Exploiting Spatio-Temporal User Behaviors for User Linkage

Wei Chen  
School of Computer Science and Technology, Soochow University, China  
wchzhg@gmail.com

Hongzhi Yin*  
School of ITEE, The University of Queensland, Brisbane, Australia  
db.hongzhi@gmail.com

Weiqing Wang  
School of ITEE, The University of Queensland, Brisbane, Australia  
weiqingwang@uq.edu.au

Lei Zhao  
School of Computer Science and Technology, Soochow University, China  
zhaol@suda.edu.cn

Wen Hua  
School of ITEE, The University of Queensland, Brisbane, Australia  
w.hua@uq.edu.au

Xiaofang Zhou  
School of ITEE, The University of Queensland, Brisbane, Australia  
zxf@itee.uq.edu.au

ABSTRACT
Cross-device and cross-domain user linkage have been attracting a lot of attention recently. An important branch of the study is to achieve user linkage with spatio-temporal data generated by the ubiquitous GPS-enabled devices. The main task in this problem is twofold, i.e., how to extract the representative features of a user; how to measure the similarities between users with the extracted features. To tackle the problem, we propose a novel model STUL (Spatio-Temporal User Linkage) that consists of the following two components. 1) Extract users’ spatial features with a density based clustering method, and extract the users’ temporal features with the Gaussian Mixture Model. To link user pairs more precisely, we assign different weights to the extracted features, by lightening the common features and highlighting the discriminative features. 2) Propose novel approaches to measure the similarities between users based on the extracted features, and return the pair-wise users with similarity scores higher than a predefined threshold. We have conducted extensive experiments on three real-world datasets, and the results demonstrate the superiority of our proposed STUL over the state-of-the-art methods.

KEYWORDS
Cross-domain; User linkage; Spatio-temporal behaviors

1 INTRODUCTION
The proliferation of GPS-enabled devices and mobile techniques has led to the emergence of large amount of spatio-temporal information. For example, the vehicles equipped with GPS can generate lots of trajectories, which consist of a sequence of points that are sampled in a short time period, to keep track of moving objects. Meanwhile, the widespread of location based social networks, such as Facebook, Twitter, and Foursquare have generated massive discrete check-in data [20], as many users share their status associated with locations and timestamps. The availability of spatio-temporal information offers a good opportunity to model users’ spatio-temporal behaviors [23][18]. On the other hand, user linkage, which aims at connecting the same users across different platforms, has attracted much attention. User linkage benefits widespread real applications, such as prediction [13][21], data fusion [28], recommendation [19][22], etc. This paper focuses on leveraging the increasingly available spatio-temporal information in user linkage.

However, to the best of our knowledge, there is only one work utilizing the users’ spatial and temporal features simultaneously to achieve user linkage [14]. In that work, locations and times are divided into bins, and each spatio-temporal record is associated with a bin (r, t) where r is a region and t represents a time interval. The similarities between users are inferred based on users’ co-occurrences in each bin. Nonetheless, time and space are intrinsically continuous. Discretization of time and space inevitably leads to information loss, especially for the points near the boundaries. Assume that u0 is a user on platform A while u1 and u2 are two users on platform B. To simplify the problem, we assume that there is only one activity record v0, v1 and v2 for each user u0, u1 and u2 respectively. The distributions of these activity records in terms of space and time are given in Figure 1(a) and 1(b) respectively. Based on [14], u0 and u2 have a larger probability to be linked together, as they co-occur in both the spatial bin r1 and the temporal bin t1. However, compared with u2, u1 is more similar to u0 in terms of both spatial distribution in Figure 1(a) and temporal distribution in Figure 1(b). Thus, the discretization based method cannot capture the similarity between features that are divided into different bins. Besides, discretization of time and space always begs the question of selecting the region or time interval size, and the size is invariably too small for some regions and too large for others.

*This author is the corresponding author.

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.

CIKM’17, November 6–10, 2017, Singapore.
© 2017 ACM. ISBN 978-1-4503-4918-5/17/11 . . . $15.00
DOI: http://dx.doi.org/10.1145/3132847.3132898
In addition to the aforementioned problems, another major issue that needs to be addressed in user linkage is to prioritize the extracted spatio-temporal features. Riederer et al. treat each feature equally in [14]. However, we argue that these spatio-temporal features are not equally important in user linkage. In the task of user linkage, we seek to identify what are unique about a user, in order to distinguish him/her from others. In this process, the “uncommon”, which we refer to as discriminative, spatio-temporal features actually play a more important role. For example, some locations are very popular and frequently visited by many people, such as the downtown in the city and a popular restaurant, whereas other locations are more specific only to a few people, such as a private house. In a popular public place, it is more likely for two different users to co-locate by coincidence. Thus, such spatial features are less helpful to identify the same users.

To tackle the aforementioned problems, we propose a novel spatio-temporal user linkage model STUL to exploit the continuous spatio-temporal features associated with user activities, and compute their weights based on their discriminability. In this model, time and space are treated as continuous variables. A novel method extended from density-based clustering (DP) [15] is proposed to extract users’ spatial features. Compared with the bin-based method in [14], this approach is more tolerant to anomalous points and more likely to capture the real spatial behaviors of a user, as it treats the space in a continuous way and thus is able to extract regions with arbitrary shapes. To model users’ temporal behaviors, an approach based on Gaussian Mixture Model (GMM) is proposed in STUL. This approach aims at modeling users’ periodical behaviors centered on multiple time points.

Inspired by TF-IDF (Term Frequency Inverse Document Frequency), STUL develops a novel weight scheme to compute weights for the extracted spatio-temporal features, by lightening the common features with smaller weights and highlighting the discriminative features with larger weights. Based on the extracted features and associated weights, STUL develops three novel similarity measures and combines them in a unified way to compute the final similarities between users. The pair-wise users with similarity scores higher than a given threshold are returned as the linked user pairs.

To sum up, we make the following major contributions in this study.

- We propose a novel model STUL, where a DP-based method is developed to model the spatial behaviors, and an approach based on GMM is used to model the users’ temporal behaviors. STUL introduces a novel weight scheme to distinguish the discriminative features from the common features. Based on the extracted features and associated weights, STUL develops a novel similarity measure which combines three different similarity measures to compute the similarities between users.
- We conduct extensive experiments on real datasets, and the experiment results demonstrate the high performance of STUL.

The rest of paper is organized as follows. The related work is presented in Section 2. We formulate the problem in Section 3. The spatio-temporal features are extracted in Section 4. We measure the similarities between users in Section 5 and conduct experiments in Section 6, which is followed by the conclusion in Section 7.

2 RELATED WORK

2.1 User Linkage

The user linkage across different platforms was firstly proposed in [24], where the disconnected platforms were connected with the proposed approaches that leverage social medias collective patterns. Following that study, a novel method to identify users based on web profile matching has been proposed to aggregate online friends into a single integrated environment [17], which also extended its effectiveness by incorporating the user’s friend network. Next, the user’s profile that contains more abundant information was characterized to accomplish user linkage [2][11][25]. To merge user’s contacts from different social services or compose more complete social graph for many social-powered applications, a Conditional Random Fields based approach was designed for user profile matching [2]. Especially, the work is extremely suitable for cases when profile data is poor, incomplete or hidden due to privacy settings. Human behaviors with regard to the usages of online usernames was firstly proposed by [11] to tackle the problem of linking users across multiple online communities, where an alias-disambiguation step was proposed to differentiate users with the same usernames. Furthermore, Zafarani et al. [25] have introduced a behavior model to find a mapping among identities of individuals across social media sites with more features, such as usernames, language, and writing styles. Unlike the aforementioned work, [12] investigated the problem of large-scale social identity linkage across different social media platforms. The developed framework HYDRA consists of three components, i.e., user’s behaviors modeling, structural consistency graph construction, and mapping function learning. In the following studies, a novel energy-based model COSNET has been proposed to address the problem of potential inconsistency of matchings between multiple networks by considering both local and global consistency among multiple networks [27], where a subgradient algorithm was developed.
to accelerate the model training. In recent advance, Riede r et al. [14] investigated the problem of linking users with location data, where locations and times were divided into bins, and the scores of user pairs were measured based on these bins. Connecting users across multiple platforms brings opportunities for many applications and insights on users’ behaviors, yet it raises privacy concerns. To tackle the problem, [5] studied the problem of reliable profile matching by exploiting public attributes of users. To avoid cross-site identity linking attacks, novel methods were developed in [1].

2.2 Mining of Discriminative Features

Mining and highlighting the discriminative features have received great attention, which underly a wide range of applications [10][9][8][6]. The discriminative color features method, which applies an effective color model, a novel similarity measure, and effective color feature extraction methods, was proposed to improve face recognition performance [10], where the discriminative color features were extracted from a compact color image representation. Zhu et al. [30] proposed a diversified discriminative feature selection method for graph classification, where the discriminative score was used to select frequent subgraph features, and a new diversified discriminative score was proposed to select features that have a higher diversity. To overcome the problem of complicated data process in social network area, an unsupervised feature selection algorithm was proposed due to the unlabeled nature of datasets in social network [6]. The idea of [4] is similar to our study, i.e., highlighting the discriminative features with higher score, and lightening the common features with smaller score. However, since their methods were designed to characterize objects in heterogeneous information network, and each object was represented by a feature tree, these methods cannot be applied to our study.

In spite of the great contributions made by the aforementioned studies, none of them consider the discrimination and continuity of the spatio-temporal features while linking across domains users with these features. To tackle the problem, we propose the model STUL in this study.

3 PROBLEM FORMULATION

In spatio-temporal databases, there are two different and important types of data, i.e., the check-in data and the trajectory data.

Definition 1. User Activity. A user activity on both types of data is defined as \( d = (u, \text{lat}, \text{lng}, t) \), where \( u \) is the user id, \( \text{lat} \) and \( \text{lng} \) denote the latitude and longitude of the location where this activity is taken and \( t \) represents the time.

The check-in data is an important type of spatio-temporal data with low sample rate, where the time span between two locations is usually large. The check-in places associated with user activities in Twitter, Facebook, and Foursquare are meaningful (i.e., restaurants, hotel and so on). These meaningful check-ins are informative in modeling users’ spatio-temporal behaviors. Thus, following many existing work [9][23], check-in data are applied directly without any preprocess in STUL.

Trajectory data is another important type of spatio-temporal data. Different from the check-in data, trajectory data are densely sampled, which means that two adjacent points in a trajectory are sampled in a short time period. The dense sampling of trajectory data brings redundancy, where many points are meaningless, for example, a point in a high way. We argue that a meaningless point in trajectory data is not helpful in identifying users’ spatial-temporal behavior patterns. To remove the meaningless points, we only focus on the regions where a user visits repeatedly. To find these regions, we introduce stay point, which is an important notion in existing research work [29].

Definition 2. Stay Point [29]. Given a trajectory \( \tau = (p_1, p_2, \ldots, p_n) \), where \( p_i \) is in the form of \((\text{lat}, \text{lng}, t)\), \( \text{lat} \) and \( \text{lng} \) denote the latitude and longitude respectively, and \( t \) is the corresponding timestamp, a stay point \( s \) stands for a region where a user stayed over a certain time interval. Given a time threshold \( \delta_t \) and a distance threshold \( \delta_d \), if there exists a group of consecutive points \( P = (p_i, p_{i+1}, \ldots, p_j) \) of \( \tau \) such that \( \forall j < k \leq \), Distance\((p_i, p_j) \leq \delta_d \), and \( |p_j - t - p_i| \geq \delta_t \), then we have a stay point \( s \) in the form of \( (s, \text{lat}, s, \text{lng}) = (\sum_{i=1}^{j} p_i \text{lat}, \sum_{i=1}^{j} p_i \text{lng}) \).

The notion stay point is proposed to mine regions where an individual stayed longer than a threshold time period. Based on this notion, we introduce the points used to mine the meaningful regions based on which we build our model.

Definition 3. Stay Region Candidate Points. Given a trajectory \( \tau = (p_1, p_2, \ldots, p_n) \), the start point \( p_1 \), the end point \( p_n \), each point \( p_i \) of \( P = (p_i, p_{i+1}, \ldots, p_j), \forall j < k \leq j \), Distance\((p_i, p_j) \leq \delta_d \), and \( |p_j - t - p_i| \geq \delta_t \) is defined as stay region candidate point, denoted as \( r_c \).

We call the meaningful regions used to mine users’ spatio-temporal behaviors as “stay regions”. To remove the redundancy in trajectory data, only the points that can generate a stay point \( s \) are considered in modeling users’ spatio-temporal behaviors. These points are referred as “Stay Region Candidate Points”. Additionally, it is worth noting that the start point and end point of the trajectory are also significant. For instance, given a set of historical trajectories, we may find a user’s home region and work region directly from all start points and end points, as many users usually commute between their homes and companies. Aforementioned points are defined as stay region candidate points, since we can extract the stay regions of a user with them.

Problem 1. Given two sets of users \( U_1 = \{u_{11}, u_{12}, \ldots, u_{1n}\} \) and \( U_2 = \{u_{21}, u_{22}, \ldots, u_{2m}\} \) on two platforms respectively, where each user is associated with a set of user activities defined in Definition 1, we aim at finding linked user pairs across the two platforms.

Overview of our method. The overview of our method is presented in algorithm 1, which contains two components.
First, a density based algorithm is used to extract the stay region distribution, and the Gaussian Mixture Model is used to extract the temporal features, which contain the global time distribution and local time distribution. Additionally, we assign different weights to these spatio-temporal features. The details of this component are discussed in Section 4. Second, we develop novel methods to calculate the similarity between users, and add the user pair \( u_i, u_j \) into the result on condition that \( S(u_{i1}, u_{j2}) \geq \theta \), where \( \theta \) is a predefined threshold. The details of this component are discussed in Section 5.

Algorithm 1: Overall Algorithm

**Data:** two sets of users \( U_1 = \{u_{i1}, u_{i2}, \ldots, u_{i4}\} \) and \( U_2 = \{u_{j1}, u_{j2}, \ldots, u_{j4}\} \), a threshold \( \theta \)

**Result:** a set of user pairs

1. **for each user** \( u \) in \( U_1 \) and \( U_2 \) **do**
   - extract stay regions of \( u \) with a density based method, and assign different weights to these regions:
   - the global and local time distribution of \( u \) with GMM, and assign different weights to these time clusters;

2. **for each user** \( u_{i1} \) in \( U_1 \)
   - **for each user** \( u_{j2} \) in \( U_2 \) **do**
     - calculate the similarity \( S(u_{i1}, u_{j2}) \);
     - if \( S(u_{i1}, u_{j2}) \geq \theta \), add the user pair \( u_{i1}, u_{j2} \) into the result;

4 FEATURE EXTRACTION

To model users’ spatial behaviors, we extract the stay regions. To model users’ temporal behaviors, we extract the time distribution that includes global time distribution and time distribution in each stay region (i.e., local time distribution). Note that, we need to first extract all stay region candidate points if the input data consists of trajectories. We directly extract the spatio-temporal features if the input data consists of check-in records.

4.1 Stay Region Distribution

Stay regions are areas that contain more points than others, and a user will visit repeatedly. To extract these regions, we use an advanced density-based clustering (DP) method [15][3], which is based on the idea that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. This method has the following advantages: 1) DP has been proved to be able to ignore anomalous points. 2) DP is able to extract regions with arbitrary shapes. 3) Compared with other clustering algorithms, DP requires less parameters, i.e., calculating the local density and the distance from points with higher local density for each stay region candidate point. Given a set of stay region candidate points \( \{r_1^c, r_2^c, \ldots, r_n^c\} \), the local density \( p_{r^c} \) and the distance from points with higher local density \( \delta_{r^c} \) are defined as:

\[
p_{r^c} = \sum_j \chi(d_{r^c,j} - d_c), \begin{cases} 
\chi(x) = 1, & \text{if } x < 0 \\
\chi(x) = 0, & \text{otherwise}
\end{cases}
\]

\[
\delta_{r^c} = \begin{cases} 
\min \{p_{r^c} \delta_{r^c}^c\}, & \text{if } p_{r^c} > p_{r^c}^c \\
\max \{d_{r^c,j} - d_c\}, & \text{otherwise}
\end{cases}
\]

where \( d_c \) is the cutoff distance given by users. \( p_{r^c} \) denotes the number of points that are closer to the candidate point \( r^c \) than \( d_c \). \( \delta_{r^c} \) is the minimum distance between \( r^c \) and any other point with higher density, especially, \( \delta_{r^c} \) is measured as the maximum distance \( d_{r^c,j} - d_c \) if \( r^c \) has the highest density. Following the calculation of \( p_{r^c} \) and \( \delta_{r^c} \), the top-k centers with the highest value \( \xi_{r^c} = p_{r^c} \times \delta_{r^c} \) are returned as cluster centers, and we assign different labels to these centers. Then, we assign each remaining point to the same cluster as its nearest neighbor of higher density. Obviously, these clusters are geographical regions that a user is more likely to visit and take activities than other places. Following the cluster extraction, we transform each cluster into a regular region, which is the foundation of similarity measure between two users. Given a cluster with points \( \{r_1^c, \ldots, r_n^c\} \), the corresponding stay region is generated by connecting the marginal points of the cluster, denoted as \( R \).

In real life, many people tend to visit popular areas, such as the downtown of a city, a large bus station, and a popular cinema. Then, many extracted stay regions of different users on a platform are overlapped. Such a phenomenon brings the great challenge to distinguish one user from others and makes it hard to link the user to his/her account from another platform. To address the problem, we need to assign different weights to these stay regions. Given a set of users with corresponding regions in Table 1(a), the calculation of region weights presented in Table 1(b) consists of two steps. Next, we present the calculation of the weight \( \omega(R) \).

<table>
<thead>
<tr>
<th>Table 1: Region Weight Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) User Region</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>User</td>
</tr>
<tr>
<td>( u_1 )</td>
</tr>
<tr>
<td>( u_2 )</td>
</tr>
<tr>
<td>( \cdots )</td>
</tr>
<tr>
<td>( u_m )</td>
</tr>
</tbody>
</table>

First, we compute the similarity between \( R_1^c \) and the regions of other users \( \{u_2, u_3, \ldots, u_m\} \), i.e., the similarity between \( R_1^c \) and \( \{R_2^c, \ldots, R_n^c, \ldots, R_n^c\} \). The similarity is defined as \( \sum_{R_n \in D_R} S(R_1^c, R_n) \), where \( D_R \) denotes the
Time distribution is another important feature in our model. Where \( \sum_{i} r_i \) is used to find optimal parameters with timestamp set \( u \) of a user \( u \). The weight of \( R_i \) and timestamps are assigned to different time slices, \( m \) is used to adopt and timestamps are assigned to different time slices, \( u \) is the possible maximum value of \( x \). The smaller \( \sum S(R_i, R_o) \) means it is more likely to distinguish \( u_1 \) from other users with the stay region \( R_i \). Normalizing the vector, we obtain the weight of \( R_i \):

\[
\omega(R_i) = \frac{1}{1 + \sum S(R_i, R_o)} \sum_{1}^{N} \sum_{1}^{N} S(R_i, R_o)
\]

### 4.2 Global Time Distribution

Time distribution is another important feature in our model. Unlike the study in [9], where the discretization method is adopted and timestamps are assigned to different time slices, we use the Gaussian Mixture Model (GMM) to model the time distribution of a user. This is because time is continuous, and a user usually takes activities around some time points [23]. For example, a user may get up around 7:00 am, have lunch around 12:00 am, and drive home around 18:00 pm. GMM is able to model these temporal behaviors, where timestamps are centered on some points.

In this section, we extract the temporal features from the global perspective, where the stay region factor is omitted. Given a set of stay region candidate points \((r_1^t, r_2^t, \ldots, r_n^t)\) of a user \( u \), the Expectation Maximization (EM) algorithm is used to find optimal parameters with timestamp set \((r_1^t, r_2^t, \ldots, r_n^t)\).

The Gaussian probability density function is:

\[
N(r_i^t; \mu_k, \Sigma_k) = \frac{1}{\sqrt{2\pi\Sigma_k}} \exp\left[-\frac{1}{2}(r_i^t - \mu_k)^2\Sigma_k^{-1}\right]
\]

**E-step:** The probability of the sample \( r_i^t \) generated by the \( k \)-th cluster \( (\mu_k, \Sigma_k) \) is:

\[
\gamma_{ik} = \frac{\alpha_k N(r_i^t; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \alpha_k N(r_i^t; \mu_j, \Sigma_j)}
\]

where \( K \) is the number of clusters.

**M-step:** The maximum likelihood method is used to update model parameters as follows:

\[
\alpha_k = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik}, \quad \mu_k = \frac{1}{\sum_{i=1}^{n} \gamma_{ik}} \sum_{i=1}^{n} \gamma_{ik} r_i^t, \quad \Sigma_k = \frac{1}{\sum_{i=1}^{n} \gamma_{ik}} \sum_{i=1}^{n} \gamma_{ik} (r_i^t - \mu_k)^2
\]

After updating parameters, we obtain a set of time clusters \((T_1, T_2, \ldots, T_n)\) for each user. Next, we introduce how to measure the similarity between two time clusters with the over-lapping coefficient (OVL) method [7][16]. Given \( T_1 = (\mu_1, \sigma_1) \) and \( T_2 = (\mu_2, \sigma_2) \) with following density functions:

\[
f_1(x) = \frac{1}{\sqrt{2\pi\sigma_1}} \exp\left(-\frac{(x - \mu_1)^2}{2\sigma_1^2}\right)
\]

\[
f_2(x) = \frac{1}{\sqrt{2\pi\sigma_2}} \exp\left(-\frac{(x - \mu_2)^2}{2\sigma_2^2}\right)
\]

First, we calculate the intersections of \( f_1(x) \) and \( f_2(x) \). Assume \( \sigma_1 > \sigma_2 \), we obtain the following intersections:

\[
x_1 = \left( \sqrt{2\sigma_1^2 \sigma_2^2 \ln\frac{\sigma_1}{\sigma_2}} - (\sigma_1^2 \mu_2 - \sigma_2^2 \mu_1) + \frac{(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)^2}{\sigma_1^2 - \sigma_2^2} \right) / \sqrt{\sigma_1^2 - \sigma_2^2}
\]

\[
x_2 = \left( -\sqrt{2\sigma_1^2 \sigma_2^2 \ln\frac{\sigma_1}{\sigma_2}} - (\sigma_1^2 \mu_2 - \sigma_2^2 \mu_1) + \frac{(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)^2}{\sigma_1^2 - \sigma_2^2} \right) / \sqrt{\sigma_1^2 - \sigma_2^2}
\]

Then, the similarity \( S(T_1, T_2) \) is given as:

\[
S(T_1, T_2) = \int_{x_1}^{x_2} f_2(x)dx + \int_{x_1}^{x_2} f_1(x)dx + \int_{x_2}^{x_1} f_2(x)dx
\]

(7)

Given the user \( u_1 \) with time clusters \((\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_i, \sigma_i)\), the computation of \( \omega(T_i) = (\mu_i, \delta_i) \) consists of two steps. First, we compute the similarity between \( T_i \) and the time clusters of other users, i.e., \( \sum_{T_j \in D_T} S(T_i, T_j) \), where \( D_T \) denotes the extracted time clusters of \( u_1 \) and \( D_T \) denotes the extracted time clusters of all users. Then, we obtain a set \((\sum S(T_i, T_j), \sum S(T_i, T_j), \ldots, \sum S(T_i, T_j))\) for \( u_1 \), where \( \sum_{T_j \in D_T} S(T_i, T_j) \) is denoted as \( \sum S(T_i, T_j) \).

Second, the function \( f(x) = \frac{N}{1 + x} \) is also used to assign larger weights to clusters with smaller \( \sum S(T_i, T_j) \), and we obtain the vector \((\sum S(T_i, T_j), \sum S(T_i, T_j), \ldots, \sum S(T_i, T_j))\). Then, the weight \( \omega(T_i) \) is defined as:

\[
\omega(T_i) = \frac{N}{\sum_{T_j \in D_T} S(T_i, T_j)}
\]

(8)
4.3 Local Time Distribution

Considering the time distribution from global perspective, we can obtain time clusters around some certain time points, such as $T_0 = (7:00, 2)$, $T_1 = (12:00, 1)$, and $T_2 = (18:00, 3)$. By further analyzing the user behaviors with spatial information, we may extract $T_0$ in home region, extract $T_1$ in school region, and extract $T_2$ in bus station, that is each stay region has its own time distribution. It is worth noting that this feature is of critical importance in modeling user behaviors. Considering the example in Table 2, where parameters correspond to global time distribution, location corresponds to stay region, and local time denotes the time cluster extracted in current stay region, $T_2\{R_3\}$ means we can extract the time cluster $T_2 = (8:00, 2)$ in stay region $R_3 = (3, 4; 8, 9)$, where (3, 4) and (8, 9) denote the lower-left and upper-right of current stay region $R_3$, respectively. We assume that $<u_0, u_1>$ is an actual linked pair. Based on the global time distribution, we can remove the link $<u_0, u_3>$ from the candidate set $\{<u_0, u_1>, <u_0, u_2>, <u_0, u_3>\}$. However, we cannot confirm which pair in $\{<u_0, u_1>, <u_0, u_2>\}$ is the actual linked pair based on stay region distribution and global time distribution. Fortunately, by taking the time distribution in each stay region into account, we can prune the link $<u_0, u_2>$, since $u_0$ and $u_2$ have opposite distribution in stay regions.

The example in Table 2 demonstrates the importance of modeling local time distribution in each stay region. To address the problem, EM algorithm is also used to evaluate parameters, and OVL is used to measure similarities. Namely, given a stay region of $u_i$, we extract the time clusters $\{(\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_m, \sigma_m)\}$ in this region, and assign weight to each cluster $\{(\mu_i, \sigma_i)\}$ based on the approaches proposed in Section 4.2.

5 SIMILARITY MEASURE

Based on the extracted spatio-temporal features and their weights, we develop novel approaches to measure the similarity between two users.

(1) **Stay Region Similarity**: Assume the stay regions of $u_1$ and $u_2$ are $\{(R_1, \omega(R_1^1)), (R_1^2, \omega(R_1^2)), \ldots, (R_1^n, \omega(R_1^n))\}$ and $\{(R_2, \omega(R_2^1)), (R_2^2, \omega(R_2^2)), \ldots, (R_2^n, \omega(R_2^n))\}$, the stay region similarity between $u_1$ and $u_2$ is defined as:

$$S(u_1, u_2)_r = \sum_{i=1}^{m} \sum_{j=1}^{n} S(R_1^i, R_2^j) \omega(R_1^i) \omega(R_2^j)$$

where $S(R_1^i, R_2^j)$ is calculated according to Eq.(1).

(2) **Global Time Similarity**: Assume the global time distributions of $u_1$ and $u_2$ are $\{(T_1^1, \omega(T_1^1)), (T_1^2, \omega(T_1^2)), \ldots, (T_1^n, \omega(T_1^n))\}$ and $\{(T_2^1, \omega(T_2^1)), (T_2^2, \omega(T_2^2)), \ldots, (T_2^n, \omega(T_2^n))\}$, the global time similarity $S(u_1, u_2)$ is given as:

$$S(u_1, u_2) = \sum_{i=1}^{k} \sum_{j=1}^{l} S(T_1^i, T_2^j) \omega(T_1^i) \omega(T_2^j)$$

where $S(T_1^i, T_2^j)$ is calculated based on Eq.(7).

(3) **Local Time Similarity**: Assume the time distribution in a stay region $\{(R_1, \omega(R_1))\}$ of $u_1$ is $\{(T_1^1, \omega(T_1^1)), (T_1^2, \omega(T_1^2)), \ldots, (T_1^n, \omega(T_1^n))\}$, and the time distribution in an extracted stay region $\{(R_2, \omega(R_2))\}$ of $u_2$ is $\{(T_2^1, \omega(T_2^1)), (T_2^2, \omega(T_2^2)), \ldots, (T_2^n, \omega(T_2^n))\}$, the local time similarity in these two regions is defined as $\sum_{i=1}^{k} \sum_{j=1}^{l} S(T_1^i, T_2^j) \omega(T_1^i) \omega(T_2^j)$. The similarity between $u_1$ and $u_2$ based on this feature is defined as:

$$S(u_1, u_2)_l = \sum_{i=1}^{m} \sum_{j=1}^{n} S(R_1^i, R_2^j) \omega(R_1^i) \omega(R_2^j)$$

Finally, the similarity between $u_1$ and $u_2$ is defined as:

$$S(u_1, u_2) = S(u_1, u_2)_r + S(u_1, u_2)_l + S(u_1, u_2)_r$$

6 EXPERIMENT

6.1 Experiment Settings

Three real-world datasets, which contain the trajectory data in Beijing and the check-in data in Foursquare, Twitter, and Instagram, are used to conduct experiments. The details of these datasets are presented in Table 3.

**Beijing Walk Trajectories - Beijing Car Trajectories**: The first dataset is collected by (Microsoft Research Asia) GeoLife project in Beijing 2 with different transportation modes, such as walk, car, bus, bike, etc. This dataset records a broad range of users’ outdoor movements, such as shopping, sightseeing, driving, and cycling. To investigate the performance of the proposed model in linking the same users across different domains, we select the walk trajectory data and the car trajectory data, since they contain

2https://www.microsoft.com/en-us/research/people/yuzheng

### Table 2: An Example of Local Time Distribution

<table>
<thead>
<tr>
<th>User</th>
<th>Platform A</th>
<th>Platform B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>$[T_{11}, T_{12}]$</td>
<td>$[T_{21}, T_{22}]$</td>
</tr>
<tr>
<td>$u_2$</td>
<td>$[T_{31}, T_{32}]$</td>
<td>$[T_{41}, T_{42}]$</td>
</tr>
<tr>
<td>$u_3$</td>
<td>$[T_{51}, T_{52}]$</td>
<td>$[T_{61}, T_{62}]$</td>
</tr>
</tbody>
</table>
more locations than other transportation modes. The selected dataset contains 182 actual linked user pairs in real life. It contains 14337 walk trajectories with 2190957 locations, and 5475 car trajectories with 925380 locations. Each location is a tuple in the form of (user-id, trajectory-id, latitude, longitude, timestamp).

**Foursquare-Twitter.** Foursquare and Twitter are popular social networks, where users can post status associated with location information and timestamps. To evaluate the performance of our model in connecting the users across these two platforms with spatio-temporal features, we use the dataset provided by [14]. However, some users in the given dataset only contain one or two check-in records, which are insufficient to extract users’ spatio-temporal features. Consequently, we preprocess the original dataset, as the ground truth. The dataset contains 3924 check-ins during time interval \( t \), where each check-in is stored as a tuple (user-id, latitude, longitude, timestamp). To link users across Instagram and Twitter, we use the dataset provided by [26][14]. However, some users in the dataset also only contain one or two check-in records. Finally, we obtain a new dataset with 89 linked user pairs as the ground truth. The dataset contains 3924 check-ins in Foursquare and 35784 check-ins in Twitter, where each check-in is stored as a tuple (user-id, latitude, longitude, timestamp).

**Instagram-Twitter.** Instagram is another popular social network, where users can share photos associated with locations and timestamps. To link users across Instagram and Twitter, we use the dataset provided by [14]. Some users in the dataset also only contain one or two check-in records. As a consequence, we remove users who have less than 30 check-ins. Finally, the new dataset consists of 908 ground truth user pairs, and contains 267029 check-ins in Instagram and 357949 check-ins in Twitter, where each check-in is in the form of (user-id, latitude, longitude, timestamp).

In the rest of paper, we use BJW-BJC, FS-TW, and IT-TW to denote the first, second, and third dataset respectively for the sake of convenience.

**Compared Algorithms.** To investigate the performance of the proposed model, we compare the performances of the following algorithms.

Baseline 1 (GC): Existing work [9] has proposed a method to discretize the space into grid cells, and calculate the density \( f(c) \) for each grid cell \( c \) with a kernel method:

\[
f(c) = \frac{1}{n^2} \sum_{i=1}^{n} \frac{1}{2\pi} \exp \left( -\frac{|c - loc_i|^2}{2\gamma^2} \right)
\]

\[
\gamma = \frac{1}{2} (\delta_x^2 + \delta_y^2) \cdot \frac{1}{n^{\frac{\alpha}{2}}}
\]

where \(|c - loc_i| \) is the distance between \( c \) and location \( loc_i \), \( \delta_x \) and \( \delta_y \) denote the standard deviations of the whole locations in its x and y-coordinates, respectively. Next, the algorithm returns the top-15% grid cells with the maximum density as dense regions that a user will visit repeatedly. Based on these regions, we calculate the similarity between users. For instance, given two users \( u_1 \) and \( u_2 \) with returned grid cells \( C_1 = \{c_{11}, c_{12}, \ldots, c_{1m}\} \) and \( C_2 = \{c_{21}, c_{22}, \ldots, c_{2n}\} \), respectively. The similarity \( S(u_1, u_2) \) is defined as:

\[
S(u_1, u_2) = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|c_{1i} \cap c_{2j}|}{|c_{1i} \cup c_{2j}|}
\]

Baseline 2 (LT): Locations and times are divided into bins, where each record in region \( l \) during time interval \( t \) is associated with bin \((i, j, t)\). Then, the user similarity is measured by the corresponding bins [14].

Baseline 3 (STUL-S): This is a simplified version of STUL, where the extracted stay region distribution and time distribution are directly used to measure the similarity between users without consideration of their weights. The similarity function is defined as:

\[
S(u_1, u_2) = \sum_{i=1}^{m} \sum_{j=1}^{n} S(R_{1i}, R_{2j}) + \sum_{i=1}^{m} \sum_{j=1}^{n} S(T_{1i}, T_{2j})
\]

\[
+ \sum_{i=1}^{m} \sum_{j=1}^{n} S(R_{1i}, R_{2j}) \sum_{i=1}^{m} \sum_{j=1}^{n} S(T_{1i}, T_{2j})
\]

STUL: All extracted features are taken into account to measure the similarity between users, i.e., the similarity \( S(u_1, u_2) = S(u_1, u_2)_1 + S(u_1, u_2)_2 + S(u_1, u_2)_3 \), defined in Eq.(12). Unlike STUL-S, we assign different weights to the extracted features based on their uniqueness and discriminability.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Users</th>
<th>Trajectories</th>
<th>Locations/Check-ins</th>
</tr>
</thead>
<tbody>
<tr>
<td>BJW-BJC</td>
<td>Walk</td>
<td>182</td>
<td>14337</td>
<td>2190957</td>
</tr>
<tr>
<td></td>
<td>Car</td>
<td>182</td>
<td>5475</td>
<td>925380</td>
</tr>
<tr>
<td>FS-TW</td>
<td>Foursquare</td>
<td>89</td>
<td>-</td>
<td>3924</td>
</tr>
<tr>
<td></td>
<td>Twitter</td>
<td>89</td>
<td>-</td>
<td>35784</td>
</tr>
<tr>
<td>IT-TW</td>
<td>Instagram</td>
<td>908</td>
<td>-</td>
<td>267029</td>
</tr>
<tr>
<td></td>
<td>Twitter</td>
<td>908</td>
<td>-</td>
<td>357949</td>
</tr>
</tbody>
</table>

**Table 3: Properties of the Given Datasets**
as the fraction of the actual linked user pairs contained by the returned result \cite{12},

$$
Recall = \frac{k}{n}, \quad Precision = \frac{k}{m}
$$

$$
F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}
$$

where \(n(n = |U_1| = |U_2|)\) is the number of given user pairs, \(m\) is the number of returned user pairs, and \(k\) is the number of actual linked user pairs in the returned result.

### 6.2 Performance Evaluation

The precision, recall, and F1 of all algorithms are presented in Fig.2. As expected, LT has better performance than GC, since GC only considers grid cells with high density, whereas both spatial and temporal information are taken into account in LT. The simplified model STUL-S outperforms the discretization based methods GC and LT, and the reason is twofold. On one hand, STUL-S has considered the continuity of user regions. Compared with the rectangle grid cells, the stay regions extracted by STUL-S with arbitrary shapes are more likely to model the real regions of a user. On the other hand, STUL-S has considered the continuity of time, where the time clusters centered on some time points are extracted by the Gaussian Mixture Model. Unlike STUL-S, STUL assigns different weights to the extracted features, where the discriminative features are highlighted with larger weights and the common features are lightened with smaller weights. Then, the actual linked users are more likely to be connected with these features. As a result, STUL performs best in all datasets.

In addition, the results in Fig.2 demonstrate the universality of the proposed model STUL. This is because the trajectories in BJW-BJC are densely sampled, where the adjacent points in a trajectory are sampled in a short time period. The check-in records in Foursquare, Twitter, and Instagram are sampled with a low rate, where the time span between two check-ins is usually large.

**Performance w.r.t. Varied \(\theta\).** As discussed in algorithm 1, given two user datasets \(U_1\) and \(U_2\), the user pair \(<u_{11}, u_{22}> (u_{11} \in U_1, u_{22} \in U_2)\) is added into the result if \(S(u_{11}, u_{22}) \geq \theta\). Naturally, the increase of \(\theta\) will lead to the decrease of recall, as many candidate user pairs are filtered. After filtering user pairs with small similarities, the remaining user pairs with high similarities are more likely to be the actual linked user pairs, as which usually have larger similarities than unmatched user pairs in real life. Consequently, the increase of \(\theta\) also leads to the increase of precision. To balance the precision and recall, we set \(\theta = 0.25\) in BJW-BJC, \(\theta = 0.5\) in FS-TW, and \(\theta = 0.5\) in IT-TW.

**Performance w.r.t. Varied Cutoff Distance.** The cutoff distance \(d_c\) is an important parameter of STUL, as \(p\) and \(\delta\) are calculated based on \(d_c\), where points within \(d_c\) are considered. From Fig.4, we observe that STUL achieves higher recall with the increase of \(d_c\). This is because, given a larger \(d_c\), the extracted stay regions will contain more points and have larger areas. Then, the across domain users are more likely to have common areas and larger user similarity, which leads to the increase of recall. However, for users on the same platform, a larger \(d_c\) will lead to the decrease of the distinction of their stay regions, since these regions tend to contain same points and have similar distribution with a larger \(d_c\). Then, it is difficult to link one of them to the actual linked user from another platform. This difficulty leads to the decrease of precision. Considering the diversity of the given datasets, we set different cutoff distances to balance the precision and recall, i.e., \(d_c = 3000\) m in the dataset BJW-BJC, \(d_c = 300\) m in FS-TW, and \(d_c = 1000\) m in IT-TW.

**Performance w.r.t. Varied \(\xi\).** Another import parameter of STUL is \(\xi\), as the top-k centers with the maximum \(\xi = p \times \delta\) are returned. In Fig.5, we set \(p = (0, 1, 2, 3, 4, 5), \delta = (0, 100, 200, 300, 400, 500)\), thus \(\xi\) is varied from 0 to 2500. This figure shows that the model STUL achieves the best performance in some certain points, although the change in performance is small when using different \(\xi\). A too large or too small \(\xi\) is not appropriate. On one hand, given a small \(\xi\), the users in a platform tend to have similar features as too many stay regions are returned. The overlap of these features leads to the bad performance of STUL. On the other
hand, given a large $\xi$, we can extract important and discriminative stay regions, as many centers with small $\xi$ are filtered. However, the number of the returned regions is small, which also leads to the bad performance of STUL. To achieve the best performance, we set $\xi = 1600$ in BJW-BJC, $\xi = 900$ in FS-TW, and $\xi = 100$ in IT-TW.

Efficiency. Besides the high effectiveness, STUL is very efficient, where the average time it takes to find an actual


[9] Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. Mining Periodic Behaviors for Moving Objects. In *KDD*. 1099–1108.


