IET Research Journals

Journals
The Institution of Engineering and Technology

Friendship prediction model based on factor graphs integrating geographical location

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Liang Chen[1], Shaojie Qiao[1*], Nan Han[2*], Chang-an Yuan[3,4], Xuejiang Song[5], Ping Huang[2], Yueqiang Xiao[2]

[1] School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China.
[2] School of Management, Chengdu University of Information Technology, Chengdu 610103, China.
[3] School of Computer and Information Engineering, Nanning Normal University, Nanning Guangxi 530299, China
[4] Guangxi College of Education, Nanning Guangxi 530007, China
[5] Chengdu Tanmer Technology Co., Ltd, Chengdu 610000, China.
* E-mail: sjqiao@cuit.edu.cn, hannan@cuit.edu.cn

Abstract:
With the development of network services and location-based systems, many mobile applications begin to use users' geographical location to provide better services. In terms of social networks, geographical location is actively shared by users. In some applications with recommendation services, before the geographical location recommendation is provided, we have to obtain user's permission. This kind of social network integrated with geographical location information is called location-based social networks (abbreviate for LBSN). In the LBSN, each user has location information when he or she checked in hotels or feature spots. Based on this information, we can identify user's trajectory of movement behavior and activity patterns. In general, if there is friendship between two users, their trajectories in reality are likely to be similar. In this study, according to user's geographical location information over a period of time, we explore whether there exists friend relationship between two users based on trajectory similarity and the structure theory of graphs. In particular, we propose a new factor function and a factor graph model based on user's geographical location to predict the friendship between two users in the real LBSN networks.

Keywords: friendship prediction, geographical location, trajectory similarity, factor graph, LBSN

1 Introduction

The rapid development of the Internet in recent years has promoted the emergence of various location-based services, which provide users with more personalized recommendation services based on their geographical location information, such as food delivery, taxi hailing, travel, etc. More location-based services begin to request users' location information directly or indirectly to improve their experience. Private data protection [1] should not be neglected and become an important problem of personal privacy security. There are some research work [2, 3] relevant on private data protection and some of these techniques have been applied to real-world systems. The privacy security of mobile devices is related to the interests or habits of each end user, and the approaches proposed in [4, 5] on the security of private information have been paid more attention in recent years. The information shared by these mobile devices facilitates a plenty of social research. In social networks, users often share logs or photos having location information in their social communities, and the friends sharing their everyday activities are more likely to be in the same position [6–8], that is to say, everyday interactions between friends make their activity area have intersection which will partially reflect the correlation of their locations, i.e., trajectory similarity [9, 10]. If we can discover the connections of friends from their location information, then we can improve the accuracy of existing link prediction algorithms[11], in order to enhance the performance of recommendation systems.

Currently, several studies based on LBSN are applied to recommendation system, among which the research on friend suggestion system [7, 12] often clusters users' home, work, restaurants and other central locations according to their location information and check-in records, which aims to calculate the similarity of check-in locations between two users. In addition, the author described the types of locations via information entropy [13], and then treated the intersection of locations w.r.t. two users as the similarity between users. Li [14] employed the method of multi-layer network combination to combine more information into the network to build a friend model. Other state-of-the-art recommendation [15–17] models do not explore the connection between users' location information. For example, the model proposed by [15] combines users' historical location and current location for recommendation, which is helpful for improving user's experience. Most link prediction methods focus on the importance of location to visitors, ignoring the strength [18, 19] of the relationships between those visitors. The drawbacks of these approaches lie in that: they are lack of extensibility, and each approach does work in a specific area. In addition, relevant research generally retrieve features [12, 20, 21] from geographical location information without taking into consideration the correlation between location information. Since different networks have different characteristics, we need to find the connections of users' geographical locations in the LSBN networks, and this connection is also applicable to most LBSN networks, that is to say, the model established based on connections between users is scalable in the LBSN network.

Factor graph model is a probability graph model, which plays a very important role in link prediction. Tang [22] and Cen [23] proposed a partiallylabeled pairwise factor graph model, where the relation prediction method does not only obtain good performance, but also has good scalability. But, for LBSN networks, the geographical location information shows the similar behavior of users. In this study, the relationship between users will be extracted to build as a factor function, and we design a factor graph model to predict whether there is friend relationship between two users.

Original contributions. The main contribution of this study is that, we propose a friend relationship learning and prediction model based on geographical location information and factor graph model in LBSN networks. In the proposed model, the geographic location information contained in these social networks is retrieved from user's trajectory data in LBSN, and the factor function is established based on the similarity of trajectories to learn these features. In addition, we use two real data sets in experiments, i.e., Brightkite and Gowalla, and the results show that our proposed model outperform the state-of-the-art classification methods.

The rest of the paper is organised as follows. Section 2 introduces the problem statement and the graph theory. Section 3 presents the calculation method, the definition of trajectory similarity and the analysis of trajectory similarity in the factor graph with multiple correlation. Section 4 gives the theoretical fundamentals of factor graph and the learning and prediction phases in the factor graph. Section 5 shows the experimental results of the proposed model by comparing it with other methods. Lastly, Section 6 concludes this paper and discusses the relationship prediction approach in machine learning.

2 Problem formulations

Generally speaking, we define a user in the social network as a node v in the graph, and the relationship between users is defined as an edge e in the graph, where $e \in v \times v$. Therefore, a social network is described by $G = (V, E)$, where V and E represent the set of nodes and edges in the network, respectively. In addition to these two basic components, different heterogeneous networks include other additional information. For example, there are many unlabeled nodes E^U in social networks, and each node v has a different parameter x. Based on the aforementioned concepts, we give the definition of social networks.

Definition 1. *[Partially labeled attribute location based networks] In this network, only partial nodes are labeled, and each node contains five-tuple attribute information, the network is denoted by* $G = (V, E^U, E^L, R^L, C, X)$*, where* E^L *represent a tagged edge set which is associated with* R^L*,* $E = E^L \cup E^U$*,* C *represents the location information retrieved from users' check-ins, and* X *is a property matrix associated with the set of users* V*, in which each row corresponds to a user, and each column is an attribute, one of the elements* x_{id} *in* X *denotes the* d^{th} *attribute of user* v_i*.*

From the above definition of a graph, we can further formulate the problem. For predicting the friendship of users in location-based social networks, given a partially labeled attribute network, the prediction of friendship in the network can be defined by the following function:

$$f : G = \left(V, E^L, E^U, R^L, C, X\right) \rightarrow Y \tag{1}$$

where Y is the output set of friendship which is predicted by the proposed model, and we can predict the tag type y_i of all E^U.

Presently, most of the researches on relationship mining in location-based social networks aims to collect more features and improve the classification accuracy by proving that these information is more effective and valid. However, most of existing approaches do not have good expansibility and cannot be applied to location-based social networks.

The check-in information of users is uploaded over a period of time and the users' location information is very limited, for example, 1,145 users uploaded less than five location information in Brightkite, while the complete information is 221. Therefore, in order to balance the number of trajectories between users, all uploaded location information is grouped by day, and then partitioned into time slices. The time of one day from 0:00 AM to 24:00 PM is divided into η time periods for location merging.

Definition 2. *[Users daily activity trajectory] In order to distinguish between weekdays and weekends, these two kinds of trajectories are collceted respectively. The definition is given as follows:*

$$Tr^i_{type} = \{L_1, L_2, ..., L_\eta\}, i \in V, type \in \{work, week\} \tag{2}$$

3 Trajectory similarity measurement and multivariate correlation analysis

There are many geographical correlations between users, such as the distance [24] between home, work, restaurant and so on. However, trajectory similarity [25] can best reflect the relationship between users, because the activity trajectories of users with a close relationship will affect each other, and their activities have similarity, including working, entertainment and eating. The similarity of the trajectories of user's social activities was high between two users who are friends. Then, we show how trajectory similarity is measured, and then explore the distribution of binary and ternary similarity.

3.1 Trajectory similarity measurement

Each person has his or her own activity trajectory every day, and there are certain similarities between people who are close to each other [26]. Therefore, the measure of similarity is of great help in determining the relationship between two persons. The trajectory measurement approaches can be divided into several categories, such as common point-based measurement methods EDR [27], LCSS, DTW, etc. In the shape-based method, Frechet distance [28] is often applied. In the point-based measurement method, EDR does not only consider the influence of noise, but also the common substring. For the activity trajectories of two users, when the distance of users w.r.t. a point is less than a threshold, we can regard this point as a point in a mutual sub-trajectory, which is a similar trajectory point.

In regard of the LSBN data, Gowalla and Brightkite have thousands of check-in records of users and it is time consuming to calculate the similarity of trajectories. In terms of trajectory modeling, Mazumdar [29] proposed a method to use entropy matrix to model the user's historical data. Generally speaking, the activity trajectory of weekday users is mostly the same, while the trajectory of weekend users are often different. Therefore, before measuring similarity, we need to retrieve user's trajectories. The weekday trajectory is the general activity track, denoted by Tr_{work}, and the weekend track is expressed by Tr_{week}. In addition, noise [30] may appear in user's trajectory, which shows a big bias in latitude and longitude. So, in this study, the data satisfying $d(x_i, x_{mean}) > \omega$ are removed. In the phase of trajectory sampling, a position mean value in a certain interval is viewed as the representative point during this period. It is worthwhile to note that in a certain period of time, the user's behavior is mostly the same. For example, before 8 AM, the user is likely to be at home, from 8 AM to 12 PM and from 14 PM to 18 PM, the user is likely to be at work. From 12 PM to 14 PM and from 19 PM to 24 PM, and the user is likely to be in a restaurant or outdoors. Based on the above discussion, we should take into full consideration these factors in the phase of trajectory sampling. After both trajectories of two users are obtained, the similarity of their trajectories can be calculated based on the EDR (Edit Distance on Real sequence) similarity algorithm given below.

Definition 3. *[Edit Distance on Real sequence(EDR)] Given two trajectory sequence of moving objects* $Q = \{q_1, q_2, ..., q_m\}$ *and* $R = \{r_1, r_2, ..., r_n\}$*,* $Sim(Q, R)$ *is used to recursively calculate whether each point in the sequence is similar to the others, it is defined as follows:*

$$Sim(Q, R) = \min \begin{cases} Sim(Rest(Q), Rest(R)) + subcost, \\ Sim(Rest(Q), R) + 1, \\ Sim(Q, Rest(R)) + 1 \end{cases} \tag{3}$$

where $m = 0$ *or* $n = 0$*,* $Sim(Q, R) = n$ *or* m*,* m *and* n *represent the lengths of the sequences* Q *and* R*, respectively,* $Rest(Q)$ *and* $Rest(Q)$ *indicate that pointers in the sequence* Q *and* R *move back one bit, i.e.,* $Rest(Q) = \{q_2, q_3, ..., q_m\}$*, and subcost is formalized as follows:*

$$subcost = \begin{cases} 0 & if\ Dist(Head(Q), Head(R)) < \epsilon \\ 1 & otherwise \end{cases} \tag{4}$$

where $Dist(Head(Q), Head(R))$ is the actual distance between the first point of Q and R. If $Dist(\cdot)$ is less than ϵ, we view it as 0. When we calculate the trajectory similarity of users, we will calculate the similarity of the two trajectories by the following equation:

$$\begin{aligned} Sim(Tr^i, Tr^j) = \min(&Sim(Tr^i_{work}, Tr^j_{work}), \\ &Sim(Tr^i_{week}, Tr^j_{week})), \end{aligned} \tag{5}$$

EDR can reduce noise points by quantifying distances to 0 and 1, and edit distance can improve the local time behavior, especially if local time-shifting is not a big deal. The EDR results may be biased when local time trends are large. In order to make the result more accurate, we can calculate the similarity after normalizing the trajectory.

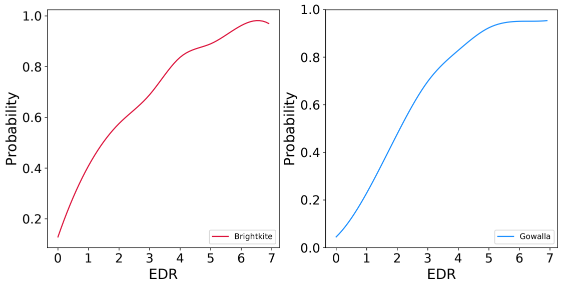

Fig. 1: Trajectory similarity of Brightkite and Gowalla, where the x-axis represents similarity calculated by Equation 5.

As shown in Fig. 1, in terms of two LBSN networks, with the improvement of trajectory similarity, the probability of friendship between these two users will also increase. However, in the actual case, the proportion after estimating the similarity of trajectories is greater than 4 is very small.

3.2 Multivariate correlation analysis

Here, we will introduce the binary and ternary associations [31] based on the trajectory similarity algorithm in Section 3.1, and analyze the similarity distribution under different relationship combinations.

In the network, we call the common connection of two edges of the same user as a binary relationship [31]. Another special structure is that three users form a triangle relationship, which is regarded to a basic ring. Because there are three relationships, a factor is often used to represent them in a factor graph, which is viewed as a ternary relationship. Different edges in these combinations may have different similarity, so we can statistically analyze the distribution under different relationships and different combinations of similarities. From the distribution of similarity and friendship probability shows in Fig. 1, with the increase of similarity, the probability of friendship also increases significantly. Binary and Ternary relationship in a factor graph is given in Fig. 2. We use the functions $h(\cdot)$ and $g(\cdot)$ to represent the factor functions of binary and ternary correlations, and we treat the trajectory similarity as the measurement to establish the features under different relationship combinations.

As shown in the Fig. 3, the similarity distribution of the two edges w.r.t a random node is different from the distribution of friend nodes. In regard to the Brightkite data, the similarity distribution of the edges with a friend relationship aggregates mostly around 3, while the similarity distribution of the randomly combined edges is mostly around 1, having a difference of 2. For the Gowalla data, the similarity of friends is obviously higher, and the random edges also concentrates, with a gap of 3. In terms of binary relation, the similarity of two edges is used to calculate the difference, which can show the difference in similarity of two edges. In terms of ternary relation, the difference of three similarity combinations are calculated respectively, and their mean values are used to represent the feature.

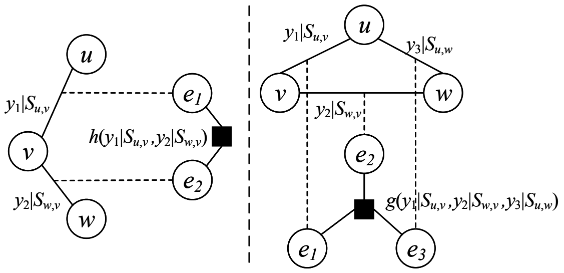

Fig. 2: Binary and Ternary relationship in a factor graph

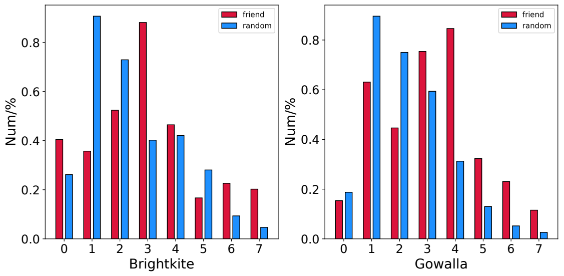

Fig. 3: Similarity of trajectory correlation. The x-axis represent different EDR. We randomly select nodes to calculate similarity of users' trajectories.

4 A new trajectory similarity factor graph model

The proposed model is based on the track similarity relationship of users in the network mined on LBSN, and the factor function is established and added into the factor graph model based on the geographical location characteristics. Before we input the original network into the model, we need to transform the original node-oriented network into an edge-oriented network. The nodes of the binary relationship in the original network are represented by a binary factor node. In addition, we need to add a triple factor function to the ternary relationship. In the proposed model, the factor functions used by binary and triadic factor nodes use the trajectory information of the adjacent nodes, while node v_i contains the tag information and attribute feature vectors. Then, we propose the global probability distribution of the factor graph model as follows:

$$\begin{aligned} p(Y|G) = &\frac{1}{Z}\prod_{e_i} f(y_i, x_i) \prod_{\wedge_{ij}} h\left(y_i|S_i, y_j|S_j\right) \\ &\prod_{\Delta_{ijk}} g\left(y_i|S_i, y_j|S_j, y_k|S_k\right), \end{aligned} \tag{6}$$

where $f(y_i, x_i)$ is the factor function associated with edges in the network. In a factor graph, each node is connected with an independent factor node. y_i in the function represents the type of a tag, and x_i is the attribute corresponding to the node, so the factor function represents the functional relation between the node feature and the tag. The function $h\left(y_i|S_i, y_j|S_j\right)$ represents the functional relation between trajectories of three users in a binary relationship, and only the similarity of two pairs of users is compared. The trajectories of users can be partitioned into weekdays and weekends. Additionally, the function $g\left(y_i|S_i, y_j|S_j, y_k|S_k\right)$ represents the relation between the trajectories of three users in a ternary relation, but we need to compare the trajectories of three pairs of users. In the factor graph, the total probability distribution can be figured out by the product of each factor function. In Equation 7, Z represents the normalized constant which is defined as follows:

$$Z = \sum_{Y} \prod_{e_i} f(y_i, x_i) \prod_{\wedge_{ij}} h(y_i|S_i, y_j|S_j) \prod_{\triangle_{ijk}} \\ g(y_i|S_i, y_j|S_j, y_k|S_k) \tag{7}$$

Equation 7 is used to calculate the normalized factor of the global distribution in a factor graph, which can be derived from the normalized factor of each function in the global distribution. These normalized factors are used to express the calculation results as a probability in the phase of probability calculation.

The definition of the factor function in the factor graph is very important. We define two different factor functions based on the similarity of trajectories. Here, we will give the definition of the factor function in detail.

The factor function is defined as follows. First, it is the factor function $f(\cdot)$ which is independent of node and represents the relation between the node attribute and the relation tag:

$$f(y_i, x_i) = \frac{1}{Z_\lambda} \exp\left\{\lambda^T \phi(y_i, x_i)\right\}, \tag{8}$$

where, Z is also a normalized constant, λ^T represents the parameter vector with the same dimension as x_i. The function $\phi(y_i, x_i)$ is an attribute vector function associated with the label y_i. In Equation 9, F represents the friendship label and S represents the stranger label (not friends). Equation 9 implies that the basic feature of a node is represented by a vector, which is used for the point product calculation with the parameter vector.

$$\phi(y_i, x_i) = (\mathbb{1}_{y_i=F} x_i, \mathbb{1}_{y_i=S} x_i)^T, \tag{9}$$

The factor function $h(\cdot)$ in the binary relation represents the relation between two adjacent nodes with real values having the trajectory similarity. There are three users in the binary relation, so there are three trajectories. Here, only the relation between these two similarity conditions and the label y of the node is considered, which are $y_i|S_i$ and $y_j|S_j$, respectively. According to the aforementioned trajectory similarity measurement function, the factor function can be formulated as follows:

$$h(y_i|S(i), y_j|S(j)) = \frac{1}{Z_\alpha} \exp\left\{\alpha^T \mathbf{h}(y_i|S(i), y_j|S(j))\right\} \tag{10}$$

where α^T is used to represent a parameter vector, and a new function $\mathbf{h}(\cdot)$ is used to obtain the new vector associated with the node label and trajectory similarity. After multiplying these two equal dimensions, a new function distribution is formed by using the power function e. As for the function $\mathbf{h}(\cdot)$, the detail is given as follows:

$$\mathbf{h}(y_i|S(i), y_j|S(j)) = \varphi(y_i, y_j) \cdot H(S(i), S(j))^T \tag{11}$$

where function $\varphi(\cdot)$ generate a vector for the combination of labels, so $\dim \varphi(\cdot) = \dim Y^2$. Notation $abs(S(i) - S(j))$ is taking the absolute value. $\varphi(\cdot)$ can defined as follows:

$$\varphi_{a,b}(y_i, y_j) = \begin{cases} 1 & y_i = Y^a, y_j = Y^b; \\ 0 & otherwise, \end{cases} \tag{12}$$

where a and b represent the labels of two nodes, which means that when nodes are labeled Y^a and Y^b with a valid value at the corresponding position of the vector. Equation 12 represents the characteristics of the node label combination.

$S(\cdot)$ in Equation 11 and the previous equation represents the trajectory similarity of the users on both sides, and $S(\cdot)$ is defined as follows:

$$S_{a,b}(i) = Sim\left(Tr^a, Tr^b\right) \tag{13}$$

where $Sim\left(Tr^a, Tr^b\right)$ is used to calculate the similarity of trajectories Tr^a and Tr^b. We define a threshold value ϵ, when the similarity is greater than ϵ, we consider them to be similar, and then we assign a valid value. Actually, the setting of this threshold will affect the experimental results. An appropriate value can be found by analyzing different algorithms through experiments.

Similar to the definition of the factor function of the binary relation, the definition of the ternary relation takes into account the third user's trajectory and the label of an newly added edge, so the dimension of the parameter vector is not the same as that of the binary relation. The detailed definition is given as follows:

$$\begin{aligned} g(\{y_v|S(v)\}) &= \frac{1}{Z_\alpha} exp\{\beta^T \mathbf{g}(\{y_v|S(v)\})\} \\ &= \frac{1}{Z_\alpha} exp\left\{\beta^T \left(\varsigma(\{y_v\})^T \cdot G(\{S(a)\})\right)\right\} \end{aligned} \tag{14}$$

where $v \in \{i, j, k\}$ and the function $G(\cdot)$ in the above equations are defined as follows:

$$H_s(\{S(v)\}) = \begin{cases} 1 & abs(S(i) - S(j)) = s; \\ 0 & otherwise, \end{cases} \tag{15}$$

$$G_s(\{S(v)\}) = \begin{cases} 1 & min(S(i), S(j), S(k)) = s; \\ 0 & otherwise, \end{cases} \tag{16}$$

Equation 15 and Equation 16 indicate the generation of features based on trajectories' similarity, which means that we set the constant value 1 at the corresponding position in the vector. These two factor functions represent the nonlinear feature representation of the similarity of the input. In reality, the number of parameters defined in a factor graph is directly related to the number of labels and the range of similarity.

Model learning. In the factor function, we define a parameter vector for each factor function, that is, (λ, α, β). In the phase of model learning, we need to learn these parameters, so here we use the maximized logarithmic similarity function to calculate the gradient of the parameters. For relationship nodes with labels,

$$\mathcal{O}(\lambda, \alpha, \beta) = \log p\left(Y^L|G\right) = \log \sum_{Y|Y^L} p(Y|G) \tag{17}$$

To facilitate understanding, we define the parameter as follows: $\theta = \{\lambda, \alpha, \beta\}$,

$$\begin{aligned} s(y_i) = (&\phi(y_i, x_i), \mathbf{h}(y_i|S(i), y_j|S(j)), \\ &\mathbf{g}(y_i|S(i), y_j|S(j), y_k|S(k)))^T \end{aligned} \tag{18}$$

So we can redefine joint probability of Equation 6 as follows:

$$\begin{aligned} p(Y|G) &= \frac{1}{Z} \prod_i \exp\left\{\theta^T s(y_i)\right\} = \frac{1}{Z} \exp\left\{\theta^T \sum_i s(y_i)\right\} \\ &= \frac{1}{Z} \exp\left\{\theta^T S\right\} \end{aligned} \tag{19}$$

Put Equation 19 into Equation 17 to obtain that:

$$\begin{aligned}\mathcal{O}(\theta) &= \log p\left(Y^L|G\right) = \log \sum_{Y|Y^L} \frac{1}{Z} \exp\left\{\theta^T S\right\} \\ &= \log \sum_{Y|Y^L} \exp\left\{\theta^T S\right\} - \log Z \\ &= \log \sum_{Y|Y^L} \exp\left\{\theta^T S\right\} - \log \sum_{Y} \exp\left\{\theta^T S\right\}\end{aligned} \tag{20}$$

So here we can use the gradient descent method to solve this function. Firstly, we need to take the partial derivative of this log-likelihood objective function. Here, we solve the parameter θ and the following equation can be obtained:

$$\begin{aligned}\frac{\partial \mathcal{O}(\theta)}{\partial \theta} &= \frac{\partial\left(\log \sum_{Y|Y^L} \exp\left\{\theta^T S\right\} - \log \sum_{Y} \exp\left\{\theta^T S\right\}\right)}{\partial \theta} \\ &= \frac{\sum_{Y|Y^L} \exp \theta^T S \cdot S}{\sum_{Y|Y^L} \exp \theta^T S} - \frac{\sum_{Y} \exp \theta^T S \cdot S}{\sum_{Y} \exp \theta^T S} \\ &= \mathbb{E}_{p_\theta(Y|Y^L,G)} S - \mathbb{E}_{p_\theta(Y,G)} S\end{aligned} \tag{21}$$

where $\mathbb{E}_{p_\theta(Y|Y^L,G)} S$ is the expectation if the graph is labeled, and $\mathbb{E}_{p_\theta(Y,G)} S$ is the expectation if the label is unknown. So, we need to calculate the global distribution of the factor graph with and without labels. The expectation given in Equation 21 is the key step to calculate each parameter's gradient in the learning process, so we need to calculate the probability distribution of each node with and without labels in order to figure out the expectation.

An efficient method for calculating the probability distributions in factor graphs is loopy belief propagation (LBP) [31]. In the phase of learning, LBP is used to calculate the probability distribution and marginal probability of $(Y|Y^L, G)$ with labels, and then (Y, G) without labels. The first propagation of the message is different in the above two cases. In the first round of calculation, the gradient can be fuzzy and the parameters can be uniformly initialized. When the message propagation in LBP runs after a finite number of iterations, the probability distribution tends to be stable. When the change of gradient becomes smaller and less than a threshold, the algorithm converge, then we can calculate the marginal distribution of each node.

Inferring Unlabeled Friend Relationships. In the learning process, after a certain number of iterations, the algorithm converges, the unlabeled node V can be predicted based on the parameters θ obtained in the phase of training according to the maximum and propagation algorithm by the following equation: $Y^* = \arg\max_{Y|Y^L} p(Y|G, \theta)$.

5 Experiment

5.1 Datasets

In this study, we use two real location-based services network data, i.e., Brightkite and Gowalla. These two data sets also include a large amount of check-in data besides the basic edge and node information. The description of these two data sets are given as follows:

Gowalla–the data set contains 196,591 nodes, 950,327 edges, and 6442,890 check-in data corresponding to each user.

Brightkite–the data set contains 58,228 nodes, 214,078 edges, and 449,1143 check-in data for each user.

We compared the predicted results with the edge provided in the data where two users are ground-truth friends. The negative samples in the data set are generated by using the random sampling method, and the actual connections are established in the network and labeled. In the phase of sampling, we try to balance the number of positive and negative samples, but in the real network, negative samples will not be labelled.

5.2 Comparison methods

Inferring the friendship relation can be regarded as a classification problem, so we use the commonly-used classification methods, such as SVM and LP [32]. In experiments, we extracted many topological features and geographical location features for classification. Topological features include common neighbors(CN), Degree, JC, PA, etc. Geographical location features mainly include distance and trajectory similarity of three representative locations (home, work and restaurant). As for the effectiveness of these special detection, the research [33] gives a detailed description of the link prediction on the LBSN network. Some of the attributes are given as follows:

Table 1 Summarization of the attributes used in the basic classification method and our model. where u and v represent nodes, and the neighbors of node u are represented by $\Gamma(u)$.

Attribute	Equation	Example
CN	$F_{u,v} = \Gamma(u) \cup \Gamma(v)$	CommonFriend_[$F_{u,v}$]
Degree	$D_u = count(\Gamma(u))$ $D_v = count(\Gamma(v))$	Degree_U_[D_u] Degree_U_[D_v]
JC	$J_{u,v} = \frac{\lvert\Gamma(u) \cap \Gamma(v)\rvert}{\lvert\Gamma(u) \cup \Gamma(v)\rvert}$	JaccardsCoefficient_[$J_{u,v}$]
PA	$p_{u,v} = \lvert\Gamma(u)\rvert \cdot \lvert\Gamma(v)\rvert$	PreferenceAttachment_[$P_{u,v}$]
TS	$S_{u,v} = Sim(Tr^u, Tr^v)$	TrajectorySimilarity_[$S_{u,v}$]
Dist(home)	$D^h_{u,v} = Dist(L^h_u, L^h_v)$	DistenceHome_[$D^h_{u,v}$]
Dist(work)	$D^w_{u,v} = Dist(L^w_u, L^w_v)$	DistenceHome_[$D^w_{u,v}$]

SVM, this is a supervised learning method. The data set is partitioned into the training set and testing set. SVM uses the attribute vector x_i of each relational label to train the model, and its decision boundary is the maximum-margin hyperplane to classify the learning samples. The learned parameters are used for quantitative classification. We implemented this algorithm by using the svm-light package. We mainly focused on the penalty factor C and γ in SVM. In the phase of training, we tested each parameter with the grid search method to determine the optimal parameter values. In addition, we used ten-fold cross validation to group the data sets for training to avoid overfitting.

LP, is a Semi-supervised learning. Label propagation (LP) [32] spreads labels based on proximity to the relation. Using the relation between samples, a complete graph model is established, which is suitable for undirected graph. Each node label is propagated to the adjacent node according to trajectory similarity. At each step of node propagation, each node updates its label according to the label of its adjacent node. In the phase of label propagation, keep the label of labeled data unchanged so that it can transmit the label to unlabeled data. Lastly, when the iteration terminates, the probability distribution of similar nodes tends to be similar and can be grouped into a class. LP does not require tuning parameters because the phase of label propagation is based on the network structure. In order to obtain the best classification results, the edge weight is specified according to the topological similarity and is used to distinguish the propagation priority.

The proposed method (TS-FGM), our proposed model on factor graphs includes binary and ternary factors. In addition, we combine the factor graph model with the common binary and ternary factors, which is called the multivariate correlation factor graph model (MC-FGM). By comparing the effectiveness of the two methods as factor functions, it is proved that the proposed similarity multivariate correlation can achieve better results in LBSN networks. In experiments, we only divided the data set into training set and testing set. We do not use the cross-validation method, because the data used in a factor graph is a complete network and the phase of calculating the probability distribution is based on the information transferred between nodes. The learning and prediction processes of these two methods are similar. In the phase of gradient descent, the proposed methods will predict the unknown labels in each iteration of calculation. The parameter gradient can be as small as possible after convergence. We use the method of dynamically changing the step size to make the models converge fast. The definition of the factor function of MC-FGM is similar to Equation 10, which is given below:

$$h(y_i, y_j) = \frac{1}{Z_\alpha} \exp\left\{\alpha^T \varphi(y_i, y_j)\right\} \tag{22}$$

$$g(\{y_v\}) = \frac{1}{Z_\alpha} exp\left\{\beta^T \left(\varsigma(\{y_v\})\right)\right\} \tag{23}$$

5.3 *Performance analysis*

5.3.1 Accuracy performance analysis: According to Table 2, the proposed TS-FGM method has a great improvement in the prediction accuracy by comparing with SVM, achieving around 24% improvement in Brightkite data set and 15% improvement in Gowalla data set, respectively. When compared with the LP method, the precision is improved by about 7%, and the prediction accuracy of positive and negative samples is also higher than that of LP. The method MC-FGM is a simplified version of TS-FGM, where the similarity of trajectories is not taken into account in feature extraction. In general, no more features are generated for multivariate correlation. In terms of the prediction performance, the TS-FGM method still improves the accuracy by about 5% compared with the MC-FGM method. In the Brightkite as well as the Gowalla datasets, the predicted performance of the Brightkite data was generally superior to that of Gowalla. According to the topological analysis of these two networks, the topology structure of Brightkite is more complex than that of Gowalla, so there are more multivariate correlations, e.g., ternary correlations. The best prediction accuracy value of our method in Gowalla reached to 88.75%. In contrast, the performance of SVM is the worst and LP was stable.

Table 2 Performance of friend prediction with different approaches (%)

Data set	Method	Precision	Recall	F1	Acc.
Brightkite	SVM	70.43	54.98	61.76	66.54
	LP	84.91	55.25	66.94	83.03
	MC-FGM	85.84	56.01	67.79	85.12
	TS-FGM	**91.53**	**56.88**	**70.16**	**93.65**
Gowalla	SVM	74.84	64.36	69.20	73.06
	LP	82.93	59.57	69.33	79.41
	MC-FGM	84.04	59.86	69.92	81.24
	TS-FGM	**87.35**	**62.08**	**72.57**	**88.78**

5.3.2 Factor contribution analysis: In the section, we will analyze the factor contribution and we analyze the predictive performance of the model by removing certain factors and combining some factor functions. As shown in table Table 3, we added three factor functions one by one to compare the prediction accuracy. We can see that the prediction accuracy is very low with only attribute feature factor functions, and the prediction performance is greatly improved by adding the binary correlation factors in both data sets, i.e., Brightkite(+6%) and Gowalla(+10%). According to the performance by adding ternary factors in these two data sets, the ternary correlation in Brightkite data greatly improved the prediction accuracy, making the prediction results reach to 93.65%, but the improvement in Gowalla was less obvious than that in binary correlation. In summary, the proposed similarity factor does play an important role in prediction.

Table 3 Contribution of different factor functions in prediction accuracy(%)

Data set	Factors used	Pre.	Rec.	F1	Acc.
Brightkite	Attributes	81.06	54.04	64.85	77.34
	+ TS|Binary cor.	85.11	55.25	66.94	83.03(+6%)
	+ TS|Ternary cor.	91.53	56.88	70.16	93.65(+10%)
Gowalla	Attributes	78.64	61.71	69.16	75.97
	+ TS|Binary cor.	85.37	62.35	72.06	85.70(+10%)
	+ TS|Ternary cor.	87.35	62.08	72.57	88.78(+3%)

5.3.3 Analysis of feature function: In terms of our proposed factor function, $H(\cdot)$ and $G(\cdot)$ are used to represent the similarity feature functions under binary and ternary correlations, respectively. Generally speaking, the feature function needs to express two input variables as a valid feature value. In our model, we used the definitions of these two feature functions that can achieve the best prediction results.

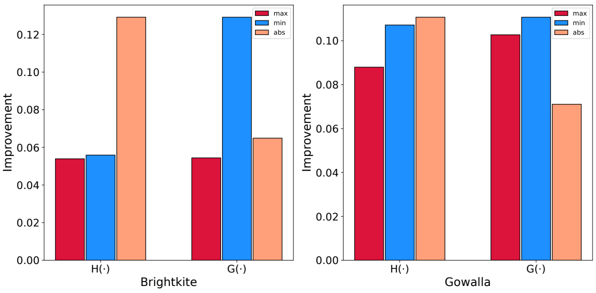

Fig. 4: Prediction performance of binary and ternary correlation factor functions on different datasets.

According to Fig. 4, the feature functions of the binary correlations performed the best for abs(absolute value of difference in $\{S_u, S_v\}$), where $S_i(i \in \{u, v\})$ represents the trajectory similarity between two users) in both Brightkite and Gowalla, and better for min(minimum value in $\{S_u, S_v\}$) than for max(maximum value in $\{S_u, S_v\}$) and abs in the ternary association. It is worthwhile to note that we can also use the Sigmod function to define the threshold to represent features if we do not consider the time complexity.

6 Conclusion

In this paper, we mainly studied how to extract users' geographical location connections and build a model to predict the friend relations in social networks based on the hidden information and factor graphs, and we conduct experiments on two real LBSN networks i.e., Brightkite and Gowalla. The cardinality of Gowalla data is five times that of Brightkite, so we used a sampling method for Gowalla to remove most nodes with no check-in information and the ones with very little information. Based on trajectory similarity, we studied the representation of binary and ternary network associations. Based on these preliminaries, we propose the TS-FGM model. According to the experimental results, our method is better than other classification algorithms in predicting accuracy. In terms of efficiency, Gowalla has a larger number of data, which is time consuming. In our future research, we will focus on reducing the time complexity of the message propagation process in the factor graph [23]. In addition, our experiment has also proved that the location information is indeed effective in improving the accuracy. So, if we can extract more effective location information, we can further improve the performance of the proposed algorithm.

7 Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 61772091, 61802035, 61962006, 61702058, 71701026), the Sichuan Science and Technology Program (Grant Nos. 2018JY0448, 2019YFG0106, 2019YFS0067), the Natural Science Foundation of Guangxi (Grant No. 2018GXNSFDA138005), a Project Supported by SiChuan Landscape and Recreation Research Center (Grant No. JGYQ2018010), the Innovative Research Team Construction Plan in Universities of Sichuan Province (Grant No. 18TD0027), Guangdong Province Key Laboratory of Popular High Performance Computers (Grant No. 2017B030314073), the Key R&D Program of Guangdong province (Grant No. 2018B030325002).

8 References

1 Obiri-Yeboah, J., Man, Q. 'Data security of android applications'. In: Proceedings of International Conference on Natural Computation & Fuzzy Systems & Knowledge Discovery. (IEEE), 2016. pp. 8–14

2 Fernandes, E., Rahmati, A., Jung, J., Prakash, A.: 'Security implications of permission models in smart-home application frameworks', *IEEE Security and Privacy*, 2017, **15**, (2), pp. 24–30

3 Barrera, D., Kayacik, H.G., van Oorschot, P.C., Somayaji, A. 'A methodology for empirical analysis of permission-based security models and its application to android'. In: Proceedings of the 17th ACM Conference on Computer and Communications Security. (New York, NY, USA: ACM), 2010. pp. 73–84

4 Chen, H., Li, w.: 'Mobile device user's privacy security assurance behavior: A technology threat avoidance perspective', *Information and Computer Security*, 2017, **25**, pp. 330–344

5 Papageorgiou, A., Strigkos, M., Politou, E., Alepis, E., Solanas, A., Patsakis, C.: 'Security and privacy analysis of mobile health applications: The alarming state of practice', *IEEE Access*, 2018, **6**, pp. 9390–9403

6 Abbasi, O., Alesheikh, A., Sharif, M.: 'Ranking the city: the role of location-based social media check-ins in collective human mobility prediction', *ISPRS International Journal of Geo-Information*, 2017, **6**, (5), pp. 136

7 Qiao, S., Han, N., Wang, J., Li, R., Gutierrez, L.A., Wu, X.: 'Predicting long-term trajectories of connected vehicles via the prefix-projection technique', *IEEE Transactions on Intelligent Transportation Systems*, 2018, **19**, (7), pp. 2305–2315

8 Qiao, S., Han, N., Zhou, J., Li, R., Jin, C., Gutierrez, L.A.: 'Socialmix: A familiarity-based and preference-aware location suggestion approach', *Engineering Applications of Artificial Intelligence*, 2018, **68**, pp. 192–204

9 Qiao, S., Shen, D., Wang, X., Han, N., Zhu, W.: 'A self-adaptive parameter selection trajectory prediction approach via hidden Markov models', *IEEE Transactions on Intelligent Transportation Systems*, 2015, **16**, (1), pp. 284–296

10 Qiao, S., Han, N., Zhu, W., Gutierrez, L.A.: 'TraPlan: an effective three-in-one trajectory-prediction model in transportation networks', *IEEE Transactions on Intelligent Transportation Systems*, 2015, **16**, (3), pp. 1188–1198

11 Qiao, S., Han, N., Gao, Y., Li, R.H., Huang, J., Guo, J., et al.: 'A fast parallel community discovery model on complex networks through approximate optimization', *IEEE Transactions on Knowledge and Data Engineering*, 2018, **30**, (9), pp. 1638–1651

12 Xu.Rui, G., Li, W., Wei.Li, W.: 'Using multi-features to recommend friends on location-based social networks', *Peer-to-Peer Networking and Applications*, 2017, **10**, (6), pp. 1323–1330

13 Bishop, C.M.: 'Pattern recognition and machine learning'. (springer, 2006)

14 Li, N., Chen, G. 'Multi-layered friendship modeling for location-based mobile social networks'. In: Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MOBIQUITOUS, Toronto, Canada, July 13-16. (IEEE), 2009. pp. 1–10

15 Bagci, H., Karagoz, P. 'Context-aware friend recommendation for location based social networks using random walk'. In: Proceedings of the 25th International Conference Companion on World Wide Web. (Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee), 2016. pp. 531–536

16 Backstrom, L., Leskovec, J. 'Supervised random walks: Predicting and recommending links in social networks'. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. WSDM'11. (New York, NY, USA: Association for Computing Machinery), 2011. pp. 635–644

17 Roth, M., Ben.David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., et al. 'Suggesting friends using the implicit social graph'. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. (ACM), 2010. pp. 233–242

18 Valverde.Rebaza, J., Roche, M., Poncelet, P., de Andrade.Lopes, A. 'Exploiting social and mobility patterns for friendship prediction in location-based social networks'. In: Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR). (IEEE), 2016. pp. 2526–2531

19 Qiao, S., Tang, C., Jin, H., Long, T., Dai, S., Ku, Y., et al.: 'PutMode: prediction of uncertain trajectories in moving objects databases', *Applied Intelligence*, 2010, **33**, (3), pp. 370–386

20 Crandall, D.J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., Kleinberg, J.: 'Inferring social ties from geographic coincidences', *Proceedings of the National Academy of Sciences*, 2010, **107**, (52), pp. 22436–22441

21 Scellato, S., Noulas, A., Mascolo, C. '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. (ACM), 2011. pp. 1046–1054

22 Tang, W., Zhuang, H., Tang, J. 'Learning to infer social ties in large networks'. In: Joint european conference on machine learning and knowledge discovery in databases. (Springer), 2011. pp. 381–397

23 Cen, Y., Zhang, J., Wang, G., Qian, Y., Meng, C., Dai, Z., et al.: 'Trust relationship prediction in alibaba e-commerce platform', *IEEE Transactions on Knowledge and Data Engineering*, 2020, **32**, (5), pp. 1024–1035

24 Cho, E., Myers, S.A., Leskovec, J. '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), 2011. pp. 1082–1090

25 Huo, Z., Meng, X., Hu, H., Huang, Y. 'You can walk alone: trajectory privacy-preserving through significant stays protection'. In: International conference on database systems for advanced applications. (Springer), 2012. pp. 351–366

26 Yin, H., Hu, Z., Zhou, X., Wang, H., Zheng, K., Nguyen, Q.V.H., et al. 'Discovering interpretable geo-social communities for user behavior prediction'. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE). (IEEE), 2016. pp. 942–953

27 Chen, Z., Shen, H.T., Zhou, X., Zheng, Y., Xie, X. 'Searching trajectories by locations: an efficiency study'. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. (ACM), 2010. pp. 255–266

28 Agarwal, P.K., Avraham, R.B., Kaplan, H., Sharir, M.: 'Computing the discrete fréchet distance in subquadratic time', *SIAM Journal on Computing*, 2014, **43**, (2), pp. 429–449

29 Mazumdar, P., Patra, B.K., Babu, K.S., Lock, R.: 'Hidden location prediction using check-in patterns in location-based social networks', *Knowledge and Information Systems*, 2018, **57**, (3), pp. 571–601

30 Liben.Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: 'Geographic routing in social networks', *Proceedings of the National Academy of Sciences*, 2005, **102**, (33), pp. 11623–11628

31 Newman, M.E.: 'The structure and function of complex networks', *SIAM review*, 2003, **45**, (2), pp. 167–256

32 Zhu, X., Ghahramani, Z., Lafferty, J.D. 'Semi-supervised learning using gaussian fields and harmonic functions'. In: Proceedings of the 20th International conference on Machine learning (ICML-03). (Washington DC), 2003. pp. 912–919

33 Bayrak, A.E., Polat, F.: 'Effective feature reduction for link prediction in location-based social networks', *Journal of Information Science*, 2019, **45**, (5), pp. 676–690