Effective and Efficient User Account Linkage Across Location Based Social Networks

Wei Chen[†] Hongzhi Yin[‡]* Weiqing Wang[‡] Lei Zhao[†] Xiaofang Zhou[‡]
[†]*School of Computer Science and Technology, Soochow University*
[‡] *School of Information Technology and Electrical Engineering, The University of Queensland*
[†]wchzhg@gmail.com [†]zhaol@suda.edu.cn
[‡]db.hongzhi@gmail.com [‡]{weiqingwang,zxf}@itee.uq.edu.au

***Abstract*—Sources of complementary information are connected when we link the user accounts belonging to the same user across different domains or devices. The expanded information promotes the development of a wide range of applications, such as cross-domain prediction, cross-domain recommendation, and advertisement. Due to the great significance of user account linkage, there are increasing research works on this study.**

With the widespread popularization of GPS-enabled mobile devices, linking user accounts with location data has become an important and promising research topic. Being different from most existing studies in this domain that only focus on the effectiveness, we propose novel approaches to improve both effectiveness and efficiency of user account linkage. In this paper, a kernel density estimation (KDE) based method has been proposed to improve the accuracy by alleviating the data sparsity problem in measuring users' similarities. To improve the efficiency, we develop a grid-based structure to organize location data to prune the search space. The extensive experiments conducted on two real-world datasets demonstrate the superiority of the proposed approach in terms of both effectiveness and efficiency compared with the state-of-art methods.

I. Introduction

The proliferation of GPS-enabled devices, such as vehicles, mobile phones, and smart bracelets, leads to the increasing availability of location data from two perspectives: 1) the volume of location data increases unprecedentedly; 2) the resources of location data tend to be more diverse. Recently, much more location data have been generated by newly-emerging location-based social networks (LBSNs) [1][2][3], such as Foursquare, Twitter, and Instagram. Many users have registered accounts on these platforms, and posted their statuses associated with location information, referred as "check-ins". Compared with other online activities [4][5], such as commenting, tagging, and following, "check-ins" bridge the gap between the real world and the virtual world with the geographical data [6][7]. The study of check-in data provides an unprecedented opportunity to analyze users' real world behaviors and potentially improve a variety of location-based services [8][9]. For example, in [10], check-in data are used to link user accounts across different platforms. Obviously, compared with the information collected from one specific platform, we can obtain more comprehensive user information after identifying and linking user accounts across platforms, since the sources of complementary information are integrated. From a commercial perspective, this expanded information will benefit many location-aware applications, such as maps, cross-domain recommendation, and advertisement. As a consequence, linking user accounts across location based social networks has attracted increasing attention. However, despite of the significance of the study, following inevitable problems bring great challenges for this work.

* This author is the corresponding author

A. Challenges

Data Sparsity. The density of the check-in data for each user is of critical importance to user account linkage across LBSNs. This is because we can model a user's real behaviors more precisely with more data, which enables us to link user accounts across different platforms more accurately. Unfortunately, user-generated check-in datasets are extremely sparse. Compared with the traditional GPS datasets, where users' geographical location is automatically recorded by the GPS devices and the time period between two consecutive points is usually short, the check-in process is user-driven on location-based social networks, i.e., users decide whether to check in at a specific place or not due to privacy concerns. Such user-driven mechanism leads to the data sparsity problem from the following aspects. First, the number of check-in records generated by each user is rather limited, as many users are reluctant to post their statuses due to privacy concerns [11]. Second, the spatial span of check-in data is extreme large [6]. For example, a user may usually check in at Boston, but with the latest check-ins at California. Third, the time spans between consecutive check-ins are usually wide, where some users even have more than one-year gaps between consecutive check-ins [6][12]. All these behaviors lead to the significant sparsity of geographical data in location-based social networks, which greatly increases the difficulty of data analysis.

To illustrate the data sparsity problem of user check-ins more clearly, we conducted an analysis on two real cross-domain datasets, i.e., the dataset Foursquare-Twitter (FS-TW) provided by [13][10] and the dataset Instagram-Twitter (IT-TW) provided by [10]. The analysis results are presented in Figure 1. The distribution of the average number of check-in records is presented in Fig. 1(a). Obviously, most users in the given datasets have a small volume of check-in records (i.e., usually less than 50 check-in records). Fig. 1(b) reveals the data sparsity problem from a different perspective, where the density of each user is calculated based on the area of corresponding dataset and defined as the number of check-in records per km^2. For most users, they have less than 0.2 check-ins in one km^2. Note that, in Fig. 1(b) we omit the user whose density is larger than 0.6, since the number of these users is very small.

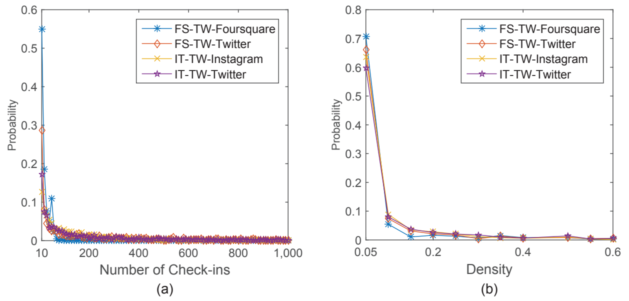


Fig. 1. (a) The distribution of the number of check-ins; (b) The distribution of density, i.e., the number of check-ins/km^2

Data Missing. Data missing in user account linkage refers to the phenomenon that, some of a user's check-in records are missing on certain platforms, as a user is not likely to post the check-in about the same activity many times across different platforms. For example, many users have registered accounts in Facebook, Twitter, and Instagram simultaneously, yet they may just select one platform to post a check-in after taking an activity in a venue or location. Moreover, users' consistent preferences make the situation worse. Users' consistent preferences refer to that users may prefer one platform to post their check-ins consistently, which can be caused by various factors (e.g., the influence of social friends). For example, if most of a user's friends use Facebook, he/she may always prefer posting check-ins in Facebook, which means that most of his/her activities are missing on other platforms.

Negative Coincidence. Coincidence occurs when two users happen to have many historical check-in records at the same places across different domains but many of them are not the same user[8]. Such phenomenon tends to happen in popular and crowded places (e.g., supermarkets, cafeteria, and schools), where many different users tend to visit repeatedly [14][15]. The phenomenon is a negative factor to link the same user, thus it is called "negative coincidence" in this study.

B. Our Solutions

A straightforward approach to discover two actual linked user accounts is to measure their similarity by comparing the common locations that they both have visited. However, users normally share few co-visited locations across different platforms due to the data sparsity and data missing problems introduced in Section I-A. To overcome this challenge, we propose a kernel density estimation (KDE) based method to accurately characterize the spatial pattern of an individual's check-in activities and then perform user account linkage based on their spatial patterns, inspired by [9]. Although KDE is able to alleviate the data sparsity, this approach is inherently time consuming [16][17].

To improve the computational efficiency of KDE, we propose a grid-based KDE. Specifically, we divide the space into $d \times d$ grid cells, and then each user is represented by a sequence of grid cells with corresponding confidences. Compared with representing a user with a sequence of check-in locations, the grid-based method is more efficient as the total number of grid cells is much smaller than that of locations. To further improve the efficiency, we propose a novel pruning strategy to significantly reduce down the number of cell-pair computations. To be more specific, we first construct a square region centered at each grid cell g_i, and only the cells within the square region will be considered for g_i in KDE. Another benefit brought by grid-based KDE is to relieve the data missing issue. This is because although a user often posts different check-in activities on different social network platforms, the spatial distribution (e.g., the cell distribution) of his/her check-in records generated on each platform tends to be similar to each other [18].

To address the third challenge, we design an entropy-based weight scheme for locations and grid cells to reduce the impact of negative coincidence. As we introduced before, people tend to visit popular locations, which leads to the location coincidence. Obviously, these locations usually have large entropy in terms of the visited users. However, they are less useful in distinguishing users from each other. In contrast, a private location, such as home or office, is more discriminative in identifying users. Therefore, our entropy-based weight scheme will penalize locations and grid cells with high entropy by assigning low weights.

C. Summary of Our Contributions

In this study, several approaches have been proposed to tackle the challenges that we are facing in user account linkage across location-based social networks. To sum up, we make the following contributions.

- We propose a general method to perform user account linkage with location data by considering both effectiveness and efficiency simultaneously.
- To tackle the data sparsity problem, we design a novel algorithm based on kernel density estimation. To tackle the data missing problem, we divide the space into grid cells and calculate the confidence for each cell. Furthermore, we design an entropy-based weight scheme for the grid cells with the goal of alleviating the challenges caused by negative coincidence.
- We conduct extensive experiments on two real-world datasets, and the results demonstrate that the proposed approach outperforms the state-of-the-art methods in terms of both effectiveness and efficiency.

The rest of the paper is organized as follows. We present the related work in Section II, and formulate the problem in Section III. The outline of our study is presented in Section IV. We construct the index structure in Section V and present the calculation of grid cell weight in Section VI. The experiment results are reported in Section VII, and the paper is concluded in Section VIII.

II. Related Work

The related studies, which contain the cross-domain user account linkage and the applications of kernel density estimation in spatio-temporal database, are discussed in this section.

A. User Account Linkage

The increasing popularity of social networks has enabled more and more people to participate in multiple online services [19]. Linking the same users across different platforms brings a great opportunity to fully understand users' behaviors and provide better recommendations. The study is firstly proposed in [20], where cross-community identities are connected with corresponding websites by measuring the identity similarity with usernames. Vosecky et al. [21] proposed a method to identify users based on web profile matching and further extended its effectiveness by incorporating the user's friend network. To investigate whether users can be identified across systems based on their tag-based profiles, an aggregate profile was constructed by combining usernames and user tags [22]. Following these studies, more abundant information was considered to link user accounts [23][24][25][26][27]. To build a comprehensive user profile for improving online services, various sources of complementary information were integrated [23], where the username features, prior-username features, and the relation between the candidate usernames and prior usernames were taken into account. To match user accounts from different online social networks, Peled et al. [24] used supervised learning techniques to construct different classifiers, where three main types of features were utilized, i.e., name based features, social network topological based features, and general user info based features. To address the multi-platform user identity linkage problem, Mu et al. [28] proposed two effective algorithms, a batch model ULink and an online model ULink-On, based on latent user space modeling. Recent advances [29][30] focused on using location data to achieve user account linkage. By utilizing the user-generated location data in social media platforms, a co-clustering-based framework was proposed [29], where account clusterings in spatial and temporal and dimensions were carried out synchronously. To address the challenges in general cross-domain case, where users have different profiles independently generated from a common but unknown pattern, a generic and self-tunable algorithm that leverages any pair of sporadic location-based datasets was proposed to determine the most likely matching between users [30]. To measure user similarity with trajectories gathered from different data sources, Cao et al. [31] proposed a framework called Automatic User Identification (AUI), which is based on a novel similarity measure called the signal based similarity (SIG). To examine the question whether publicly available spatio-temporal user data can be used to link newly observed location data to known user profiles, Seglem et al. [32] developed many novel methods. More importantly, the study can be seen as an adversary approach of trying to breach the privacy of users, as it does not try to maintain privacy of users.

B. Kernel Density Estimation in Spatio-Temporal Database

Kernel density estimation (KDE) is a statistical technique for estimating a probability density function from a random sample set [33][34]. As a common tool, KDE has been explored in various areas for different purposes [17], especially in spatio-temporal database [35][9][36]. To study the personalized geographical influence of locations on a user's behaviors, the kernel density estimation was used to model the personalized distribution of the distance between any pair of locations [35]. To understand urban human activity and mobility patterns, Hasan et al. [37] applied a two dimensional Gaussian kernel to estimate the check-in density of each grid cell. To investigate the spatio-temporal clustering of trajectories, a Gaussian kernel function was used to calculate the spatio-temporal kernel density of each trajectory unit [38]. To determine the geographical point of a text document, Hulden et al. [36] investigated an enhancement of common methods by kernel density estimation.

Connecting user accounts across different social platforms has been well studied by previous studies from different perspectives, yet there is no work investigates novel approaches to further synchronously improve the effectiveness and efficiency of user account linkage with location data. To achieve higher performance, and address the challenges: data sparsity, data missing, and location coincidence that are introduced in Section I, we investigate the application of kernel density estimation, construct a grid index, and calculate the grid cell weight based on corresponding entropy.

III. Problem Definition

In this section, we first present the notations used throughout the paper and then formulate the problem.

TABLE I
DEFINITIONS OF NOTATIONS

Notation	Definition
u	A user account
U	A set of user accounts
l	A location in the form of (lat, lng)
r	A check-in record in the form of (u, l, t)
R	A set of check-in records
g	A grid cell
$\varpi(g)$	Confidence for the grid cell g
$\omega(g)$	Weight for the grid cell g
$G(u)$	Grid representation of u
$H(g)$	Entropy of the grid cell g
S_Δ	Similarity threshold
$S(u_1, u_2)$	Similarity between user accounts u_1 and u_2

On social networks, many users share ideas and check in after taking activities at a place. Then, the following information will be recorded and sent to the server: a unique user account id that distinguishes a user account from others; location information that consists of latitude and longitude; and time-stamp of the check-in record [8].

Definition 1 ***Check-in Record***. *A check-in record of a user is defined as $r = (u, l, t)$, where u is a user account, l is defined as (lat, lng) with lat represents latitude and lng represents longitude, and t is the time-stamp of the check-in.*

Note that, the location data $(i.e., lat\ and\ lng)$ of records are used to measure the similarity between a user account pair, and time-stamps are used to distinguish records from each other. For instance, given two records $r_1 = (u_1, l_1, t_1)$ and $r_2 = (u_2, l_2, t_2)$, they are defined as different records if $t_1 \neq t_2$ even though $u_1 = u_2$ and $l_1 = l_2$. This definition is appropriate, as a user may frequently check in at the same place where he/she usually visits. The semantic information behind the records with same location may be diverse. For example, a user may check in at a cafeteria alone on Monday, but check in with his/her friends at the same place on Sunday.

Problem Formulation. Given two sets of user accounts $U_1 = \{u_{11}, u_{12}, \cdots, u_{1m}\}$ and $U_2 = \{u_{21}, u_{22}, \cdots, u_{2n}\}$ on two different location based social networks, where each user account is associated with a set of check-in records, our goal is to identify all account pairs (u_{1i}, u_{2j}) of the same user from $\{(u_{1i}, u_{2j}) | u_{1i} \in U_1, u_{2j} \in U_2\}$.

Existing study [10] focuses on maximizing the weighted matching on the bipartite graph to return the linked user account pairs, where $|U_1| = |U_2|$ and for each user account u_{1i} in U_1, one and only one user account u_{2j} in U_2 is returned, i.e., the mapping between U_1 and U_2 is one-to-one. However, in reality, different platforms usually have different numbers of users and the mappings between user accounts might be many-to-many as some users may have more than one account on a single platform. Compared with their work, the problem studied in our paper is more general and practical, which is reflected in following two aspects. First, our problem covers the situation where $|U_1| \neq |U_2|$, i.e., the number of user accounts on different platforms can be different. Second, given a user account u_{1i} in U_1, our problem aims at returning all possible user accounts u_{2j} matching with u_{1i}.

IV. Proposed Algorithm

In this section, we propose a basic kernel density estimation-based solution to perform user account linkage across location-based social networks. The intuition behind our solution is that given two user accounts u_{1i} and u_{2j} of the same user on two different LBSNs, the spatial distributions of her/his generated check-in records on the two LBSNs are similar to each other, even if the user posts different check-ins on these two platforms. Our solution contains two main components. Firstly, for each user account pair (u_{1i}, u_{2j}) in the Cartesian product $U_1 \times U_2 = \{(u_{1i}, u_{2j}) | u_{1i} \in U_1, u_{2j} \in U_2\}$, we compute their similarity $S(u_{1i}, u_{2j})$. Secondly, based on the inferred similarity and a user-defined similarity threshold S_Δ, we decide whether these two accounts belong to the same user.

A straightforward way to measure the similarity between two user accounts with discrete check-in records is to directly compare the records happened at same locations. Unfortunately, as discussed in Section I, user generated check-in records on location-based social networks are extremely sparse. Moreover, the issue of data missing worsens the situation. In light of these two challenges, we propose a kernel density estimation (KDE) based solution, inspired by its success in modeling individual-level location data [9]. Kernel density estimation is a non-parametric method for estimating the probability density function of a sample set with unknown distribution. Given a set of locations $L = (l_1, l_2, \cdots, l_n)$ and a location l', where each location is a two-dimensional tuple in the form of (lat, lng), the density of l' over L is estimated as follows:

$$f(l'|L,h) = \frac{1}{n}\sum_{i=1}^{n} K_h(l', l_i) \tag{1}$$

$$K_h(l', l_i) = \frac{1}{2\pi h}\exp\big(-\frac{(l'-l_i)^2}{2h^2}\big) \tag{2}$$

where $K(\cdot)$ is the Gaussian kernel function, h is a bandwidth parameter, and $(l' - l_i)$ is defined as the Euclidean distance between l' and l_i.

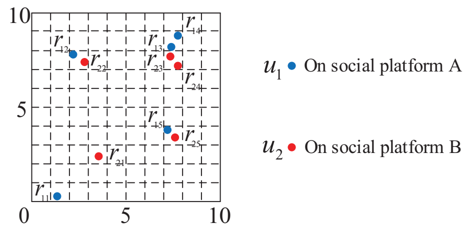
We use the probability density function $f(l'|L,h)$ to denote the similarity between L and l'. The similarity value $f(l'|L,h)$ will be large if l' is close to the points in L. In contrast, the value of $f(l'|L,h)$ is small if points in L are far away from l'. Given two users accounts u_1 and u_2, assume the check-in record sets of u_1 and u_2 are $R_{u_1} = (r_{11}, r_{12}, \cdots, r_{1n})$ and $R_{u_2} = (r_{21}, r_{22}, \cdots, r_{2m})$, respectively. The similarity between u_1 and u_2 is defined as:

$$\begin{aligned} S(u_1, u_2) &= \sum_{i=1}^{n} f(l_i | R_{u_2}, h) \\ &= \sum_{i=1}^{n} \frac{1}{m} \sum_{j=1}^{m} K_h(r_{1i}.l_{1i}, r_{2j}.l_{2j}) \end{aligned} \tag{3}$$

where $r_{1i}.l_{1i}$ denotes the location $(i.e., lat\ and\ lng)$ of the record r_{1i}.

V. Grid Structure

Kernel density estimation is an important statistical technique in data analysis. According to Eq. (3), it requires mn kernel evaluations to measure the similarity $S(u_1, u_2)$ as $|R_{u_1}| = n$ and $|R_{u_2}| = m$. It is obvious that the naive evaluation of KDE is very time consuming, especially for large-scale datasets with millions of check-ins. To speed up the evaluation of KDE, we propose a grid-based index structure to organize the location data.



(a) 10×10 grid on the sparse dataset

	u_1					u_2				
record	r_{11}	r_{12}	r_{13}	r_{14}	r_{15}	r_{21}	r_{22}	r_{23}	r_{24}	r_{25}
grid id	2	73	88	88	38	24	73	78	78	38

(b) Record information

Fig. 2. Grid Structure

As shown in the example of Fig. 2(a), we divide the space into 10×10 square cells, and the grid map and record information are presented in Fig. 2. By assigning each cell a unique numerical id from bottom to top and from left to right, u_1 and u_2 are represented by a set of discrete cells, i.e., $\{2, 73, 88, 38\}$ and $\{24, 73, 78, 38\}$. Due to the personal interests and geographical influence [15], the probabilities that a user visits different places and regions are different in real life. Thus, we propose to compute the grid confidence for each user account using Eq. (4).

Definition 2 ***Grid Confidence****. Given a user account u with a set of check-in records $R_u = (r_1, r_2, \cdots, r_n)$, the confidence of each grid cell visited by u is defined as:*

$$\varpi(g_i) = \frac{N(g_i)}{|R_u|} \tag{4}$$

where $N(g_i)$ denotes the number of records falling into the grid cell g_i.

Based on the grid id and the computed grid confidence, the grid representation of each user account is defined as $G(u) = \{(g_1, \varpi(g_1)), (g_2, \varpi(g_2)), \cdots, (g_m, \varpi(g_m))\}$. Note that, the grid representation has the following nice properties.

- Property 1: $\forall g_i \in G(u)$, $\varpi(g_i) > 0$.
- Property 2: $\forall g_i, g_j \in G(u)$, if $i \neq j$, then $g_i \neq g_j$.
- Property 3: $\varpi(g_1) + \varpi(g_2) + \cdots + \varpi(g_m) = 1$.

For each user, we just consider the grid cells that contain non-zero historical check-in records, thus we have the property 1 and property 2. Besides, we can obtain the property 3 based on Eq. (4). Continue the example in Fig. 2, we have $G(u_1) = \{(2, 0.2), (73, 0.2), (88, 0.4), (38, 0.2)\}$ and $G(u_2) = \{(24,\ 0.2), (73, 0.2), (78, 0.4), (38, 0.2)\}$. Based on the grid representation, we redefine the computation of KDE and similarity $S(u_1, u_2)$ as follows:

$$\begin{aligned} f(g_{1i}|G(u_2), h) &= \frac{1}{l}\sum_{j=1}^{l} K_h(g_{1i}, g_{2j}) \\ K_h(g_{1i}, g_{2j}) &= \frac{1}{2\pi h}\exp\big(-\frac{(g_{1i} - g_{2j})^2}{2h^2}\big)\varpi(g_{1i})\varpi(g_{2j}) \\ S(u_1, u_2) &= \sum_{i=1}^{k} f(g_{1i}|G(u_2), h) \\ &= \sum_{i=1}^{k}\frac{1}{l}\sum_{j=1}^{l} K_h(g_{1i}, g_{2j}) \end{aligned} \tag{5}$$

where the grid representations of u_1 and u_2 are $G(u_1) = \{(g_{11}, \varpi(g_{11})), (g_{12},\ \varpi(g_{12})), \cdots, (g_{1k}, \varpi(g_{1k}))\}$ and $G(u_2) = \{(g_{21}, \varpi(g_{21})), (g_{22}, \varpi(g_{22})), \cdots, (g_{2l}, \varpi(g_{2l}))\}$, respectively. $(g_{1i} - g_{2j})$ denotes the Euclidean distance between the center coordinates of cells g_{1i} and g_{2j}. Compared with the naive evaluation of KDE, the grid representation is a coarse-grained method, where the grid cell is the basic unit that may contain many points. Note that, implementing KDE with grid representation is able to: reduce the number of kernel function evaluation; and alleviate the data missing problem. In detail, the grid cell based KDE has the following two advantages.

Advantage 1: After dividing the space and representing user accounts with grid cells, the number of kernel function $K(\cdot)$ calculated in Eq. (5) is significantly less than that in Eq. (3). This is because, we have $|G(u)| \leq |R_u|$ for each user account u. In the worst case, $|G(u)| = |R_u|$, where each grid cell visited by u only contains one check-in record. In the best case, $|G(u)| = 1$, where all records fall into one grid cell. In Fig. 2, $|R_{u_1}| = |R_{u_2}| = 5$, $|G(u_1)| = |G(u_2)| = 4$. Then, the number of $K(\cdot)$ calculated in Eq. (3) is $|R_{u_1}| \cdot |R_{u_2}| = 25$, which is larger than that in Eq. (5), i.e., $|G(u_1)| \cdot |G(u_2)| = 16$.

Advantage 2: Such grid-based method is able to further alleviate the data missing issue introduced in Section I. Although a user is most likely to check-in at different locations and generate different numbers of check-in records on two location-based social networks, an individuals mobility usually centers at different personal geographical regions, and the probability that the user visits these regions tends to be similar across two LBSNs [15][10]. For example, the user may have 100 records in Facebook, where 30 records are generated in the home region and 40 records created in the work region, thus the corresponding confidences are 0.3 and 0.4, respectively. In Twitter, he may have similar confidences in the home region (280-310 records) and work region (390-420 records), where the number of total records is 1000. Such a phenomenon makes users have similar confidences in corresponding grid cells across different platforms.

To further improve the efficiency of KDE evaluation, we propose a pruning strategy to significantly reduce down the number of cell-pair comparison and avoid the intensive evaluation of the kernel functions. Specifically, to compute $f(g_{1i}|G(u_2), h)$ in Eq. (5), we first construct a square region centered at grid cell g_{1i}, and only the cells g_{2j} within the square region will be considered in computing function $f(\cdot)$, as other grid cells of u_2 are far away from g_{1i} and their effect or contributions can be ignorable. The square region is composed of $k \times k$ grid cells and centers at g_{1i}, and our approach achieves its best performance with $k = 3$ in the experiment. The results reported in Section VII demonstrate that this strategy is feasible and efficient. By integrating this pruning strategy, we redefine the function $f(g_{1i}|G(u_2), h)$ as follows:

$$f(g_{1i}|G(u_2), h) = \frac{1}{z}\sum_{j=1}^{z} K_h(g_{1i}, g_{2j}) \tag{6}$$

where z is the number of grid cells in the square region.

VI. Grid Cell Weight

Intuitively, the popular places, such as shopping mall, cafeteria, and cinema, are more attractive and more likely to be visited by many people than personal private places, such as home and office. This phenomenon leads to the low peculiarity of popular places, i.e., these places are useless for distinguishing users from each other. In contrast, the personal private places visited by less people are more discriminative, based on which we are more likely to identify the accounts of the same user [18]. In other words, the importance of different places are different. To achieve user account linkage with higher accuracy, we need to assign different weights to the grid cells to indicate their importance. Inspired by [8], we propose to use Entropy from information theory to compute the importance of each grid cell in this section.

A. Shannon Entropy Based Grid Cell Weight

Entropy is the expected value of the information contained in each message in relation to the importance of the message [1]. Shannon entropy is a typical entropy, which is a common tool and has been widely used in various applications [8][39]. Given a set of user accounts $U = \{u_1, u_2, \cdots, u_n\}$ in a specific platform, the Shannon entropy of each grid cell g is defined as:

$$H(g) = -\sum_{i=1}^{n} \frac{N_{u_i}(g)}{|R_{u_i}|} \log \frac{N_{u_i}(g)}{|R_{u_i}|} \quad (7)$$

where $N_{u_i}(g)$ is the number of records generated by u_i in current cell g, R_{u_i} is the set of historical check-in records of u_i. Based on the discussion in [8], we know that a high value of the grid cell entropy indicates the high popularity and the low discriminative ability of g. On the other hand, a low entropy implies the high privacy and differentiating power of g. As our goal is to highlight discriminative cells with large weights and penalize popular cells with low weights, we define the grid cell weight $\omega(g)$ as follows:

$$\omega(g) = \exp\big(-H(g)\big) \quad (8)$$

Based on the computed grid weights, we upgrade the grid representation of user accounts as $G(u_1) = \{(g_{11}, \varpi(g_{11}), \omega(g_{11})), (g_{12}, \varpi(g_{12}), \omega(g_{12})), \cdots, (g_{1k}, \varpi(g_{1k}), \omega(g_{1k}))\}$ and $G(u_2) = \{(g_{21}, \varpi(g_{21}), \omega(g_{21})), (g_{22}, \varpi(g_{22}), \omega(g_{22})), \cdots, (g_{2l}, \varpi(g_{2l}), \omega(g_{2l}))\}$. Then, we redefine the Gaussian kernel function $K(\cdot)$ and the similarity $S(u_1, u_2)$ between u_1 and u_2,

$$\begin{aligned} K_h(g_{1i}, g_{2j}) &= \frac{1}{2\pi h} \exp\Big(-\frac{(g_{1i}-g_{2j})^2}{2h^2}\Big)\cdot \\ &\quad \varpi(g_{1i})\varpi(g_{2j})\omega(g_{1i})\omega(g_{2j}) \\ S(u_1, u_2) &= \sum_{i=1}^{k} f(g_{1i}|G(u_2), h) \\ &= \sum_{i=1}^{k} \frac{1}{z} \sum_{j=1}^{z} \frac{1}{2\pi h} \exp\Big(-\frac{(g_{1i}-g_{2j})^2}{2h^2}\Big)\cdot \\ &\quad \varpi(g_{1i})\varpi(g_{2j})\omega(g_{1i})\omega(g_{2j}) \end{aligned} \quad (9)$$

Different from the kernel functions proposed in Eq. (2) and Eq. (5), we take both the confidence $\varpi(g)$ and the grid cell weight $\omega(g)$ into account. This method is more reasonable than just considering one of the two factors. On one hand, if we only consider the confidence, the popular grid cells which are frequently visited by most users, will play a more important role than personal private cells. Then, it is hard to link the accounts of the same user in this case. On the other hand, the grid cell weight based method will give high importance to the outliers, which are prevalent in spatio-temporal database and happen to be visited by few users. The large weights of outliers also bring the great challenge for user account linkage. Only considering the confidence and weight simultaneously can achieve good performance: the popular grid cells with high confidence will be assigned with small weights, and the outliers with large weights will tend to have small confidence values.

B. Renyi Entropy Based Grid Cell Weight

Renyi entropy is a generalized version of Shannon entropy and it's defined as follows in our application:

$$H(g) = \frac{1}{1-q} \log \sum_{i=1}^{n} \Big(\frac{N_{u_i}(g)}{|R_{u_i}|}\Big)^q \quad (10)$$

From this definition, we can see that Shannon entropy is a special case of Renyi Entropy (i.e., $q = 1$). The adjustable q makes Renyi entropy much more expressive and flexible compared with Shannon entropy. Following the study in [8], the parameter q indicates entropy's sensitivity to the number $N_{u_i}(g)$. Specifically,

- If $q > 1$, the entropy rewards the higher value of $N_{u_i}(g)$.
- If $q < 1$, the entropy penalizes the higher value of $N_{u_i}(g)$.
- If $q = 1$, even though Eq. (10) is undefined at $q = 1$, its limit exists when $q \to 1$ and becomes the Shannon entropy, and the proof is presented in [8].

According to the Renyi entropy, we give the new definition of the grid cell weight $\omega(g)$:

$$\begin{aligned} \omega(g) &= \exp\big(-H(g)\big) \\ &= \exp\Big(-\frac{1}{1-q} \log \sum_{i=1}^{n} \Big(\frac{N_{u_i}(g)}{|R_{u_i}|}\Big)^q\Big) \\ &= \Big(\sum_{i=1}^{n} \Big(\frac{N_{u_i}(g)}{|R_{u_i}|}\Big)^q\Big)^{\frac{1}{q-1}} \end{aligned} \quad (11)$$

Next, we calculate the new Gaussian kernel function $K(\cdot)$ and $S(u_1, u_2)$ by replacing the weight in Eq. (9) with the weight computed in Eq. (11).

By constructing grid structure and computing grid cell weight based on the Renyi entropy, we design an algorithm GCRE in Algorithm 1. It contains the calculation of grid representation (line 2-3), user pair pruning (line 6-7), similarity measure (line 10), and result process (line 11-13). Note that, $CC(G(u_1), G(u_2))$ denotes the total number of grid cells that have been visited by u_1 and u_2 in the $k \times k$ square region.

Continue the example in Fig. 2, we can obtain $G(u_1) = \{(2, 0.2, 1.2), (73, 0.2, 0.55), (88, 0.4, 1.11), (38, 0.2, 0.55)\}$ and $G(u_2) = \{(24, 0.2, 1.2), (73, 0.2, 0.55), (78, 0.4, 1.11), (38, 0.2, 0.55)\}$ based on Eq. (11), where we set $q = 0.1$, $k = 3$, and $h = 300m$. Then, we have $CC(G(u_1), G(u_2)) = 4$, and obtain $S(u_1, u_2) = 0.00159$ based on Eq. (9).

[1]https://en.wikipedia.org/wiki/Entropy

Algorithm 1: Grid Cell and Renyi Entropy Based Algorithm (GCRE)

Data: two user account sets $U_1 = \{u_{11}, u_{12}, \cdots, u_{1m}\}$ and $U_2 = \{u_{21}, u_{22}, \cdots, u_{2n}\}$
Result: a set of user account pairs O

```
for each user account in U_1 and U_2 do
    calculate the grid confidence based on Eq. (4);
    compute the grid cell weight based on Eq. (11);
end
for each use pair {(u_1i, u_2j) | u_1i ∈ U_1, u_2j ∈ U_2} do
    if CC(G(u_1i), G(u_2j)) = 0 then
        prune user pair (u_1i, u_2j);
    end
    else
        compute S(u_1i, u_2j) based on G(u_1) and G(u_2);
        if S(u_1i, u_2j) ≥ S_Δ then
            add (u_1i, u_2j) into O;
        end
    end
end
return O;
```

VII. Experiment Study

Extensive experiments are conducted in this section. First, we describe the experiment setup, which contains dataset introduction, baseline algorithms presentation, and evaluation metrics discussion. Then, the effectiveness and efficiency of the proposed methods are reported.

TABLE II
Parameter Settings

Parameter	Value
k	1, **3**, 5, 7, 9
q	0, 0.05, **0.1**, 0.15, 0.2, 0.25
h	100, 200, **300**, 400, 500, 600
$grid$	3000, 5000, 7000, **9000**, 11000, 13000
S_Δ(FS-TW)	0.0006, 0.0008, **0.001**, 0.0012, 0.0014, 0.0016
S_Δ(IT-TW)	0.0001, 0.0002, 0.0003, **0.0004**, 0.0005, 0.0006

A. Experiment Setup

Two cross-domain real datasets are used to conduct experiments. The parameter settings are presented in Table II, and statistics of datasets are reported in Table III.

TABLE III
Statistics of Datasets

Dataset	Domain	Users	Records	Date Range
FS-TW	Foursquare	862	13177	2006.10 - 2012.11
	Twitter	862	174618	2008.10 - 2012.11
IT-TW	Instagram	1717	337934	2010.10 - 2013.09
	Twitter	1717	447366	2010.09 - 2015.04

Foursquare-Twitter (FS-TW). Foursquare and Twitter are two widely used social networks, where users can post statuses associated with location information. To investigate the performance of the proposed approach in linking cross-domain user accounts, we use the dataset provided by [13][10], where they select users that have more than one record presented in both platforms. The dataset contains 862 users with 13177 Foursquare records and 174618 Twitter records, where a record is in the form of (userid, latitude, longitude, time-stamp). Note that we use the lower left and upper right to denote the location range of a dataset. Location ranges of Foursquare and Twitter are $\{(-89.99, -159.67), (71.38, 176.19)\}$ and $\{(-85.54, -176.05), (84.11, 178.03)\}$, respectively.

Instagram-Twitter (IT-TW). Instagram is another popular photo-sharing application and service, where users can share pictures and videos with location information through mobile, desktop, laptop, and tablet. To link the user accounts across Instagram and Twitter with location data, we use the dataset processed by [10]. Similarly, each user of the dataset has check-in records generated in both platforms. The dataset contains 1717 users with 337934 Instagram records and 447366 Twitter records, where a record is stored as a tuple (userid, latitude, longitude, time-stamp). Location ranges of Instagram and Twitter are $\{(-53.16, -170.27), (71.03, 177.43)\}$ and $\{(-74.05, -159.76), (71.03, 175.81)\}$, respectively.

B. Compared Methods.

We compare the performance of our method with several state-of-the-art location based user account linkage approaches. Although existing methods [23][25][28] also work on user account linkage, their results are not comparable here, as they use different input data, such as text messages, user profile, and language style.

- GRID: The first method is based on of the work proposed by [14], where the top-$p\% = 15\%$ grid cells with the maximum density are returned to denote a user. Based on these grid cells, we develop a method to measure the similarity between a user account pair. Assume the returned top-$p\% = 15\%$ grid cells of u_1 and u_2 are $R_1 = \{(r_{11}, f(r_{11})), (r_{12}, f(r_{12})), \cdots, (r_{1n}, f(r_{1n}))\}$ and $R_2 = \{(r_{21}, f(r_{21})), (r_{22}, f(r_{22})), \cdots, (r_{2m}, f(r_{2m}))\}$, respectively. Then, we define the similarity $S(u_1, u_2)$ as follows:
$$S(u_1, u_2) = \sum_{r_{1i} \in R_1}^{m} \sum_{r_{2j} \in R_2}^{n} \frac{|r_{1i} \cap r_{2j}|}{|r_{1i} \cup r_{2j}|} \cdot min(f(r_{1i}), f(r_{2j}))$$
where $f(r_{ij})$ denotes the density of the j-th grid cell.
- BIN: The second method is proposed by [10], where each action record in region l during time interval t is associated with bin (l, t). The similarity between u_1 and u_2 is defined as:
$$S(u_1, u_2) = \sum_{t \in T} \sum_{l \in L} S(u_1, u_2, l, t)$$
where the similarity $S(u_1, u_2, l, t)$ in the bin (l, t) is:
$$\frac{P[A_1(u_1, l, t) = a_1 \wedge A_2(u_2, l, t) = a_2 | \sigma_I(u_1) = u_2]}{P[A_1(u_1, l, t) = a_1] \cdot P[A_2(u_2, l, t) = a_2]}$$
where $A_i(u_i, l, t)$ is the number of actions in the given bin (l, t) of u_i, $P[\cdot]$ is the likelihood, and $\sigma_I(u_1) = u_2$ means u_1 and u_2 are the same user.

- DG: The third method is proposed by [18], where a density-based clustering method (DP) is used to extract the stay regions of a user, and a Gaussian Mixture Model (GMM) based approach is proposed to model users' temporal behaviors. Then, the similarity between a user account pair is measured based on these features.
- EPOCH: The fourth method is proposed by [32], where the input data is temporally partitioned into equal sized time intervals, and a k-nearest neighbor classification approach is employed to link cross-domain users.
- GS: The fifth method is a variant of the approach proposed by [31]. Based on the idea of [31], $\{(g_1, o_1), \cdots, (g_m, o_m)\}$ is used to denote the observed co-occurrences of two users, where o_i $(1 \le i \le m)$ denotes the corresponding frequency, and the weight of g_i is defined as:

$$\omega(g_i) = f_s(o_i) = \frac{\eta}{1 + e^{-\gamma o_i}} - \frac{\eta}{2}$$

where η and γ are set to 16 and 0.2, respectively [31]. Then, we can give the similarity $S(u_1, u_2)$ as follows:

$$S(u_1, u_2) = \sum_{i=1}^{m} \omega(g_i) \cdot \frac{o_i}{|R_{u_1}|} \cdot \frac{o_i}{|R_{u_2}|}$$

- GKR-KDE: Our approach is denoted as GKR-KDE, where we divide the space into grid cells, construct the $k \times k$ square region, and calculate the grid cell weight based on Renyi entropy.

To further validate the benefits brought by grid index, square region, Shannon entropy, and Renyi entropy, we design baseline methods: **KDE**, **G-KDE**, **GK-KDE**, and **GKS-KDE**, the properties of which are presented in Table IV. **KDE** is the first simplified version of GKR-KDE, where the kernel density estimation (KDE) is directly used to measure the similarity $S(u_1, u_2)$, i.e., the function defined in Eq. (3). **G-KDE** is the second simplified version of GKR-KDE, where each user is represented by a set of grid cells, and the similarity $S(u_1, u_2)$ is defined in Eq. (5). **GK-KDE** is the third simplified version of GKR-KDE, where only the grid cells that fall into the $k \times k$ square region are considered, and the function is defined in Eq. (6). The method **GKS-KDE** calculates the grid confidence based on Eq. (4), and assigns different weights to grid cells with Shannon entropy. Then, it defines a different similarity $S(u_1, u_2)$ in Eq. (9).

TABLE IV
PROPERTY OF METHODS

	Grid Index	$k \times k$ Square Region	Shannon	Renyi
KDE	×	×	×	×
G-KDE	√	×	×	×
GK-KDE	√	√	×	×
GKS-KDE	√	√	√	×
GKR-KDE	√	√	×	√

C. Evaluation Metrics

To evaluate the effectiveness of above algorithms, we use precision, recall, and F1. Given two sets of user accounts $U_1 = \{u_{11}, u_{12}, \cdots, u_{1m}\}$ and $U_2 = \{u_{21}, u_{22}, \cdots, u_{2n}\}$, we return the user account pair (u_{1i}, u_{2j}) with $S(u_{1i}, u_{2j}) \ge S_\Delta$. The precision is defined as the fraction of user account pairs contained by the returned result that are correctly linked, and the recall is defined as the fraction of the actual linked user account pairs contained by the returned result [18],

$$Recall = \frac{\alpha}{\beta}, \; Precision = \frac{\alpha}{\gamma}$$
$$F1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

where γ is the number of actual linked user account pairs in the ground truth, β is the number of returned user account pairs, and α is the number of actual linked user account pairs in the returned result. Additionally, to evaluate the efficiency of the proposed algorithms, we compare the time cost of them.

Note that we report the best performance of baseline methods GRID, BIN, DG, EPOCH, and GS in both datasets FS-TW and IT-TW in the sequel.

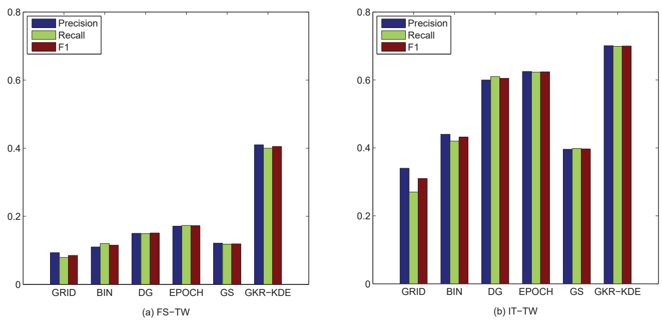


Fig. 3. Performance of proposed approaches in FS-TW and IT-TW

D. Effectiveness Evaluation

The performance of different methods are reported in Fig. 3, where the precision, recall, and F1 are presented. As expected, all methods perform better than GRID, since only the common grid cells with high density are considered while measuring the similarity between a user account pair in GRID. All of BIN, DG, EPOCH, and GS do not perform well as they did in [30][18][31][32]. This is because we have proposed a novel evaluation metric, where all user account pairs with similarity larger than S_Δ are returned. Such metric makes our approach become more general and applicable to many applications, especially when two datasets have different numbers of user accounts and there exist many-to-many mappings. Compared with GRID, BIN, DG, EPOCH, and GS, our method GKR-KDE performs best, due to the following reasons. On one hand, the kernel density estimation based similarity measurement is able to tackle the challenges data sparsity and data missing introduced in Section I, since we use a set of grid cells with corresponding confidence to denote a user. On the other hand, we calculate the grid cell weight based on Renyi entropy, where the important and discriminative grid cells visited by few visitors are highlighted. In contrast, the popular grid cells with large entropy are assigned with small weights, due to the low discrimination of them. Additionally, we find that all methods have better performance in the dataset IT-TW. This is because, it is a denser dataset compared with FS-TW, where users have more check-in records. Naturally, we are more likely to model real behaviors for users across Instagram and Twitter, and find the actual linked user accounts.

TABLE V
AVERAGE RUNNING TIME (S) OF DIFFERENT METHODS

	GRID	BIN	DG	EPOCH	GS	GKR-KDE
FS-TW	24.77	4.01	3.13	2.88	2.36	**0.251**
IT-TW	120.01	46.63	4.58	3.96	3.57	**0.232**

E. Efficiency Evaluation

The average running time is another important factor needed to be considered while linking cross-domain user accounts. As seen in Table V, we report the average running time of GRID, BIN, DG, EPOCH, GS, and GKR-KDE in different datasets. Obviously, GRID is the most time consuming method, as it needs to take all historical records into account while computing the density of a grid cell. Calculating the density for all grid cells before returning the top-$p\%$ ones leads to the large time cost of the method. The second method BIN is also time consuming, since we need to measure the similarity $S(u_1, u_2, l, t)$ in each bin (l, t). For DG, we need to spend much time to extract stay regions and time clusters, especially the weight calculation of these features. For EPOCH, after finding the top-k frequent transitions, applying the k-nearest neighbor classification approach to classify the user of a new trace is time consuming. In GS, we need to spend much time to find the common grid cells of two users, and calculate the user similarity based on these cells. Without surprise, our method GKR-KDE outperforms all baseline approaches with significant less time cost. This is because we only consider the grid cells with $\varpi(g) > 0$ while measuring the user account similarity with grid representation, i.e., the Property 1 and Property 2 introduced in Section V. In addition, we have constructed a $k \times k$ square region for each grid cell, where only the grid cells that fall into the region are considered in Eq. (6). Such action further reduces the number of grid cells to be considered. Consequently, GKR-KDE is much more efficient than other approaches.

F. Impact of Different Factors

To explore the benefits of integrating grid index, $k\times k$ square region, Shannon entropy, and Renyi entropy, we compare the average running time of KDE, G-KDE, GK-KDE, GKS-KDE, and GKR-KDE in Table VI, while the precision, recall, and F1 of them are reported in Fig. 4. In addition, to study the scalability of GKR-KDE, we randomly select 13251 users with 1140780 locations (Gowalla1), 22478 users with 2135643 locations (Gowalla2), and 30578 users with 2969148 locations (Gowalla3) from the dataset Gowalla provided by [40]. We split each dataset of Gowalla1, Gowalla2, and Gowalla3 into two parts to simulate the cross-domain user linkage. The time cost of these methods is presented in Table VI.

From the results in Table VI, we observe that the time cost of them can be roughly divided into three levels. The first level, KDE spends much more time than others since it needs to calculate the Gaussian kernel function $K(\cdot)$ between any two records according to Eq. (2). The second level, the grid index based method G-KDE needs less running time than KDE, as the cardinality of grid representation of each user is smaller than that of his/her check-in records. The example in Fig. 2 and the Advantage 1 in Section V detailedly illustrate the superiority of constructing grid index. The third level, GK-KDE, GKS-KDE, and GKR-KDE have similar and the least time cost, due to the existing of $k \times k$ square region, where many grid cells are omitted according to Eq. (6). Meanwhile, the similar time cost of them also means that considering Shannon entropy or Renyi entropy do not influence the efficiency. The high efficiency (less than 1s) also demonstrates the high scalability of GKR-KDE.

TABLE VI
AVERAGE RUNNING TIME (S) OF OUR METHODS

	KDE	G-KDE	GK-KDE	GKS-KDE	GKR-KDE
FS-TW	10.09	1.33	0.241	0.254	**0.251**
IT-TW	23.13	3.44	0.233	0.236	**0.232**
Gowalla1	220.51	8.31	0.257	0.268	**0.221**
Gowalla2	453.65	16.52	0.488	0.471	**0.469**
Gowalla3	685.48	25.98	0.675	0.698	**0.682**

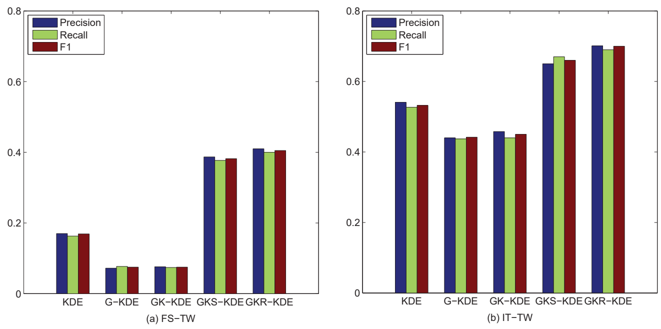


Fig. 4. Impact of different factors on effectiveness

The impact of different factors on effectiveness is shown in Fig. 4, where we observe that GKR-KDE consistently outperforms the four baselines in both datasets, indicating that GKR-KDE benefits from synchronously considering three factors in a joint way. KDE performs better than G-KDE, as all records are directly used to measure the similarity between two user accounts, yet it also leads to the huge time cost presented in Table VI. The similarity results of G-KDE and GK-KDE denote that constructing the $k \times k$ square region has no influence on effectiveness. An individual's mobility usually centers at some specific regions, omitting the grid cells out of these regions is reasonable, since our goal is to find the actual linked user account pair (u_{1i}, u_{2j}) instead of precisely measuring the similarity between two user accounts. In addition, the high performance of GKS-KDE and GKR-KDE reveals the importance of calculating the grid cell weight based on entropy, where the important and discriminative grid cells are highlighted. Of course, GKR-KDE outperforms GKS-KDE also means Renyi entropy is more appropriate for determining the popularity of a grid cell. Assigning the grid

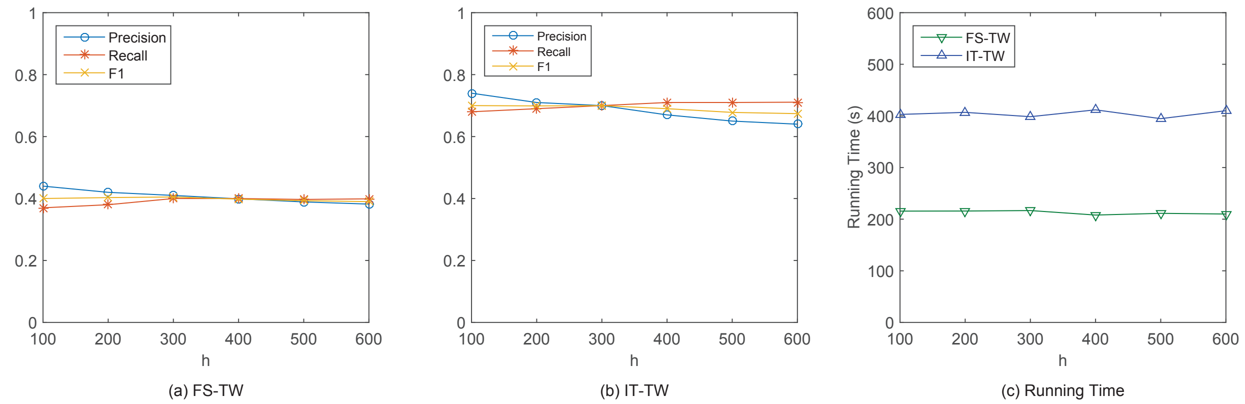


Fig. 5. Performance of GKR-KDE w.r.t. varied h

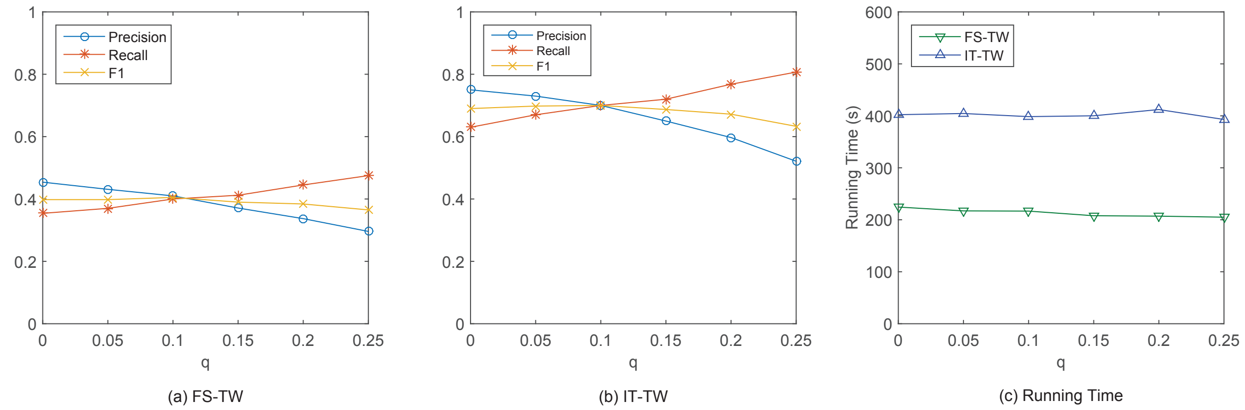


Fig. 6. Performance of GKR-KDE w.r.t. varied q

cell weight based on such entropy can further improve the effectiveness of cross-domain user account linkage.

G. Impact of Parameters

To obtain the best performance of GKR-KDE, tuning parameters, such as q, bandwidth h, grid granularity, square region length k, and the similarity threshold S_Δ, is of critical importance. We therefore study the impact of different parameters in this section.

Varying bandwidth h**.** From the results in Fig. 5(a) and (b), we observe that the effectiveness of GKR-KDE is sensitive to the value of bandwidth h, producing a larger and larger user account similarity $S(u_1, u_2)$ according to Eq. (9) with the increase of h. Thus, it leads to the increase of recall as many user account pairs are returned, yet it also leads to the decrease of precision as many user account pairs contained by the returned result are not actual linked. To balance precision and recall, we set $h = 300\ m$ in both FS-TW and IT-TW. Additionally, the running time in Fig. 5(c) denotes that the efficiency of GKR-KDE and the value of h is not relevant.

Varying q**.** As discussed in Section VI-B, the elegance of using the Renyi entropy lies inside the parameter q. According to Eq. (11), we can obtain a larger and larger grid cell weight $\omega(g)$ with the increase of q. Then, it leads to the increase of user account similarity when other parameters are fixed. Just like the results of varying bandwidth h, the precision and recall have opposite change in Fig. 6(a) and (b), and the reasons behind the phenomenon are similar. To balance precision and recall, we set $q = 0.1$ in both FS-TW and IT-TW. During the calculation of user account similarity, the number of grid cells to be considered is equal for different q, thus the running time in each dataset almost does not change in Fig. 6(c).

Varying grid granularity. The grid granularity is another important parameter of GKR-KDE, where the selection of such parameter has two extremes: 1) extreme coarse granularity, the whole space is regarded as one grid cell that contains all check-in records; 2) extreme fine granularity, where each record is a grid cell and the method degrades into the naive kernel density estimation. Obviously, a too large or too small grid granularity is not appropriate for balancing precision and recall, as presented in Fig. 7(a) and (b). The time cost of GKR-KDE is sensitive to the grid granularity, since the increase of which means the records of a user may fall into more grid cells and the cardinality of grid representation of the user becomes larger. As a result, we have observed the increase of the running time of GKR-KDE while varying the grid granularity from 3000×3000 to 13000×13000 in Fig. 7(c). Compared with using actual locations to measure the user similarity, the grid based GKR-KDE may lose some information, the running time see a very clear increase while the effectiveness is relative steady. Taking various factors into account, we divide the space

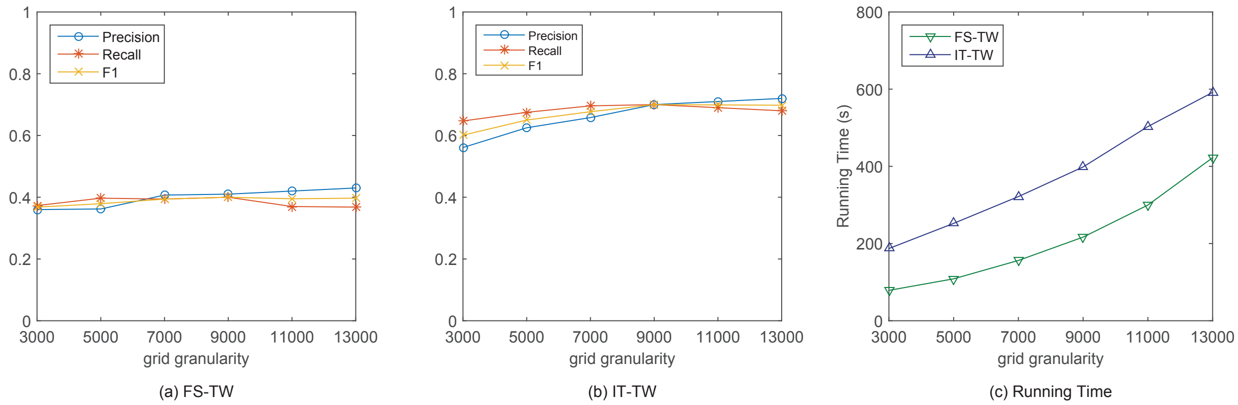


Fig. 7. Performance of GKR-KDE w.r.t. varied grid granularity

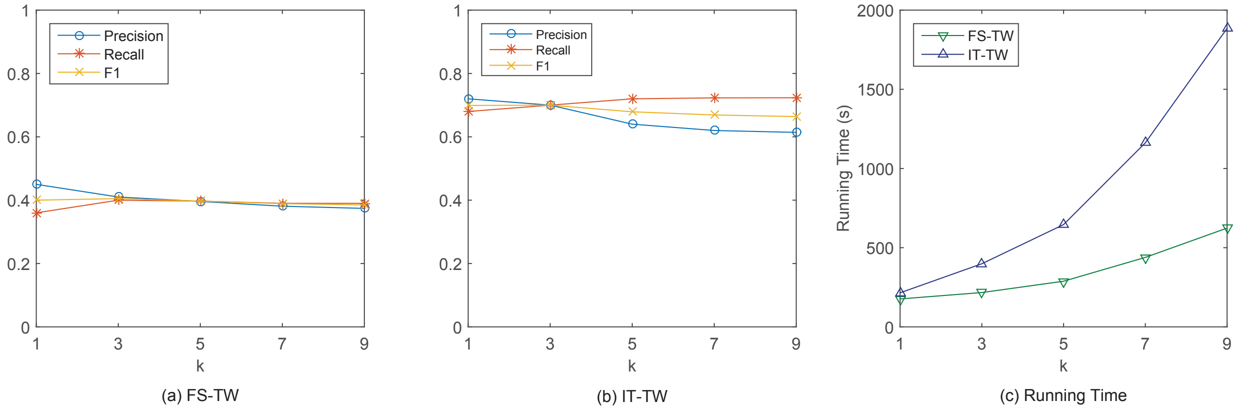


Fig. 8. Performance of GKR-KDE w.r.t. varied k

into 9000×9000 grid cells.

Varying k. The square region length k determines how many grid cells needed to be considered in function $f(\cdot)$. The results in Fig. 8 reveal that $k = 3$ is a good choice, as which can balance precision and recall and has a acceptable time cost. Given $k = 1$, the method GKR-KDE degrades into comparing the common grid cells of two user accounts. Thus, it leads to the decrease of recall. Given a too large k (e.g., $k =$ grid granularity), GKR-KDE needs to take all grid cells into account, where the time cost is unacceptable.

Varying $S_\triangle$**.** In real scenarios, the datasets across different platforms may have different numbers of user accounts and there may exist many many-to-many mappings, thus we propose a general method where the user account pairs $\{(u_{1i}, u_{2j})|u_{1i} \in U_1, u_{2j} \in U_2\}$ with $S(u_{1i}, u_{2j}) \geq S_\triangle$ are returned, as presented in Algorithm 1. Obviously, the effectiveness of our method is very sensitive to the selection of $S_\triangle$. On one hand, many actual linked user account pairs are filtered with a too large $S_\triangle$. On the other hand, the returned results may contain too many unmatched user account pairs if given a small $S_\triangle$. To balance precision and recall, and consider the characteristics of different datasets, we set $S_\triangle = 0.001$ and $S_\triangle = 0.0003$ in FS-TW and IT-TW, respectively. The results shown in Fig. 9(c) and (d) denote that the similarity threshold $S_\triangle$ does not influence the efficiency of GKR-KDE.

VIII. Conclusion

Linking users across different platforms with location data has received great attention, due to the increasing availability of spatio-temporal data, and the wide applications of the study, such as cross-domain recommendation and prediction. To achieve effective and efficient user account linkage, we have proposed several novel methods. Firstly, to tackle the data sparsity, we develop a kernel density estimation based approach to directly measure the similarity (KDE) between two user accounts instead of comparing the same places visited by them. Secondly, we construct a grid index to improve the efficiency of KDE and tackle the data missing problem, where each user is represented by a set of grid cells. Thirdly, to further improve the effectiveness of our method, we calculate the weight for each grid cell based on its entropy, where the individual grid cells are highlighted with large weight, yet the popular ones visited by many users are lightened due to the low discrimination of them. The experiments conducted on two real datasets demonstrate the superiority of our method over the state-of-the-art approaches.

Acknowledgments. The work is partially supported by ARC Discovery Early Career Researcher Award (DE160100308) and ARC Discovery Project (DP170103954). It is partially supported by the National Natural Science Foundation

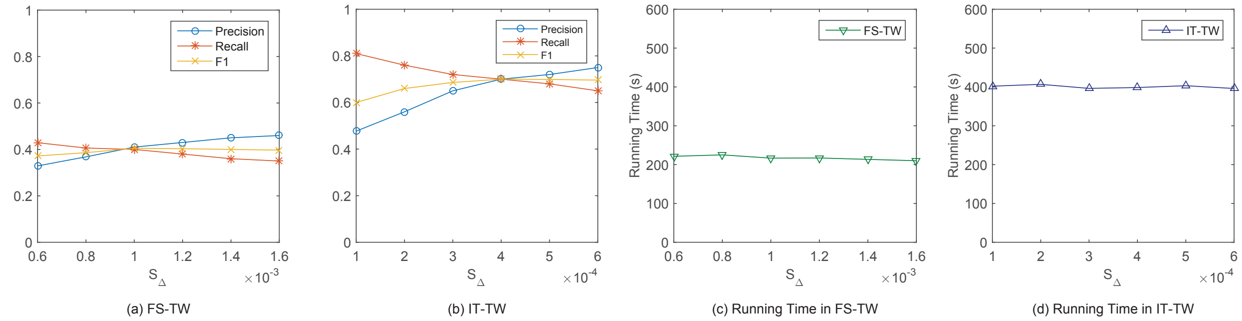


Fig. 9. Performance of GKR-KDE w.r.t. varied similarity threshold S_Δ

of China under Grant Nos. 61572335, the Natural Science Foundation of Jiangsu Province of China under Grant No. BK20151223, and Collaborative Innovation Center of Novel Software Technology and Industrialization, Jiangsu, China.

REFERENCES

[1] H. Yin, X. Zhou, B. Cui, K. Zheng, and N. Q. V. Hung, "Adapting to user interest drift for poi recommendation," *TKDE*, vol. 28, no. 10, pp. 2566–2581, 2016.

[2] H. Yin, Z. Hu, X. Zhou, H. Wang, K. Zheng, N. Q. V. Hung, and S. W. Sadiq, "Discovering interpretable geo-social communities for user behavior prediction," in *ICDE*, 2016, pp. 942–953.

[3] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *CIKM*, 2016, pp. 15–24.

[4] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, "Spatial-aware hierarchical collaborative deep learning for poi recommendation," *TKDE*, vol. 29, no. 11, pp. 2537–2551, 2017.

[5] Y. Tong, L. Chen, Z. Zhou, H. V. Jagadish, L. Shou, and W. Lv, "Slade: A smart large-scale task decomposer in crowdsourcing," *TKDE*, no. 99, pp. 1–14, 2018.

[6] H. Gao and H. Liu, "Data analysis on location-based social networks," in *Mobile social networking*, 2014, pp. 165–194.

[7] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq, "Joint modeling of user check-in behaviors for real-time point-of-interest recommendation," *TOIS*, vol. 35, no. 2, pp. 1–44, 2016.

[8] H. Pham, C. Shahabi, and Y. Liu, "Ebm: an entropy-based model to infer social strength from spatiotemporal data," in *SIGMOD*, 2013, pp. 265–276.

[9] M. Lichman and P. Smyth, "Modeling human location data with mixtures of kernel densities," in *KDD*, 2014, pp. 35–44.

[10] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *WWW*, 2016, pp. 707–719.

[11] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in foursquare." *ICWSM*, vol. 11, pp. 70–573, 2011.

[12] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou, "Spore: A sequential personalized spatial item recommender system," in *ICDE*, 2016, pp. 954–965.

[13] J. Zhang, X. Kong, and P. S. Yu, "Transferring heterogeneous links across location-based social networks," in *WSDM*, 2014, pp. 303–312.

[14] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye, "Mining periodic behaviors for moving objects," in *KDD*, 2010, pp. 1099–1108.

[15] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann, "Who, where, when and what: discover spatio-temporal topics for twitter users," in *KDD*, 2013, pp. 605–613.

[16] M. P. Wand, "Fast computation of multivariate kernel estimators," *Journal of Computational and Graphical Statistics*, vol. 3, no. 4, pp. 433–445, 1994.

[17] U. Lopez-Novoa, J. Sáenz, A. Mendiburu, and J. Miguel-Alonso, "An efficient implementation of kernel density estimation for multi-core and many-core architectures," *Journal of High Performance Computing Applications*, vol. 29, no. 3, pp. 331–347, 2015.

[18] W. Chen, H. Yin, W. Wang, L. Zhao, W. Hua, and X. Zhou, "Exploiting spatio-temporal user behaviors for user linkage," in *CIKM*, 2017.

[19] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu, "User identity linkage across online social networks: A review," *SIGKDD Explorations*, vol. 18, no. 2, pp. 5–17, 2017.

[20] R. Zafarani and H. Liu, "Connecting corresponding identities across communities." *ICWSM*, vol. 9, pp. 354–357, 2009.

[21] J. Vosecky, D. Hong, and V. Y. Shen, "User identification across social networks using the web profile and friend network." *Journal of Web Applications*, vol. 2, no. 1, pp. 23–34, 2010.

[22] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying users across social tagging systems." in *ICWSM*, 2011.

[23] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *KDD*, 2013, pp. 41–49.

[24] O. Peled, M. Fire, L. Rokach, and Y. Elovici, "Entity matching in online social networks," in *Social Computing*, 2013, pp. 339–344.

[25] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling," in *KDD*, 2014, pp. 51–62.

[26] J. She, Y. Tong, and L. Chen, "Utility-aware social event-participant planning," in *SIGMOD*, 2015, pp. 1629–1643.

[27] J. She, Y. Tong, L. Chen, and C. C. Cao, "Conflict-aware event-participant arrangement and its variant for online setting," *TKDE*, vol. 28, no. 9, pp. 2281–2295, 2016.

[28] X. Mu, F. Zhu, E.-P. Lim, J. Xiao, J. Wang, and Z. Zhou, "User identity linkage by latent user space modelling," in *KDD*, 2016, pp. 1775–1784.

[29] X. Han, L. Wang, L. Xu, and S. Zhang, "Social media account linkage using user-generated geo-location data," in *ISI*, 2016, pp. 157–162.

[30] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *WWW*, 2016, pp. 707–719.

[31] W. Cao, Z. Wu, D. Wang, J. Li, and H. Wu, "Automatic user identification method across heterogeneous mobility data sources," in *ICDE*, 2016, pp. 978–989.

[32] E. Seglem and p. y. Andreas Züfle and Jan Stutzki and Felix Borutta and Evgheniy Faerman and Matthias Schubert, booktitle=SSTD, "On privacy in spatio-temporal data: User identification using microblog data."

[33] D. W. Scott and S. J. Sheather, "Kernel density estimation with binned data," *Communications in Statistics-Theory and Methods*, vol. 14, no. 6, pp. 1353–1359, 1985.

[34] B. W. Silverman, *Density estimation for statistics and data analysis*, 1986, vol. 26.

[35] J.-D. Zhang and C.-Y. Chow, "igslr: personalized geo-social location recommendation: a kernel density estimation approach," in *GIS*, 2013, pp. 334–343.

[36] M. Hulden, M. Silfverberg, and J. Francom, "Kernel density estimation for text-based geolocation." in *AAAI*, 2015, pp. 145–150.

[37] S. Hasan, X. Zhan, and S. V. Ukkusuri, "Understanding urban human activity and mobility patterns using large-scale location-based data from online social media," in *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, 2013.

[38] P. Zhang, M. Deng, and N. Van de Weghe, "Clustering spatio-temporal trajectories based on kernel density estimation," in *ICCSA*, 2014, pp. 298–311.

[39] K. V. Bulusu and M. W. Plesniak, "Shannon entropy-based wavelet transform method for autonomous coherent structure identification in fluid flow field data," *Entropy*, vol. 17, no. 10, pp. 6617–6642, 2015.

[40] H. Yin, B. Cui, L. Chen, Z. Hu, and C. Zhang, "Modeling location-based user rating profiles for personalized recommendation," *TKDD*, vol. 9, no. 3, pp. 1–41, 2015.