Geo-Friends Recommendation in GPS-Based Cyber-Physical Social Network

Xiao Yu, Ang Pan, Lu-An Tang, Zhenhui Li, Jiawei Han
Computer Science Department
University of Illinois, at Urbana Champaign
{xiaoyu1, angpan1, tang18, zli28, hanj}@illinois.edu

***Abstract*—The popularization of GPS-enabled mobile devices provides social network researchers a taste of cyber-physical social network in advance. Traditional link prediction methods are designed to find friends solely relying on social network information. With location and trajectory data available, we can generate more accurate and geographically related results, and help web-based social service users find more friends in the real world. Aiming to recommend geographically related friends in social network, a three-step statistical recommendation approach is proposed for GPS-enabled cyber-physical social network. By combining GPS information and social network structures, we build a pattern-based heterogeneous information network. Links inside this network reflect both people's geographical information, and their social relationships. Our approach estimates link relevance and finds promising geo-friends by employing a random walk process on the heterogeneous information network. Empirical studies from both synthetic datasets and real-life dataset demonstrate the power of merging GPS data and social graph structure, and suggest our method outperforms other methods for friends recommendation in GPS-based cyber-physical social network.**

I. Introduction

Popular social network services like Facebook and Twitter allow users to store and share both digital information, *e.g.*, web content, and also locations and trajectories collected from the real world. Analyzing these extra data from physical world can help us better understand people's daily activities, social areas and life patterns. Social network with data collected from sensors is usually refered as Cyber-Physical Social Network [3]. In this paper, we study friend recommendation problem in cyber-physical social network. With location and trajectory information available, we improve the accuracy of the results and make on-line social services much closer to users' real life.

One major difference between virtual web-based social network and real life social network is, *new* friends in real world tend to be geographically related. Geographical similarity is hiding in users' recently GPS data. To help web-based social network users find more friends in their real life, we define potential real life friends, who have both social similarities and geographical correlation as *Geo-Friends*, and denote *Geo-Friends Finding Problem* as real life friends discovery on web-based social network.

The reason why we want to isolate geo-friends from general web-based social network friends is intuitive. Geo-friends play an important role in off-line social events, *e.g.*, holiday party, football game, or book club. Geo-friends have a much higher probability to participate these real life events than other friends from virtual social network.

Following example demonstrate the idea of recommending geo-friends in web-based social network.

Example 1: Alex wants to find some new geo-friends to join him in a local charity event. There are three candidates: Bob who shares a large number of friends with Alex, but lives in another country; Carlos who works in the same company with Alex, but shares no similarity in terms of social network structure; David who shares couple of common friends, and also go to the same gym, same comic book store as Alex does every week.

After analyzing both social structure and recently collected GPS data, social network services should recommend David as Alex's geo-friends, since he has a higher probability to participate the local event with Alex.

Previous approaches of link prediction which usually only rely on social network structure would recommend Bob. But apparently, Bob is not a good candidate for Alex's social events, since he lives in another country. Also, solely relying on location or trajectory information for geo-friend finding does not work as well. Carlos who has a very high positive geographical correlation with Alex shares no social interests with Alex. Recommending Carlos to Alex is pointless as well.

In this paper, we propose a a three-step approach, named GEo-Friends Recommendation framework (a.k.a., GEFR). First, interesting and discriminative GPS patterns are extracted from a large amount of raw GPS data. Then we combines both geo-information and social network in a pattern-based heterogeneous information network. By applying random walk to reproduce friends making process on the network, we can effectively identify potential geo-friends for a specific user.

The contributions of this paper are summarized as follows:

- Propose geo-friends recommendation problem, and discuss the differences from previously studied link prediction problem.
- Define and generate a set of GPS patterns to describe people's real life social interaction and correlation.
- Propose a random walk-based statistical framework for geo-friend recommendation (GEFR).
- Design and conduct a series of experiments on both synthetic and real-world datasets. Demonstrate the power of GEFR in various situations.

II. Backgrounds and Preliminaries

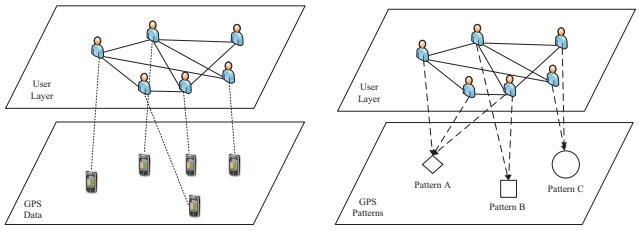
We briefly introduce the related data model, and define geo-friends finding problem in context.

A. Data Model

GPS data are continuous in both spacial and temporal dimensions. However, different devices have different data sample rate, which leads to shifting time intervals. Without losing generality, we assume constant sample rate within one application.

Definition 1 (GPS Trajectory): A GPS trajectory can be generated by sequentially connecting GPS records of a specific user following the ascending order of timestamps. We denote GPS trajectory for a person j as $S_j = \langle s_j^{(1)}, s_j^{(2)}, ..., s_j^{(n)} \rangle$, where $s_j^{(i)}$ is a GPS location record.

Definition 2 (GPS-CPN): A GPS-Based Cyber-Physical Social Network can be defined as $G(S, V, E)$, similarly to social network, V is the set of vertices, represents all the people in the network. E is the set of edges, represents all the links between people. S is the set of GPS trajectories, and each trajectory in S is associated with a specific person in V, represents this person's movements.



(a) Cyber Physical Social Network (b) Pattern-Based Heterogeneous Information Network

Fig. 1. GPS Networks

Notice that, people who carry GPS devices is a subset of vertices V in this network, so $|S| \leq |V|$. An example of GPS-CPN can be found in Figure 1(a).

B. Problem Definition

Intuitively, geo-friends recommendation is trying to find potential real life friends on web-based social network. With data model defined above, we formally define this problem as: Given a GPS-CPN $G(S, V, E)$, and a specific query person $v*$, one method should return a ranked list of people nodes in V and also for each element v' in the list, $\langle v', v* \rangle \notin E$. What's more, the ranking score in the process should consider both GPS trajectory S and social network (V, E).

III. Geo Friends Finding Framework

This section describes the three-step geo-friends recommendation framework (GEFR) in a given cyber-physical social network, including GPS pattern extraction, pattern-based heterogeneous information network building and random walk on the network. Details of the three-step approach will be presented in the following subsections.

A. GPS Pattern Extraction

Most GPS applications use raw GPS data directly, e.g., storage or visualization. However, raw GPS data are in huge size and hard to provide people with a semantic understanding of human behavior, In order to better understand the hidden information, we first present four different heuristics on geographically correlation based on empirical observations.

Common Location: GPS locations can reflect people's interests, and people tend to go to their interests related locations more often. If two people share common locations, which suggests they might share common interests, the probability that they become friends would be higher.

Common Routine: GPS trajectory segments indicate people's habits and routines. People who share similar routines, tend to become friends.

Meeting: If two people share same locations at the same timestamps in their GPS trajectory, they should be geographically related.

Hanging Out: Two people share same routine in a specific time period, which indicates they are hanging out in that time period. If two people hang out, the probability of they becoming geo-friends would be higher.

Based on above empirical observations and heuristics, we propose four different GPS patterns to capture these information. We first convert raw GPS trajectory dataset S to categorical dataset S_{cat}, and sequential dataset S_{seq} [1]. In S_{cat}, we simply discard temporal information and keep discretized locations in a unordered manner. While in S_{seq}, locations are still sequentially connected by the order of timestamps. With categorical dataset S_{cat}, sequential dataset S_{seq} and original GPS dataset S, we define GPS patterns as follow.

Definition 3 (FL-Pattern): Closed frequent patterns with $support \geq 2$ in S_{cat} is defined as Frequent Location Patterns (a.k.a., FL-Patterns), following **Common Location** heuristic.

Definition 4 (FT-Pattern): Closed sequential pattern with $support \geq 2$ and $length \geq 2$ in S_{seq} is defined as Frequent Trajectory Pattern (a.k.a., FT-Pattern), following **Common Routine** heuristic.

Definition 5 (FLT-Pattern): For each FL-Pattern, if locations share the same timestamp in all corresponding GPS trajectories, and no super-pattern with the same support can be generated by adding another time constrained location, this pattern can be defined as Frequent Location with Time Constraint Patterns (a.k.a., FLT-Patterns), following **Meeting** heuristic.

Definition 6 (FTT-Pattern): Similarly to FLT-Pattern, Frequent Trajectory with Time Constraint Pattern (a.k.a., FTT-Pattern) can be defined as closed sequential pattern with $support \geq 2$ and $length \geq 2$ in S_{seq} and it shares the same time period in corresponding GPS trajectories, following **Hanging Out** heuristic.

[1] GPS data discretization process are related to GPS error analysis and actual GPS coordinates. GPS datasets in the experiments of this paper have already been discretized and preprocessed. For detailed methods, please refer to [4] and [5].

To mine FL-Patterns, we apply FP-Growth [6] on S_{cat} and generate closed frequent patterns with $support \geq 2$. After FL-Pattern generation, there are two methods to generate FT-Patterns from S_{seq}. 1) Directly apply PrefixSpan [10] on S_{seq}, and extract all closed sequential frequent patterns, or 2) first calculate the permutation set of each FL-Pattern, generate combination set for each permutation, and then simply check each combination against S_{seq}, to make sure ascending timestamp order still hold. By collecting combinations in ascending timestamp order, FT-Pattern set could be generated as well. Method 2 could be more efficient if FL-Pattern set is small, and we used Method 2 in our experiments.

FLT-Patterns can be generated based on FL-Patterns. We first calculate the combination set for each FL-Pattern, and check the same timestamp constraint in each combination in the raw GPS dataset. By collecting combinations holding same timestamp constraint, FLT-Pattern set can be generated efficiently. Notice that, closed frequent pattern in S_{cat} are not the sufficient candidate set for FLT-Pattern mining. It is very likely that only partial FL-Pattern meet the same timestamp constraint, so it is necessary to calculate the combination set before check the time constraint. FTT-Patterns can be generated from FT-Patterns in a similar way.

B. Building Pattern-Based Information Network

By combining GPS patterns generated in last subsection, and the given social network, we can build a pattern-based heterogeneous information network $H(P, V, E')$ as follows. Given GPS-CPS $G(S, V, E)$, we first discard raw GPS trajectory set S. For each GPS pattern, we create an additional node p, and link corresponding person node v with p if this GPS pattern exists in person v's GPS trajectory history. And then create a new edge $\langle v, p \rangle$, and add it to E'. Notice that, edge set E' in heterogeneous information network contains three types of edges, which are edges between people, edges from person nodes to pattern nodes, as well as edges from pattern nodes to person nodes.

An example of GPS pattern-based heterogeneous information network is presented in Figure 1(b).

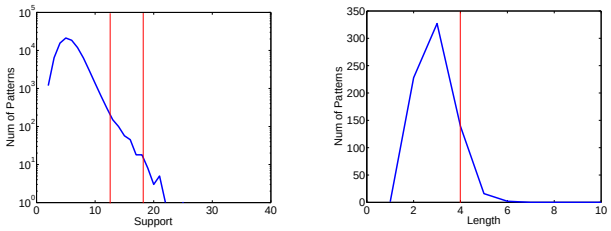
One needs to notice that, adding a large number of GPS patterns without selection, can decrease the performance of our method badly. This can be explained from different perspectives. 1) High frequency patterns usually indicate common public locations people can all related to, e.g., a bus station, or a hospital. These locations do not carry any discriminative GPS information, and they do not follow any heuristics we mentioned above. 2) The number of low length and support patterns are gigantic. Adding such patterns into the network will lead to a substantial increase of link search space for random walk process, which will reduce both the efficiency and precision.

Instead of manually refine patterns, we employ an entropy-based thresholding measure similar to [7] to filter out pattern with high or low support and length. Threshold calculation can be found in Equation 1.

$$argmax_i = -\sum_{j=0}^{j=i} p_j log(p_j) - \sum_{k=i+1}^{k=L} p_k log(p_k) \quad (1)$$

where $p_j = n_j / \sum_{a=0}^{a=i} n_a$, $p_k = n_k / \sum_{b=0}^{b=i} n_b$ and n is the pattern frequency.

We apply this measure to MIT Reality Mining dataset [2]. As presented in Figure 2, after calculation of entropy-based thresholds, we first select patterns in support histogram with threshold 13 and 18, and then filter out patterns with length lower than 4. This pattern selection strategy provides similar results as our manual parameter tuning experiments.



(a) Support Count Histogram (b) Length Count Histogram

Fig. 2. GPS Pattern Selection using Entropy Measure

After the construction of the heterogeneous information network, edge weights between nodes need to be defined before random walk process. By adding filtered pattern nodes into social network, edge set E' now contains three types of edges, including, edges from GPS pattern node p to person node v, edges from person node v to pattern node p, and also edges from person node v to person node v'. No edges are defined between GPS patterns. We first define edge weights from different types of GPS pattern nodes to person nodes. We use $w_(v, p)$ to represent raw edge weights, which will be normalized before random walk process.

$$w_{FL}(v, p) = \frac{\log(1 + length(p))}{|Nb_p(v)|} \cdot \alpha \quad (2)$$

$$w_{FT}(v, p) = \frac{\log(1 + length(p))}{|Nb_p(v)|} \cdot \beta \quad (3)$$

$$w_{FLT}(v, p) = \frac{\log(1 + length(p))}{|Nb_p(v)|} \cdot \gamma \quad (4)$$

$$w_{FTT}(v, p) = \frac{\log(1 + length(p) \cdot timespan(p))}{|Nb_p(v)|} \cdot \theta \quad (5)$$

where $Nb_p(v)$ denotes the set of pattern nodes connecting to person node v, $length(p)$ denotes the length of pattern p, and $timespan(p)$ denotes time span of a time constraint pattern p, in terms of number of timestamps. Parameter α, β, γ and θ controls pattern importance, and setting will be discussed in next section.

We define edge weights starting from pattern nodes to person nodes as follows.

$$w(p, v) = \frac{1}{|Nb_v(p)|} \tag{6}$$

where $Nb_v(p)$ denotes the set of person nodes connecting to pattern node p.

The third type of edge are from person nodes to person nodes, we define the edge weight from person node v to its neighbor v' as:

$$w(v, v') = \frac{1}{|Nb_v(v)|} \tag{7}$$

where $Nb_v(v)$ denotes the set of person nodes connected to person node v.

From above weight definitions for edges from person nodes to pattern nodes, one can notice that, edges weights from person nodes to pattern nodes are various based on pattern types, and pattern attributes, including $length$ and $timespan$. Parameters α, β, γ and θ are designed to adjust the importance of different types of pattern in different scenarios. Edge weights from pattern nodes to person nodes, and from person nodes to person nodes are solely based on number of neighbors. If number of people neighbor is smaller, the edge weight will be larger.

C. Random Walk on Heterogeneous Information Network

Random walk process on graph has been widely used in social network analysis [15] [11] [17]. To apply random walk on GPS pattern-based information network, we first need to define a transition probability matrix to describe all transition probability on the edge set of H. To represent all possible transitions on H, the size of the matrix should be $(|V| + |P|) \cdot (|V| + |P|)$. We sort all person nodes based on their ID in GPS trajectory dataset, and then append a sorted pattern nodes following the order they were extracted. By combining equations 2 to 7, we can generate a transition weight matrix $Pr^*_{(H)}$.

One thing we need to notice is that, by adding additional pattern attributes into edge weights definitions, including $length$ and $timespan$, patterns are more semantic but the sum of weights on out going edges of person nodes is no longer equals to 1, we need to normalize edge weights of pattern nodes in transition weight matrix $Pr^*_{(H)}$ to get transition probability matrix $Pr_{(H)}$ of the heterogeneous information network, following Equation 8.

$$pr(v, n) = \frac{w(v, n)}{\sum_{m \in Nb(v)} w(v, m)} \tag{8}$$

where n is a node connected to person node v, and Nbv denote all nodes connected to person node v. No normalization is required for pattern nodes, because the weight on out going edges of pattern nodes only depend on the number of person node neighbors, the sum of which is already 1. We denote normalized matrix as transition probability matrix $Pr_{(H)}$.

For ease of presentation, we simplify the representation of $Pr_{(H)}$ as

Algorithm 1: Geo Friends Recommendation Framework

Input: A GPS-based cyber-physical social network $G(S, V, E)$, where S is the GPS trajectory dataset, V is person node set, and E is edge set. A recommending target person $v*$. Number of recommended friends K. Parameters α, β, γ and θ for edge weights definitions, and λ for random walk process.

Output: A ranked list of recommended geo-friends

1, Generate *FL-Patterns*, *FT-Patterns*, *FLT-Patterns* and *FTT-Patterns* based on Definitions 4 to 7;

2, Filter out biased patterns and shrink link search space by using Equation 1;

3, Construct heterogeneous information network $H(P, N, E')$ by adding pattern nodes into GPS-based cyber-physical social network, link pattern nodes with corresponding person nodes, and define edge weiths following Equations from 2 to 7;

4, Generate transition probability matrix and normalize each column following Equation 8;

5, Iteratively update $R_N^{(t)}$ using Equations 10 or 11 regarding input query node $v*$ until it converges;

6, Rank link relevance based on the results from last step, and return the top-k nodes as the final results;

$$Pr_{(H)} = \begin{pmatrix} Pr_{(V)} & Pr_{(A)} \\ Pr_{(B)} & 0 \end{pmatrix} \tag{9}$$

where $Pr_{(V)}$ is an $|V| \times |V|$ matrix representing the transition probability between person nodes to person nodes, as defined in Equation 7. $Pr_{(V)}$ is not a symmetric matrix, since transition probability between two people nodes are defined based on number of neighbors of the starting node. $Pr_{(A)}$ is a $|P| \times |V|$ matrix representing the transition probability from GPS pattern nodes to person nodes, as defined in Equation 6. $Pr_{(B)}$ is a $|V| \times |P|$ matrix representing the transition probability from person nodes to GPS pattern nodes, as defined in Equation 2, 3, 4, 5 and 8.

We choose random walk with restart process to simulate geo-friends finding process, and estimate link relevance for a specific query person $v*$ in heterogeneous information network for several reasons. This statistical approach is stable and robust in different datasets, and also random walk with restart process satisfies some basic heuristics for identifying geo-friends.

First we present a list of GPS pattern utilizing heuristics, which random walk process can simulate:

(1) If a GPS pattern contains more geographical information or has a stronger semantic meaning, the in-coming probability from person nodes to this pattern should be higher, which

increases the probability from one person to another via this GPS pattern.

(2) If two people share more GSP patterns, the overall probability for one person link to another via these GPS pattern nodes would be higher.

(3) If one GPS pattern is rare, the out-going probability of this node would be larger, so that people connected to this pattern would have a higher probability to be linked together.

Then we discuss social network structure heuristics which random walk with restart process can simulate:

(1) Two people who share large number of common friends tend to be connected together, since the sum of transition probability between these two person nodes is higher.

(2) If person nodes are close to each other in the network in terms of number of hops, the probability of connecting them together is larger.

One can notice that, random walk with restart process satisfies all above heuristics; hence this method is a good fit for geo-friends recommendation in heterogeneous information network $H(P, E, V')$. We denote the query person as $v*$. The random walk process can be represented as:

$$R_N^{(t)} = (1-\lambda) Pr_{(H)} R_N^{(t-1)} + \lambda Pr_{(N)}^{(0)} \tag{10}$$

where R_N is a vector, that represents the link relevance from all the nodes in H to query person $v*$, and $R_N^{(t)}$ represents the link relevance of each node at the t^{th} iteration . We assign $P_n^{(0)}(v*) = 1$ where $v*$ is the query nodes, and all the other elements to 0. Similar to represent matrix $Pr_{(H)}$, we can re-write $R_N = [R_V R_P]^T$, based on Equation 10, we have:

$$\begin{aligned} & R_N^{(t)} \\ & = (1-\lambda) \begin{bmatrix} Pr_{(V)} \times R_{(V)}^{(t-1)} + Pr_{(A)} \times R_{(P)}^{(t-1)} \\ Pr_{(B)} \times R_{(V)}^{(t-1)} + 0 \end{bmatrix} + \lambda \begin{bmatrix} R_V^{(0)} \\ R_P^{(0)} \end{bmatrix} \end{aligned} \tag{11}$$

Based on Equation 11, we can iteratively update R_N until it converges. By ranking link relevance scores in $R_{(V)}$, top-k nodes can be generated easily. Overall framework can be found in Algorithm 1.

IV. Experiment

In this section, we test our methods against several competitor approaches on both synthetic and real datasets.

A. Experiment Setup

TABLE I
Datasets Summarization

Datasets	Graph Size ($\|V\|$, $\|E\|$)	# GPS Pattern	# Communities
gpsnet120	120, 435	108	6
gpsnet240	240, 1441	192	6
gpsnet600	600, 3583	576	12
gpsnet1200	1200, 18556	1170	15
mit reality	106, 88	FT: 49, BT:106	NA

1) Datasets: We generate 4 synthetic datasets with different sizes, attributes and distributions in order to cover different scenarios and thoroughly test our framework. Synthetic datasets are designed to cautiously simulate different types of relationships and communities, including, friends with high positive correlation in GPS histories, friends with different geographical social areas, neighbors who scatter in the whole social social network with high positive geographical correlation, friends within certain community, friends between communities, as well as a 10% noisy links adding to the whole social graph, etc.

Each synthetic dataset contains two parts, a social network, and a set of GPS history for each person, with 2-month timespan. The GPS datasets are designed by adding different types of GPS patterns on people's random movements.

We also attempt to apply our method on MIT Reality Mining dataset [2], which is collected from a project conducted from 2004 to 2005 in MIT Media Lab. The positioning method in this dataset is based on cellular tower estimation instead of GPS. Due to this limitation, we can not directly mine accurately mine GPS patterns as we defined in previous sections, instead, we accommodate this dataset by defining approximate FT-Pattern and FLT-Pattern. Approximate FP-Patterns in real dataset are generated by cellular tower IDs, while approximate FLT-Patterns are generated based on Blue-tooth history. Blue-tooth is actually a more reliable technique in terms of identifying and positioning other subjects within certain radius. Also, to match GPS history time span of synthetic datasets, we only use two month's GPS data (Jan 2005 and Feb 2005) in the real dataset for each subject in the network.

A more detailed summarization of all the datasets can be found in Table I. And also we present graph visualization results of synthetic dataset *gpsnet120* as well as the real dataset in Figure 3(a) and 3(e). From these figures, one can notice that, the social network structure of the real dataset is relatively sparse compared to the synthetic dataset. Although social network parts of synthetic datasets have been disarranged by adding certain amount of noisy edges, based on observation, we still can roughly find different communities on the overall network.

And also, we should mention that, to demonstrate the idea of the power of our framework *GEFR* on identifying geo-friends, major links in our synthetic datasets are geographically related.

2) Competitor Methods: To demonstrate the effectiveness of our results, we also build and apply a set of baseline methods on all the datasets. These baseline methods including popular method widely used in on-line social network services. These methods include:

Random: random selection. Same Edge: choose friends based on number of same friends. GPS Similarity: choose friends by measuring GPS location and trajectory similarity. Random Walk without GPS Patterns: Recommend friends by applying random walk with restart on the original social network.

For real dataset, we also add another baseline method named

Bluetooth. Bluetooth technique can capture people meeting frequency very accurately. So in this method, we recommend friends by returning people who share high meeting frequency. The recommended friend list by this method is sorted by the meeting frequency detected by bluetooth devices.

3) Evaluation: We use 4-fold cross validation method to finally evaluate performances of different methods. Since the problem we study in this paper is friend finding, the search space for this problem should be the overall link space.

To apply 4-fold cross validation on our datasets, we first need to randomly partition the overall link space into 4 subsets, with corresponding people nodes as well as GPS trajectory histories. And test each of the subset with all of the methods, while using the other 3 subsets as training dataset. For each test set, we sample 50 people nodes from the corresponding person node space, and use them as queries. To estimate the recommending results, if the recommended link exists in the testing set, we denote this recommendation as a hit, otherwise it would be denoted as a miss. By counting hits and misses, we calculate precision, recall as well as precision recall curve to finally estimate the overall performance of different methods on both synthetic and real dataset.

TABLE III
PERFORMANCE ON REAL DATASETS

Methods	P@1	P@5	P@10	R@10	R@20
Random	0.0132	0.0168	0.0168	0.0949	0.1760
SameEdge	0.0827	0.0659	0.0406	0.2513	0.3034
GPSSim	0.0985	0.0394	0.0288	0.2266	0.3542
BLT	0.1760	0.0880	0.0648	0.4520	0.6080
RWR	0.7419	0.1984	0.1056	0.7688	0.7809
GEFR	**0.7500**	**0.2016**	**0.1276**	**0.8011**	**0.8696**

B. Results

Precision and recall results of different methods can be found in Table II and III. *SameEdge* method performs relatively consistent on precision measures in all synthetic datasets, which suggests, by simply counting the same number of edges, *SameEdge* can capture partial link relevance information in graph structure. However, recall of *SameEdge* decreases with the graph size increasing (72.54% in *gpsnet120* vs. 38.99% in *gpsnet1200*). This suggests, *SameEdge* method works better in smaller and relatively sparse social network. When the social network grows larger, more sophisticated methods are needed to help users find friends. However, *SameEdge* performs poorly on the real dataset (Precision@1 is 8.27% and Recall@20 is 30.34%). Because the average number of edge in the real dataset is much lower than the synthetic datasets, graph structure is not able to carry enough link relevance information, which leads to a disappointing performance in this dataset.

GPSSim method in synthetic datasets gives bad performance if user only requires a small number of new friends, however, performance increases on both precision and recall if user queries more results. This observation reveals the common existence of neighbor relationship. Having a positive correlation in GPS trajectory histories is not a crucial friends identification criterion. However, in general, friends still share similar GPS trajectory than non-friends after all. This explains the increase performance in larger result sets. This method also performs badly in real dataset, however, as we mentioned before, users' locations in MIT Reality dataset are not directly captured by GPS device, instead they are estimated by cellular tower positions, which can be very inaccurate. This could be the major reason why *GPSSim* failed in real dataset.

While *SameEdge* and *GPSSim* perform poorly in real dataset. *Bluetooth* method provides acceptable results on both precision and recall measures. And also, Precision@1 of Bluetooh is much lower than Precision@1 of GEFR (17.6% vs. 75%), while Precision@10 of Bluetooh is much closer to Precision@10 of GEFR (6.48% vs. 12.76%). This again is a good evidence of existence of neighbor relationship.

RWR (Random Walk with Restart) method finds friends by using Random Walk with Restart process on social network to estimate link relevance. By simulating friends finding process on the graph, *RWR* gives a consistently good performance on all synthetic datasets and real dataset on both precision and recall. This indicates graph structure information plays an important role in friends recommendation process and we need more adaptive and suitable method like *RWR* to capture enough information.

Compared with the baseline methods, our method (*GEFR*) gives a significantly better performance on all synthetic datasets on both precision and recall measures. These results are much better than any method solely using only GPS information (*GPSSim*) or graph structure (*SameEdge* or *RWR*). This indicates, by combining GPS information and graph topological information and analyze in a heterogeneous information work fashion, *GEFR* can successfully extract discriminative patterns and useful information hidden in different data sources, and increase performance by mutually reinforce of different information. One might challenge that, Precisions at different positions of *GEFR* is somehow close to corresponding measures of *RWR*. We hereby studied the precision and recall curve between *GEFR* and *RWR* on synthetic dataset *gpsnet120*. As presented in Figure 3(d), on average, at the same recall level, *GEFR* surpasses *RWR* about 20% on precision. Specially, when recall is 90%, the precision of *GEFR* is around 60% while the precision of *RWR* is only 25%. This result provides good evidence when comparing performances of these two methods on synthetic dataset.

However, the precision and recall measures of these two methods on real dataset are very close. Similarly, we studied the precision and recall curve of these two methods, the differences of which are not as significant as in synthetic datasets, but *GEFR* is also better than *RWR*. One possible explanation is, as we mentioned before, locations in this datasets are represented by cellular tower IDs instead of actually GPS positions, so we can only mine approximate GPS patterns based on tower IDs and bluetooth data. Although we try to filter and select good patterns after mining, the overall quality of GPS patterns we are using in the information network

TABLE II
PERFORMANCE ON SYNTHETIC DATASETS

(a) Performance on *gpsnet120*

Methods	P@1	P@5	P@10	R@10	R@20
Random	0.0269	0.0245	0.0239	0.0888	0.1787
SameEdge	0.2569	0.1908	0.1451	0.5511	0.7254
GPSSim	0.0025	0.0905	0.1045	0.4918	0.7830
RWR	0.9248	0.3759	0.2023	0.9329	0.9636
GEFR	**0.9726**	**0.4139**	**0.2092**	**0.9682**	**0.9757**

(b) Performance on *gpsnet240*

Methods	P@1	P@5	P@10	R@10	R@20
Random	0.0199	0.0192	0.0187	0.0285	0.0579
SameEdge	0.3235	0.2078	0.1591	0.4026	0.5949
GPSSim	0.0000	0.1176	0.1468	0.4757	0.7992
RWR	0.9320	0.5084	0.2815	0.8828	0.9354
GEFR	**0.9701**	**0.5712**	**0.3089**	**0.9648**	**0.9707**

(c) Performance on *gpsnet600*

Methods	P@1	P@5	P@10	R@10	R@20
Random	0.0082	0.0074	0.0072	0.0091	0.0201
SameEdge	0.2535	0.1874	0.1530	0.3659	0.5570
GPSSim	0.0005	0.0686	0.0620	0.1941	0.4484
RWR	0.9418	0.5098	0.2819	0.8557	0.9117
GEFR	**0.9735**	**0.5756**	**0.3146**	**0.9553**	**0.9581**

(d) Performance on *gpsnet1200*

Methods	P@1	P@5	P@10	R@10	R@20
Random	0.0072	0.0070	0.0071	0.0048	0.0096
SameEdge	0.2562	0.2318	0.2124	0.2278	0.3899
GPSSim	0.0000	0.1413	0.1383	0.1819	0.3224
RWR	0.9437	0.8589	0.6112	0.8036	0.8817
GEFR	**0.9725**	**0.8733**	**0.6502**	**0.8603**	**0.9145**

might still be low because of the approximation. Results of this experiment are average of 5 successive runs in order to remove the random factor in both methods.

C. Parameter Setting

As we stated in Algorithm 1, user need to input parameters based on different scenarios. We here discuss the parameter setting in our experiments.

Parameter λ is the restart probability of the random walk process, which is a common parameter in all random walk based methods. For different tasks in social network analysis, this parameter should be set with different values. Based on previous study [15], link recommendation task, is more based on local social network structure instead of the whole social graph, which means we should constrain our search to a smaller range near the query person node. Yin, *et al.*, suggest a restart probability 0.9 gives the best result. To ensure our baseline method *RWR* can achieve its best performance, we set $\lambda = 0.9$ for both *RWR* and *GEFR*.

Parameters α, β, γ and θ control the importance of different GPS patterns in the friend finding process. As mentioned before, one can either use one parameter to interpret the overall importance of GPS patterns in this framework, or use four parameters for a more specific pattern importance control.

In our experiment, based on the conclusion we gained from the parameter tuning process, we set $\alpha = 1$ and $\beta = \gamma = \theta = 0.4$. In the synthetic datasets experiments, with the dataset size growing, average number of links per person node increases as well. If we correspondingly increase our GPS pattern weight parameter, performance will be increased. But this dataset size increasing factor affects other baselines as well, to play a fair game, we keep the same parameter setting for the whole synthetic experiments.

In the real data, we set parameter β for approximate FT-Pattern to 0.2 and parameter γ for bluetooth patterns to 0.4. This is also consistent with the analysis we mentioned before, approximate FT-Patterns are not very accurate and reliable while bluetooth technique are more trustworthy when detecting other subjects presence within certain radius.

V. RELATED WORK

The topics of cyber-physical social network analysis have received increasing attention in recent years. In this section, we briefly review related studies of GPS-based cyber-physical system, and link recommendation techniques.

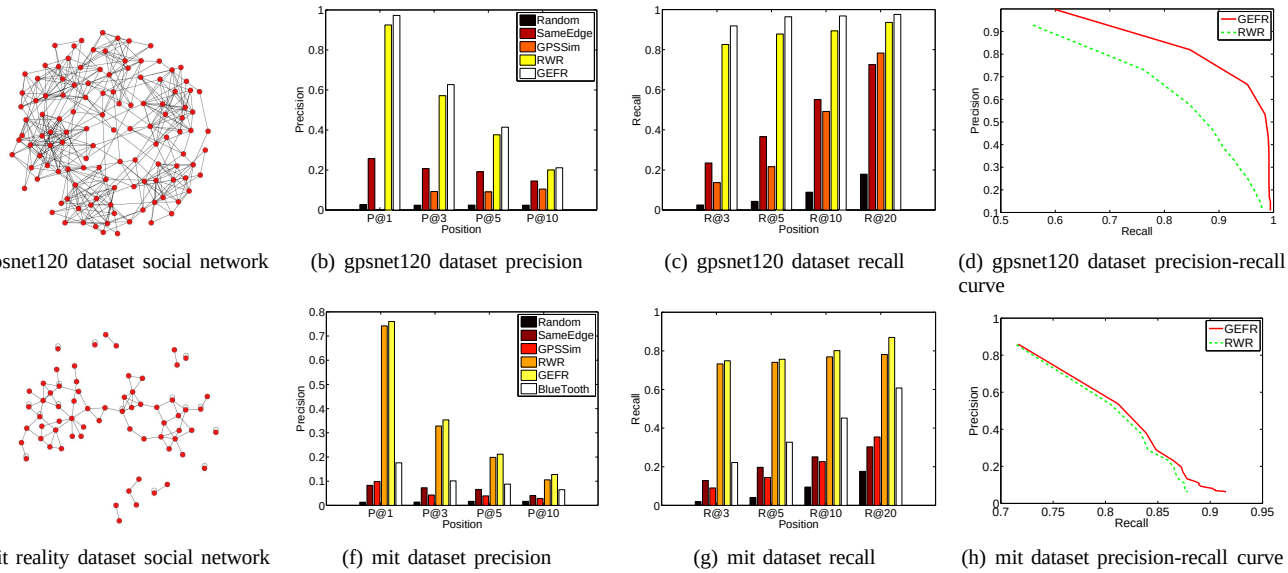
A. GPS-Based Cyber Physical Systems

Sha *et al.* give an introduction and survey on the development of cyber-physical system [12]. This survey provides application examples in the areas of energy grid, health care and transportation network. Microsoft SensorMap [13] is an early example of cyber-physical network, which allows users to browser the physical world in a digital map. However, the main objects in this application are physical objects, i.e., sensors, instead of people. Tang *et al.* study trustworthy issue in cyber-physical system [14], which is an important preprocessing step for reliable analysis.

Mining GPS location history is one of the most common and important jobs of GPS trajectory analysis. Different methods have been proposed on extracting locations from GPS history data [1] [16]. There works have different research focuses, including personalized mining, multiple user clustering, and semantic understanding, etc.

B. Link recommendation

Link recommendation or link prediction are important techniques in link analysis, which help specific users find more friends and also expand social network in terms of linkage. Most of the methods attempt to define a connection weight score between pairs of nodes in one way or another. Liben-Nowell, *et al.*, defined and studied link prediction method in [9], and also proposed methods to measure proximity of nodes in social network. Yin, *et al.*, structured this problem in an augmented graph fashion and applied random walk process on social network. The idea of third step of our framework is extended from Yin's work, while our information network is GPS pattern-based. Researchers also tried to approximately



(a) gpsnet120 dataset social network (b) gpsnet120 dataset precision (c) gpsnet120 dataset recall (d) gpsnet120 dataset precision-recall curve

(e) mit reality dataset social network (f) mit dataset precision (g) mit dataset recall (h) mit dataset precision-recall curve

Fig. 3. Performance Plots

estimate link relevance and correlation by applying probabilistic inference [8]. Inference based models are usually hard to interpret while training process is usually very time consuming and not scalable. All these methods are designed to find co-authorship or online friends, while none of them can detect geo-friends with compatibility of GPS data analysis.

VI. Conclusions

We propose the problem of identifying geographically related friends, and also a three-step statistical framework which combines geo-information with social analysis.

We first capture different types of GPS information by defining and generating four types of GPS patterns from GPS history data. Then, we build a pattern based heterogeneous information network, and defined transition probability matrix following GPS pattern definitions. By applying random walk process on this information network, link relevance between different nodes could be estimated, and potential geo-friends would be recommended to a specific query person.

Interesting future work includes, domain-oriented GPS pattern definitions, friends recommendation based on query person and specific interests, and also, real time friends recommendation by tracking user's GPS usage on the fly.

Acknowledgement

The work was supported in part by U.S. National Science Foundation grants CNS-0931975, IIS-0905215, the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and Boeing company. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] X. Cao, G. Cong, C. S. Jensen. Mining Significant Semantic Locations From GPS Data In *PVLDB*, 2010.
[2] N. Eagle, A. Pentland, and D. Lazer. Inferring Social Network Structure using Mobile Phone Data In *Proceedings of the National Academy of Sciences*, 106(36), pp. 15274-15278, 2009.
[3] R.K. Ganti, Y. Tsai, and T.F. Abdelzaher. SenseWorld: Towards Cyber-Physical Social Networks. In *International Conference on Information Processing in Sensor Networks*, pp.563-564, 22-24 April 2008.
[4] GPS Error Sources: http://www.kowoma.de/en/gps/errors.htm
[5] GPS Distance: http://en.wikipedia.org/wiki/Great-circle_distance
[6] J. Han, J. Pei, Y. Yin, R. Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. In *Data mining and knowledge discovery*, 2004.
[7] J.N. Kapur, P.K. Sahoo and A.K.C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram In *Computer Vision, Graphics, and Image Processing*, March 1985.
[8] H. Kashimaoki and N. Abe. A Parameterized Probabilistic Model of Network Evolution for Supervised Link Prediction In *Proc. of ICDM*, 2006.
[9] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. of CIKM*, 2003.
[10] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, M. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth In *17th International Conference on Data Engineering*, 2001.
[11] P. Pons and M. Latapy. Computing communities in large networks using random walks. In *Journal of Graph Algorithms and Applications*, 10(2):191–218, 2006.
[12] L. Sha, S. Gopalakrishnan, X. Liu and Q. Wang. Cyber-Physical Systems: A New Frontier. In *Machine Learning in Cyber Trust*, 2009.
[13] Microsoft SensorMap: http://tinyurl.com/6nqem2
[14] L. Tang, X. Yu, S. Kim, J. Han, C. Hung and W Peng. Tru-Alarm: Trustworthiness Analysis of Sensor Networks in Cyber-Physical System. In *Proc. ICDM*, 2010.
[15] Z. Yin, M. Gupta, T. Weninger, and J. Han. A Unified Framework for Link Recommendation Using Random Walks. In *Proc. ASONAM*, 2010.
[16] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from GPS trajectories. In *Proc. WWW*, 2009.
[17] Y. Zhou, H. Cheng and J. Yu. Graph clustering based on structural/attribute similarities In *Proc. VLDB Endow.*, 2009.