U}| * C_{max}))$, which is much faster than exhaustive search.

4.2 Social Connection Prediction Model

In our model, both locations and time slots are mapped into a multidimensional latent feature space. We use $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$ to denote the latent variables for locations and time slots, where $x_i \in X$ can be seen as a weight that measures the significance of the i-th locations. Similarly, $y_j \in Y$ measures the significance of the j-th time slot. Given the co-visitation matrix $M(a,b)$, we can then estimate how likely u_a and u_b are socially connected using a weighted sum over the matrix, defined as follows:

$$s(a,b) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_i y_j M(a,b)_{i,j} \tag{1}$$

where $s(a,b)$ can be seen as a "score". The higher the score, the more likely u_a and u_b are socially connected. We employ a logistic transformation, using the sigmoid function to convert $s(a,b)$ into an estimated probability for the classification problem.

$$\hat{Pr}((u_a,u_b) \text{ is connected}) = \frac{1}{1+e^{-s(a,b)}} \tag{2}$$

If the predicted probability is higher than a decision threshold, denoted by λ (which is to be learnt from the data), the user pair is classified as *connected*, otherwise *non-connected*.

The above model, however, does not take into consideration geographic distance between locations. The same location may have different significance for users living in different areas. The geographic distance between a user's home/work and the co-visitation locations can been seen as a personalized parameter to adjust the significance of a location. To this end, we modify Equation 1 by adding the *distance coefficients* $W = \{w_1, w_2\}$ to our model:

$$s(a,b) = \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(x_i y_j + w_1 D(i,a) + w_2 D(i,b)\Big)M(a,b)_{i,j} \tag{3}$$

Here, w_1 and w_2 are the two distance coefficients, and $D(i,a)$ measures the geographic distance between the i-th locations and u_a's home base. For simplicity, we use the geographic center of u_a's all check-ins as the estimated home base. Nevertheless, more complex method such as algorithms proposed in [2, 16] can also be used for more strict estimation. Note that by introducing the distance coefficients we only added two more parameters into our model, but it allows the classification results to be "personalized" to some extent by involving the two users' home base locations into the model.

4.3 Model Learning

Parameters in Equation 3 can be learned from a set of labelled training data by optimizing the following function:

$$\arg\min_{X,Y,W} \sum_{\forall u_a,u_b \in \mathbb{U}} E(p_{a,b}, \hat{p}_{a,b}) + \Theta(X,Y) \tag{4}$$

In the above function, $E(\cdot)$ denotes a loss function that measures the prediction error. In this paper we use the indicator function as loss function, which is commonly used for classification problems. $p_{i,j}$ is the label of a training instance (u_a,u_b) while $\hat{p_{a,b}}$ is the predicted result using the proposed model. Finally, $\Theta(X,Y)$ is the regularization term, defined as:

$$\Theta(X,Y) = \frac{\lambda_x}{2}\|X\|_2^2 + \frac{\lambda_y}{2}\|Y\|_2^2 \tag{5}$$

The regularization term is in place to prevent the model from overfitting. It provides a trade-off mechanism between model complexity (in terms of number of learnt parameters) and model performance. As showed in the equation, the regularization coefficients λ_x and λ_y are used to adjust the "penalty" of using more parameters, so that a larger number of parameters will be encouraged only when it significantly improves the prediction outcomes. These two coefficients are usually selected through a cross-validation process on training data. Since W contains only two parameters, no regularization term is introduced for it in the learning process.

Note that in the co-visitation matrix, we assigned a latent variable to each location and each time slot. As a result, it may appear that a total number of $m + n + 2$ parameters need to be learnt from the training data. However, the actual number of parameters can be much smaller. This is because the the co-visitation matrix is usually highly sparse. If the two users have never reported a co-visitation to certain location l, then the corresponding latent variable does not need to be learnt. The value of the variable is simply set to 0. Similarly, the latent variable for a time slot is set to 0 if no co-visitations occurred within the time slot. Eventually, only those locations and time slots that are significant will have a non-zero latent variable value. This makes the model parameters easily interpretable by a human.

5 EXPERIMENTS

5.1 Dataset Description

We evaluate the proposed model on the widely-used Foursquare check-in dataset [15]. In our experiments, we mine the check-in data from two of the most popular cities, including New York City (NYC), USA and Tokyo, Japan. The dataset contains about 227,428 check-ins reported in NYC and 573,703 in Tokyo. The check-ins were collected for about 10 months. From each check-in, we extract the user ID, location ID, and a time stamp. Using the user ID or location ID, we retrieve the profile of that user or location on Foursquare. The user profile includes the social connection between users ("follower - followee") and the location profile includes its category (*Food, Coffee, Nightlife, Fun, and Shopping*), coordinates, and user rating. The check-ins are grouped by user ID/location ID and sorted by their timestamps. We assume the social connections are static, i.e., the friendship states between users are not dynamic changing during the period the check-ins are collected. Modelling the formulation of new social connections is of interest per se,

Table 1: Symmetric social connection prediction results

City	Scheme	τ=15min		τ=30min		τ=45min		τ=60min	
		precision	F1-Score	precision	F1-Score	precision	F1-Score	precision	F1-Score
NYC	Random	0.1102	0.1755	0.1102	0.1755	0.1102	0.1755	0.1103	0.1755
	Count	0.1527	0.2098	0.1545	0.2132	0.1566	0.2159	0.1540	0.2101
	Matrix Factorization	0.2127	0.3005	0.2127	0.3005	0.2127	0.3005	0.2127	0.3005
	Co-visitation	0.2220	0.2917	0.2619	0.3516	0.3001	0.3781	0.2519	0.3314
	Co-visitation+Distance	0.2279	0.2998	0.2759	0.3700	0.3057	0.3850	0.2779	0.3621
Tokyo	Random	0.1008	0.1679	0.1008	0.1679	0.1008	0.1679	0.1008	0.1679
	Count	0.1507	0.2020	0.1525	0.2051	0.1522	0.2044	0.1505	0.2019
	Matrix Factorization	0.2227	0.2791	0.2227	0.2791	0.2227	0.2791	0.2227	0.2791
	Co-visitation	0.2405	0.3019	0.2562	0.3110	0.2490	0.3114	0.2220	0.2817
	Co-visitation+Distance	0.2670	0.3134	0.2759	0.3312	0.2407	0.3099	0.2177	0.2725

Table 2: One-way social connection prediction results

City	Scheme	τ=15min		τ=30min		τ=45min		τ=60min	
		precision	F1-Score	precision	F1-Score	precision	F1-Score	precision	F1-Score
NYC	Random	0.1202	0.1851	0.1202	0.1851	0.1202	0.1851	0.1202	0.1851
	Count	0.1555	0.2140	0.1537	0.2119	0.1540	0.2123	0.1535	0.2125
	Matrix Factorization	0.1723	0.2419	0.1723	0.2419	0.1723	0.2419	0.1723	0.2419
	Co-visitation	0.1696	0.2389	0.1818	0.2750	0.1820	0.2771	0.1702	0.2393
	Co-visitation+Distance	0.1777	0.2501	0.1925	0.2858	0.1925	0.2809	0.1795	0.2505
Tokyo	Random	0.1115	0.1710	0.1115	0.1710	0.1115	0.1710	0.1115	0.1710
	Count	0.1621	0.2208	0.1630	0.2219	0.1643	0.2222	0.1622	0.2217
	Matrix Factorization	0.1707	0.2253	0.1707	0.2253	0.1707	0.2253	0.1707	0.2253
	Co-visitation	0.1659	0.2250	0.1714	0.2323	0.1759	0.2336	0.1657	0.2029
	Co-visitation+Distance	0.1715	0.2309	0.1771	0.2404	0.1804	0.2451	0.1693	0.2249

which is beyond the scope of this paper and we intent to explore in future work. For our experiments, we select a subset of users that satisfy the following two conditions:

- **Check-in Active** Actively report check-ins for a time period of at least 1 month. The average number of check-ins reported per day is no less than 1.
- **Socially Active** The user has followers and also follows others.

These two conditions are in place to filter out the users who do not have enough data or lack the ground truth to test the proposed model. In the user selection process, we applied community detection algorithm [3] among most active users from the two cities and selected two communities for our experiments. Among approximate 2660 users in NYC and 3100 users in Tokyo, the two communities we selected contain 173 and 165 users respectively. The users in one community are from the same city who satisfy both conditions. The two communities have no overlap. Each user has on average 14 social connections. The total number of locations visited by these users is approximately 350 but not all the locations have been co-visited by friends. We show in the following subsections that this small set of users is sufficient to demonstrate the effectiveness of the proposed model.

5.2 Experimental Settings

For comparison purpose, we have implemented the following schemes:

- **Random** This scheme randomly assigns a user pair as friends or non-friends, each with a probability of 50%.
- **Count** This scheme counts the number of co-visitations of two users. If the number is higher than a threshold, the two users are predicted to be friends, and otherwise non-friends. The threshold is set to be the average number of co-visitations of each pair of friends among the selected users.
- **Matrix Factorization** We implement the standard Matrix Factorization [6] algorithm. Each user is represented by a latent vector of size l, and whether two users are friends is predicted by the product of their latent vectors. The latent vectors are learnt with a user-user rating matrix, which is a $n \times n$ matrix M where n is the number of users. if u_i is a friend of u_j then $M_{i,j}$ is set to 1 otherwise 0. Readers are referred to [6] for details about the learning process. In our experiment, we set $l = 5$.
- **Co-visitation** The proposed co-visitation matrix-based model using Equation 1. This model does not take into consideration the geographic distance factor.
- **Co-visitation + Distance** The proposed co-visitation matrix-based model using Equation 3, which involves the geographic distance factor.

The following granularity settings are empirically selected to generate the co-visitation matrix, as they yield the best prediction results. 1) Each location is a specific PoI, but the total number of PoIs used for each pair of users is limited to 25. We observe that in the Foursqaure dataset it is very rare that two users have co-visited more than 25 different locations. Therefore we set this limit to reduce the number of unnecessary model parameters. 2) We partition each day into 6 time slots:(12:00am to 6:00am), (6:01am to 10:00am), (10:01am to 2:00pm), (2:01pm to 5:00pm), (5:01pm to

8:00pm), and (8:01pm to 11:59pm). Note that these time slots are not evenly partitioned. Instead, we choose this typical time slot that reflects different period of a day for work or social events. As such, we use a total of 42 time slots, because each day of a week has 6 slots.

We consider two types of prediction tasks: *Predicting one-way social connections* and *predicting symmetric social connections*. Recall that for one-way social connections, the class of (u_a, u_b) may be different from (u_b, u_a) while for symmetric social connections, their class must be the same. The social connections on Foursquare are originally one-way. A user u_a can choose to follow another user u_b but meanwhile u_b may not be a follower of u_a. Nevertheless, we can extract a subset of symmetrically connected users, i.e., users who follow each other mutually, as training/testing set for symmetric social connection prediction task.

5.3 Experimental Results

The dataset we used is strongly skewed, i.e., about 90% of user-pairs are not socially connected. As such, we choose *precision* and *F1-score* as performance metrics. This is because recall and accuracy does not reflect the number of false positives in the prediction results, thus a predictor that makes 100% positive prediction will have the highest recall but has little value. We adjust the co-visitation time window τ from 15 mins to 60 mins and show its impact on these metrics. For each experiment run, we do a 3-fold cross-validation on the dataset and report the average performance. The result of symmetric social connection prediction is showed in Table 1 and that of one-way social connection prediction is showed in Table 2. Note that Random and Matrix Factorization are not affected by τ since they do not rely on co-visitations.

The proposed techniques demonstrate clear advantage in all experiments over the naive count-based and random scheme. It also outperforms Matrix Factorization based scheme in most setting. To summarize, our techniques have the best performance given that an appropriate τ value is selected to accurately capture co-visitations. In our experiments, when τ is either too small or too large, it will cause the performance to drop.

We observed that some users have never co-visited any location with some of his friends, which indicate it is impossible for any model to predict such social connections by only looking at their check-ins. Nevertheless, our experiments confirms that users' social connections are, to some extent, reflected by the occurrence of co-visitations. We find it is possible to predict some social connections with spatio-temporal data, but not all of them. This result is in accordance with our common-sense that people do visit certain locations with their friends but not all the time. However, it is not clear what is the limit of the predictive power of co-visitations in this context, which is worth further exploration.

6 CONCLUSION

Human mobility pattern is a long standing research topic. In this paper, we study the predictive power of user trajectories in estimating their social connections. Based on the hypothesis that friends tend to visit same locations at same time, we propose to model the probability that social connection exists between two users using their co-visitations. We design a comprehensive model that takes into consideration that co-visitations occur at different locations and at different time slots may have different predictive power. Using a selected subset of users in the Foursquare dataset, our experiments reveal that it is possible to predict some, but not all, social connections between LBSN users. The gain a deeper understanding of the problem and its inherent hardness, we plan to explore other predictive methods on large scale and more diversified datasets in our future work.

ACKNOWLEDGMENTS

This research has been supported by National Science Foundation AitF grant CCF-1637541.

REFERENCES

[1] Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 635–644.
[2] 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 conference on knowledge discovery and data mining*. ACM, 1082–1090.
[3] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3 (2010), 75–174.
[4] Huiji Gao, Jiliang Tang, and Huan Liu. 2012. Mobile location prediction in spatio-temporal context. In *Nokia mobile data challenge workshop*, Vol. 41. 44.
[5] Neil Zhenqiang Gong, Ameet Talwalkar, Lester Mackey, Ling Huang, Eui Chul Richard Shin, Emil Stefanov, Elaine Runting Shi, and Dawn Song. 2014. Joint link prediction and attribute inference using a social-attribute network. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 2 (2014), 27.
[6] Yehuda Koren, Robert Bell, Chris Volinsky, and others. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
[7] Defu Lian, Vincent W Zheng, and Xing Xie. 2013. Collaborative filtering meets next check-in location prediction. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 231–232.
[8] Aditya Krishna Menon and Charles Elkan. 2011. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 437–452.
[9] Kurt Miller, Michael I Jordan, and Thomas L Griffiths. 2009. Nonparametric latent feature models for link prediction. In *Advances in neural information processing systems*. 1276–1284.
[10] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. Mining user mobility features for next place prediction in location-based services. In *Data mining (ICDM), IEEE 12th international conference on*. IEEE, 1038–1043.
[11] Ben Taskar Ming-Fai Wong Pieter and Abbeel Daphne Koller. 2003. Link prediction in relational data. (2003).
[12] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T Campbell. 2011. NextPlace: a spatio-temporal prediction framework for pervasive systems. In *International Conference on Pervasive Computing*. Springer, 152–169.
[13] Michael Weiler, Klaus Arthur Schmid, Nikos Mamoulis, and Matthias Renz. 2015. Geo-Social Co-location Mining. In *Second International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data*. ACM, 19–24.
[14] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. 2010. Finding similar users using category-based location history. In *ACM SIGSPATIAL*. 442–445.
[15] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *Journal of Network and Computer Applications* 55 (2015), 170–180.
[16] Guolei Yang and Andreas Züfle. 2016. Spatio-Temporal Site Recommendation. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 1173–1178.
[17] Jihang Ye, Zhe Zhu, and Hong Cheng. 2013. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 171–179.
[18] Hongzhi Yin, Zhiting Hu, Xiaofang Zhou, Hao Wang, Kai Zheng, Quoc Viet Hung Nguyen, and Shazia Sadiq. 2016. Discovering interpretable geo-social communities for user behavior prediction. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 942–953.
[19] Zhijun Yin, Manish Gupta, Tim Weninger, and Jiawei Han. 2010. Linkrec: a unified framework for link recommendation with user attributes and graph structure. In *Proceedings of the 19th international conference on World wide web*. ACM, 1211–1212.
[20] Kai Yu, Wei Chu, Shipeng Yu, Volker Tresp, and Zhao Xu. 2006. Stochastic relational models for discriminative link prediction. In *NIPS*. 1553–1560.