You Are How You Move: Linking Multiple User Identities From Massive Mobility Traces

Huandong Wang* Yong Li* Gang Wang† Depeng Jin*

Abstract

Understanding the linkability of online user identifiers (IDs) is critical to both service providers (for business intelligence) and individual users (for assessing privacy risks). Existing methods are designed to match IDs across *two services*, but face key challenges of matching multiple services in practice, particularly when users have multiple IDs per service. In this paper, we propose a novel system to link IDs across multiple services by exploring the spatial-temporal locality of user activities. The core idea is that the same user's online IDs are more likely to repeatedly appear at the same location. Specifically, we first utilize a *contact graph* to capture the "co-location" of all IDs across multiple services. Based on this graph, we propose a set-wise matching algorithm to discover candidate ID sets, and use Bayesian inference to generate confidence scores for candidate ranking, which is proved to be optimal. We evaluate our system using two real-world ground-truth datasets from an ISP (4 services, 815K IDs) and Twitter-Foursquare (2 services, 770 IDs). Extensive results show that our system significantly outperforms the state-of-the-art algorithms in accuracy (AUC is higher by 0.1-0.2), and it is highly robust against matching order and number of services.

1 Introduction

Online services are playing critical roles in almost all aspects of users' life. It is very common for a user to have multiple online identifiers (IDs) in different services such as online social networks (OSN), e-commerce services, online games, etc. Users may even have *multiple* IDs in *a single service*, where different IDs are used for different purposes [11].

Service providers have strong motivations to massively mining user data for monetization and optimizing user experience [7]. To capture a more comprehensive understanding of user behavior, it is increasingly intriguing to link user IDs across multiple services to fuse the separated data [17,20]. However, from the user perspective, linking IDs across services may have privacy implications since more information are exposed [4,10].

To these ends, understanding the "linkability" of online IDs is critical to both service providers (for business intelligence) and users (for understanding privacy risks). Early research has explored different ways to link user IDs by using service-specific data such as user profile attributes [2] and social graphs [6]. However, these approaches depend on whether these services have the same data type. For example, e-commerce services often do not have social graphs to match with an online social network. Moreover, users may fill in fake information (*e.g.*, name, gender) in their profiles, which makes the linkage even harder.

In this paper, we explore a more generic approach to link user IDs by leveraging the spatial-temporal locality of user activities. The key intuition is that no matter what online services a user accesses, we can bind them to the user's *physical presence*, which is characterized by time and location. This becomes possible because most online services today have a mobile version with locations as parts of the service (*e.g.*, Uber, Yelp, Twiter). In addition, with some tolerance on granularity, even network accessing related information can be translated into location [5]. Our goal is to link multiple online IDs that belong to the same users across different services. This requires solving three key challenges that have not be addressed in existing work [6,13,14]:

- *Service multiplicity*: existing methods mainly focus on linking IDs of two services [8,13]. In practice, however, the number of services can easily go over two. We find that adapting existing methods by matching services one by one produces unreliable results (see §5), which is significantly influenced by the number of services and the matching orders.
- *ID multiplicity*: users may register more than one ID in a service [11]. ID matching between two services should be "set-wise", *i.e.*, linking a set of IDs. However, existing methods are designed for pair-wise ID matching, which fail to capture users' multiple IDs in a single service [13,14].

*Tsinghua National Laboratory for Information Science and Technology. Department of Electronic Engineering, Tsinghua University, Beijing, China. liyong07@tsinghua.edu.cn.

†Department of Computer Science, Virginia Tech.

- *Heterogeneous data quality*: since user mobility behavior is extremely heterogeneous [8], the quantity and resolution of location data are drastically different across services. Early works simply filter out a large number of IDs with low-quality data [6, 13], which significantly reduces the data coverage and usability.

To solve these challenges, we propose a *contact graph* model for multi-service ID linking. Instead of matching IDs of just two services in a bipartite graph, we directly map multiple services and all their IDs into one big graph. In this graph, each node is an ID (regardless of service), and an edge represents that the connected IDs visited the same physical location, which is weighted by the number of co-locations of them. The high-level intuition is that the users' daily movements are fairly predictable with repeated patterns [16]. If multiple IDs belong to the same user, they are more likely to be "co-located" to build edges and form distinguishable subgraph structures over time.

Based on the contact graph, we propose a Bayesian-based optimal ID matching algorithm, which identifies the most probable ID sets that belong to the same physical user with the *target ID*. The high-level intuition is to extract possible candidate sets from neighboring IDs, and *rank* candidate sets based on their joint probability to match the target ID. Then, we propose a Bayesian inference method to obtain the joint probability for ranking candidate sets, which is proved to be optimal.

Our system allows a service provider to match its own IDs (as target IDs) with multiple other services simultaneously. The matching is based on "sets", capturing users who have multiple IDs in the same service. In addition, the contact graph includes all IDs without arbitrarily pre-filtering any data, and produces a confidence probability for the candidate sets. This allows applications to make use of the available data based on specific contexts.

We evaluate our system based on two real-world *ground-truth* datasets. One is from a large ISP that contains 412,455 users (815,117 online IDs) and 31 million access records to 4 online services: instant message (QQ), social network (Weibo), e-commerce (Taobao) and online review (Dianping). The second dataset contains 24K check-ins from Twitter and Foursquare from 385 users (770 online IDs) [21]. We use the state-of-the-art pair-wise ID matching algorithms POIS [13] and WYCI [14] as baselines. The results show that our algorithm significantly outperforms baselines (by 0.1 in AUC), particularly on users with multiple IDs per service (by 0.2 in AUC). We have three novel contributions summarized as follows:

- First, we propose a generic and optimal ID linking algorithm utilizing the spatial-temporal locality of user activities. Our *contact graph* model achieves set-wise ID matching for multiple services. The model effectively captures users with multiple IDs per service, and mitigates the ordering effect of multi-service matching.
- Second, we propose a novel Bayesian-based method to produce confidence probability for ID matching with proof of optimality. This addresses the challenge of uneven data-quality across services: instead of arbitrarily pre-filtering low-quality data, our method keeps all IDs in the matching for maximum data utilization.
- Third, we evaluate our system based on two real-world ground-truth datasets. The results show that our system significantly outperforms the start-of-the-art in accuracy, and it is robust against number of services and matching order.

2 Related Work

Applications of ID Linking. A number of applications can benefit from linking IDs across services. For example, Zafarani et al. [20] and Yan et al. [17] leveraged linked IDs across social networks for better friend recommendations. Kumar et al. [7] investigated the user migration patterns across social media sites to provide guidance for online social network design. Yang et al. [18] leveraged linked IDs across sites for better video recommendations. All these works indicate the strong motivations for service provider to link IDs belonging to the same user.

ID Linking Methods. Most existing ID linking methods utilize either different portions of the *same* dataset [2, 6], or observe the same behavior across thematically similar domains [1, 10, 19]. For example, Korula et al. [6] linked user IDs using social graphs. Goga et al. [2] linked IDs based on user profile attributes such as user names, profile photos. Zafarani et al. [19] linked IDs based on user names through behavioral modeling. Narayanan et al. [10] linked users of Netflix and IMDB based on the similarity of their movies ratings. Mu et al. [9] used "latent user space" for linking user profiles. Goga et al. [1] described a set of similarity features for ID linking such as timestamp of posts and writing styles. All these approaches rely on service-specific features (*e.g.,* social graph), which are depended on whether two services have overlapped features. In this work, we explore the spatial-temporal locality of user activities for ID linking, which utilizes more generic information from services.

Linking ID using Location Data. A few recent works examine the possibility of linking IDs

Table 1: List of commonly used notations.

Notat.	Description
$\boldsymbol{A}$	The the set of all online IDs.
$\boldsymbol{S}$	The set of types (services) for online IDs.
$\boldsymbol{L}, \boldsymbol{T}$	The set of all locations and time bins.
$\boldsymbol{A}^s$	The set of all online IDs of type $s \in \boldsymbol{S}$.
$\boldsymbol{G}$	Contact graph of online IDs in physical world.
$\mathcal{N}(v)$	The neighbor of $v \in \boldsymbol{A}$.
$r(\xi)$	The login records for a set of online IDs $\xi \subseteq \boldsymbol{A}$.
$X(\xi, v)$	Binary variable indicating whether all IDs in $\xi \subseteq \boldsymbol{A}$ and $v \in \boldsymbol{A}$ belong to the same users.
$s(v)$	The service type of online ID $v \in \boldsymbol{A}$.
$\mathcal{P}(\boldsymbol{A})$	The set of all partitions of $\boldsymbol{A}$.
$\mathcal{P}(\boldsymbol{A}, \xi)$	The set of all partitions in which all IDs in ξ are divided into one set.

based on location data [13, 14]. Riederer et al. [13] linked two trajectory datasets with maximum weight matching. Rossi et al. [14] proposed a trajectory-based linking method based on their defined spatio-temporal distance. These approaches can only perform pair-wise ID matching between two services, and face key challenges in the multiplicity of IDs and services. These two algorithms will be used as our evaluation baselines.

3 Problem Formulation

In this section, we first propose a mathematical model and formally define the set-wise ID matching problem. Then, we introduce two important concepts, *i.e.*, *contact graph* and *partition*, and present a probability model of users' behavior to solve the set-wise ID matching problem. For readability, we summarize the major notations used throughout the paper in Table 1.

3.1 Mathematical Model. Let $\boldsymbol{A}$ represent the set of online IDs, and $\boldsymbol{S}$ represent the set of types of online IDs. For each ID $v \in \boldsymbol{A}$, we denote $s(v)$ as its type. $\forall s \in \boldsymbol{S}$, we define $\boldsymbol{A}^s$ as the set of all IDs of type s.

Given any online ID $u \in \boldsymbol{A}$, we define its mobility records as $r(u) = \{(l_1, t_1), (l_2, t_2), ...\}$, where (l_i, t_i) represents a login record in location l_i at time t_i. Specifically, locations and times are divided into bins, corresponding to geographical regions and intervals of time, respectively. We further define $\boldsymbol{T}$ as the set of all time bins, and $\boldsymbol{L}$ as the set of all regions.

For a cluster of online IDs ξ, we define their mobility records as $r(\xi) = \{r(w) | w \in \xi\}$. Then, for each pair of online IDs $u, v \in \boldsymbol{A}$, let binary variable $X(u, v)$ indicate whether they belong to the same user. That is,

$$X(u, v) = \begin{cases} 1, & \text{if } u, v \text{ belong to the same user,} \\ 0, & \text{otherwise.} \end{cases}$$

More generally, for a set of online IDs $\xi \subseteq \boldsymbol{A}$, we use $X(\xi)$ to indicate whether they belong to the same users. Thus, we have $X(\xi) = \prod_{u,v \in \xi} X(u, v)$. Similarly, for an online IDs v and a set of IDs ξ, we let $X(\xi, v)$ indicate whether they belong to the same users, which can be expressed as $X(\xi, v) = \prod_{u \in \xi} X(u, v)$.

Our goal is that for an arbitrary target ID $v \in \boldsymbol{A}$, finding online IDs belonging to the same user. Based on these notations, we formally define it as the follows:

Set-wise Identity Matching Problem (SIMP)

Given: The target ID v, a list of candidate sets of IDs $\xi_1, ..., \xi_N \subseteq \boldsymbol{A}$, and their mobility records $r(v)$ and $r(\xi_i)$ for $i = 1, ..., N$.

Problem: Find a ranking function $\phi : \{\xi_1, ..., \xi_N\} \rightarrow \{1, ..., N\}$, such that the IDs belonging to the same user with v are ranked as high as possible, which can be expressed as:

$$\min_{\phi} \sum_{i=1}^{N} X(\xi_i, v)\phi(\xi_i). \tag{3.1}$$

3.2 Contact Graph and Partition. It has been found that users' daily mobility is fairly predictable with repeated patterns [16]. If multiple IDs belong to the same user, they are more likely to be "co-located". Thus, we can define the *contact graph* of IDs to extract the subgraph structures formed by IDs belonging to the same user.

Definition 1 (Contact Graph) Contact graph of IDs is defined as a graph $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{E})$. For a pair of online IDs u, $v \in \boldsymbol{A}$, we say there exists an edge between u and v in $\boldsymbol{E}$, if u and v have mobility records at the same locations, *i.e.*, $\exists l \in \boldsymbol{L}$, such that $(l, t_1) \in r(u)$ and $(l, t_2) \in r(v)$ hold for some t_1, $t_2 \in \boldsymbol{T}$.

Note that the *contact graph* is defined only based on spatial information of mobility records of IDs, regardless of the temporal information. The intuition is to capture the candidate IDs belonging to the same user through the connectivity of the *contact graph* as much as possible. Actually, if two nodes have no common location in their trajectories, there is very low probability for them to belong to the same user. Thus, for each ID v, we can limit the candidate online IDs belonging to the same user with it as a subset of its neighbor $\mathcal{N}(v) = \{b | b \in \boldsymbol{A}, (b, v) \in \boldsymbol{E}\}$. We further define the neighbor of v with a certain type $s \in \boldsymbol{S}$ as $\mathcal{N}^s(v) = \mathcal{N}(v) \cap A^s$. As for the temporal information, we will model it in the weight of edge. Specifically, for an edge $(u, v) \in \boldsymbol{E}$, we define its weight $w(u, v)$ as the probability that they belong to the same user, *i.e.*, $w(u, v) = P(X(u, v) = 1)$. For example, nodes with more frequent "co-locations" will have larger weight between them, which be introduced in detail in Section 4.

On the other hand, in order to describe the subgraph structures of IDs belonging to different users simultaneously, we define "partition" of IDs as follows:

Definition 2 (Partition) Given a node set $\boldsymbol{V}$, $p = \{\xi_1, \xi_2, ..., \xi_n\}$ is a set of nonempty subset of $\boldsymbol{V}$, *i.e.*, $\forall k \in \{1, ..., n\}$, $\xi_k \subseteq \boldsymbol{V}$. Then, p is a partition of $\boldsymbol{V}$ if the following three conditions hold: **(1)** $\emptyset \notin p$, **(2)** $\cup_{\xi\in p}\xi = \boldsymbol{A}$, **(3)** if ξ_1, $\xi_2 \in p$ and $\xi_1 \neq \xi_2$, we have $\xi_1 \cap \xi_2 = \emptyset$.

There is an inherent partition of $\boldsymbol{A}$ composed of the true set of online IDs belonging to each user. Specifically, we assume, the number of each user's IDs of type $s \in \boldsymbol{S}$ follows independent geometric distribution with parameter θ_s. That is, the prior probability of a set of IDs ξ belonging to the same user is $P(\xi) = \prod_{s\in\boldsymbol{S}} \theta_s^{n_s(\xi)}(1-\theta_s)$, where $n_s(\xi)$ is the number of IDs of service s in ξ. Then, for a partition p, its prior $P(p)$ depends on the online IDs of each user, which can be expressed as follows:

$$P(p) = \prod_{\xi\in p} P(\xi). \tag{3.2}$$

In addition, we use $\mathcal{P}(\boldsymbol{V})$ to denote the set of all partitions of the set $\boldsymbol{V}$. Then, for a subset $\xi \subseteq \boldsymbol{V}$, we define $\mathcal{P}(\boldsymbol{V},\xi)$ as the set of all partitions in which all IDs in ξ are divided into one set, *i.e.*, $\mathcal{P}(\boldsymbol{V},\xi) = \{p|p \in \mathcal{P}(\boldsymbol{V}),\ \exists\zeta \in p, s.t.\ \xi \subseteq \zeta\}$.

3.3 Probability Model of User Behavior.

In order to formally analyze the problem, a probability model is needed to describe how users appear in different locations to generate records. Here, we do not distinguish individuals, and aim to find a general model for all users.

In order to generally model the various user behavior of generating location records in accessing different services, we binarize the number of records in discrete time bin, *i.e.*, modeling whether users have generated an observation (mobility record) in each time bin [13]. Based on this simplification, users' visitation in discrete time intervals can be modeled by Bernoulli distribution related to the visited location, *i.e.*, visitation to the location l at each time bin follows Bernoulli distribution with probability of p_l. Then, under the condition that the user has visited the location l, whether a record of an ID of type s exists follows Bernoulli distribution with probability of p_s. Specifically, for an ID $v \subseteq \boldsymbol{A}$, its observed records are generated with the probability of:

$$P(r(v)) = \prod_{l\in\boldsymbol{L}}\prod_{t\in\boldsymbol{T}}\left[(1-p_l)^{1-I_v(l,t)} + p_l p_{s(v)}^{I_v(l,t)}(1-p_{s(v)})^{1-I_v(l,t)}\right],$$

where $I_v(l,t)$ is the indicator function of whether the login record (l,t) exists in $r(v)$.

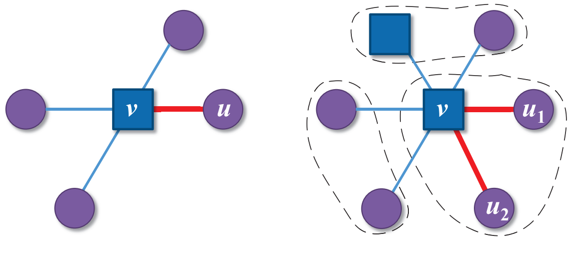
Parameters p_l and p_s can be estimated from the perspective of expectations. Let N_l^s represent the total number of records of service s in location $l \in \boldsymbol{L}$. Then, from a global viewpoint, the expected number of records of service s accessed by all users can be expressed as follow,

$$|\boldsymbol{A}^s|\cdot p_s\cdot|\boldsymbol{T}| = \sum_{l\in\boldsymbol{L}} N_l^s. \tag{3.3}$$

On the other hand, for each location l, the expected number of records at location l can be expressed as follows:

$$|\boldsymbol{A}^s|\cdot p_l\cdot p_s\cdot|\boldsymbol{T}| = N_l^s. \tag{3.4}$$

By combining (3.3) and (3.4), we achieve the estimation for p_l and p_s.



(a) v and $u \in \mathcal{N}^{s_1}(v)$. (b) A partition for $\mathcal{N}(v)$.

Figure 1: Diagrams for relationship between candidate nodes or sets in $\mathcal{N}(v)$ and the target node v.

On the other hand, for a set of online IDs $\xi \subseteq \boldsymbol{A}$, under the condition that they belong to the same user, their observed records are generated with the probability:

$$P(r(\xi)|X(\xi)=1)$$
$$= \prod_{l\in\boldsymbol{L}}\prod_{t\in\boldsymbol{T}}\left[(1-p_l)^{1-I_\xi(l,t)} + p_l\prod_{w\in\xi} p_{s(w)}^{I_w(l,t)}(1-p_{s(w)})^{1-I_w(l,t)}\right],$$

where $I_\xi(l,t)$ is the indicator function of whether the login record (l,t) exists in $r(\xi)$.

4 Method

The goal of the set-wise ID matching problem is to find a ranking function $\phi(\xi_k)$ for each candidate ID set ξ_k. Specifically, we rank candidate ID sets ξ_k based on the joint posterior probability of IDs in ξ_k belonging to the same user conditioned on the observed mobility records, *i.e.*, $P(X(\xi_k,v)=1|r(\boldsymbol{V}))$, where $\boldsymbol{V}$ is the set of candidate online IDs. As introduced above, we set $\boldsymbol{V}$ to be $\mathcal{N}(v)$. We now introduce how to calculate $P(X(\xi,v)=1|r(\boldsymbol{V}))$.

4.1 Pair-wise Matching Problem.

Let us first consider the case of pair-wise matching, *i.e.*, linking the pairs of IDs of two services that belong to the same users. In this case, each user is assumed to have at most one ID of each service. We denote the service of target ID as s_0, and the other service as s_1. Then, $\boldsymbol{V}$ can be further limited to $\boldsymbol{V} = \mathcal{N}^{s_1}(v) \cup v$, of which an example is shown in Figure 1(a).

For a target ID $v \in \boldsymbol{A}^{s_0}$ and an arbitrary ID $u \in \mathcal{N}^{s_1}(v)$, v can only belong to the same user with at most one ID in $\mathcal{N}^{s_1}(v)$, *i.e.*, $\sum_{w\in\mathcal{N}^{s_1}(v)} X(w,v) \leq 1$. Considering it is possible that v does not belong to the owners of any IDs in $\mathcal{N}^{s_1}(v)$, we have:

$$\sum_{w\in\mathcal{N}^{s_1}(v)} P(X(w,v)=1|r(\boldsymbol{V})) + \beta(v) = 1, \tag{4.5}$$

where $\beta(v)$ is the probability that v does not belong to the same user with any IDs in $\mathcal{N}^{s_1}(v)$. On the other hand, through Bayes' theorem, we have:

(4.6)
$$P(X(w,v)\!=\!1|r(\boldsymbol{V})) = \frac{P(X(w,v)\!=\!1)P(r(\boldsymbol{V})|X(w,v)\!=\!1)}{P(r(\boldsymbol{V}))}.$$

According to (3.2), for any $w \in \mathcal{N}^{s_1}(v)$, $X(w,v) = 1$ corresponds to a partition of $\boldsymbol{V}$ with the same prior probability, due to their similar component structure, *i.e.*, one 2-size set $\{w, v\}$ and other $|\boldsymbol{V}| - 2$ 1-size sets. We denote their prior probability as $P(X(w,v) = 1) = P(p_1)$. Further, we define $Q(w,v)$ as the joint probability of the observation $r(\boldsymbol{V})$ and $X(w,v) = 1$, which can be expressed as follows:

$$\begin{aligned} Q(w,v) &\triangleq P(X(w,v) = 1) \cdot P(r(\boldsymbol{V})|X(w,v) = 1), \\ &= P(p_1) \cdot P(r(w,v)|X(w,v) = 1) \prod_{o \in \boldsymbol{V} \setminus \{w,v\}} P(r(o)). \end{aligned}$$

Then, (4.6) can be simplified as follows:

$$P(X(w,v) = 1|r(\boldsymbol{V})) = \frac{Q(w,v)}{P(r(V))}. \tag{4.7}$$

Similarly, for $\beta(v)$, its corresponding partition is $p_0 = \{\{w\}|w \in \boldsymbol{V}\}$. Thus, we have:

$$\beta(v) = \frac{P(p_0) \cdot \prod_{w \in \boldsymbol{V}} P(r(w))}{r(\boldsymbol{V})}.$$

By defining $\beta'(v)$ as the numerator of $\beta(v)$ and combining (4.5) and (4.7), we have:

$$P(X(u,v) = 1|r(\boldsymbol{V})) = \frac{Q(u,v)}{\sum\limits_{w \in \mathcal{N}^{s_1}(v)} Q(w,v) + \beta'(v)},$$

So far, we have obtained the probability that u belongs to the same user with v, which solves the pair-wise matching problem.

4.2 Multiplicity of IDs and Services. Based on the pair-wise matching problem, we further investigate the problem of multiple IDs and services. For each online ID $u \in \mathcal{N}^{s_1}(v)$, we have obtained the probability that u belongs to the same user with v. However, in the general SIMP, multiple IDs in $\mathcal{N}^{s_1}(v)$ can belong to the same user with v simultaneously. IDs in $\mathcal{N}^{s_0}(v)$ can also belong to the same user with v. In addition, IDs of multiple services should also be considered. Thus, we solve the problem in $\boldsymbol{V} = \mathcal{N}(v)$, of which an example is shown in Figure 1(b).

When considering multiple online IDs, *e.g.*, $u_1, u_2 \in \mathcal{N}(v)$, random variables $X(u_1, v)$ and $X(u_2, v)$ are not independent with each other. Thus, we cannot obtain the joint probability by simply using the product of the probability for each ID. For a set of IDs $\xi \subseteq \boldsymbol{V}$, in order to calculate $P(X(\xi, v) = 1|r(\boldsymbol{V}))$, we first apply condition probability formula, which obtains:

$$\begin{aligned} P(X(\xi,v) = 1|r(\boldsymbol{V})) &= P(r(\boldsymbol{V}), X(\xi,v) = 1)/P(r(\boldsymbol{V})), \\ &\propto P(r(\boldsymbol{V}), X(\xi,v) = 1). \end{aligned}$$

Then, we utilize *partition* to further simplify this equation. Specifically, by applying the formula of total probability with respect to all possible partitions of $\boldsymbol{V}$, we have:

$$P(r(\boldsymbol{V}), X(\xi,v) = 1) = \sum_{p \in \mathcal{P}(\boldsymbol{V})} P(r(\boldsymbol{V}), X(\xi,v) = 1|p)P(p),$$

where $P(p)$ is the prior probability of the partition p. Specifically, for an arbitrary partition p, if ξ and v are divided into one set in p, we have $P(X(\xi,v) = 1|p) = 1$. Otherwise, it equals to 0. We use $\mathcal{P}(\boldsymbol{A}, \xi \cup v)$ to denote the set of all partitions in which all IDs in $\xi \cup v$ are divided into one set. Then, the right hand can be limited to $\mathcal{P}(\boldsymbol{A}, \xi \cup v)$. Combining relation of $P(r(\boldsymbol{V}), X(\xi,v)\!=\!1)$ and $P(X(\xi,v)\!=\!1|r(\boldsymbol{V}))$ based on Bayes' theorem, we have:

$$P(X(\xi,v) = 1|r(\boldsymbol{V})) \propto \sum_{p \in \mathcal{P}(\boldsymbol{V}, \xi \cup v)} P(r(\boldsymbol{V})|p)P(p). \tag{4.8}$$

In addition, for an arbitrary partition $p \in \mathcal{P}(\boldsymbol{V})$, we use $D(p)$ to represent the likelihood $P(r(\boldsymbol{V})|p)P(p)$, which can be calculated as follows,

$$D(p) = P(r(\boldsymbol{V})|p)P(p) = P(p) \prod_{\eta \in p} P(r(\eta)|X(\eta) = 1).$$

Putting it into (4.8), we obtain:

$$P(X(\xi,v) = 1|r(\boldsymbol{V})) \propto \sum_{p \in \mathcal{P}(\boldsymbol{V}, \xi \cup v)} D(p). \tag{4.9}$$

On the contrary, $X(\xi,v) \neq 1$ corresponds to partitions in which all IDs in $\xi \cup v$ are not divided into one set. Thus, we also have:

$$P(X(\xi,v) \neq 1|r(\boldsymbol{V})) \propto \sum_{p \in \mathcal{P}(\boldsymbol{V}) \setminus \mathcal{P}(\boldsymbol{V}, \xi \cup v)} D(p). \tag{4.10}$$

By combining (4.9) and (4.10), we have:

$$P(X(\xi,v) = 1|r(\boldsymbol{V})) = \sum_{p \in \mathcal{P}(\boldsymbol{V}, \xi \cup v)} D(p) / \sum_{p \in \mathcal{P}(\boldsymbol{V})} D(p). \tag{4.11}$$

So far, we have obtained the probability that all IDs in ξ belongs to the same user with v, which solves the set-wise matching problem. This ranking function based on the posterior probability is optimal, which is proved by the following theorem:

Theorem 1 The ranking function based on the posterior probability $P(X(\xi,v) = 1|r(\boldsymbol{V}))$ is the optimal solution of *Set-wise Identity Matching Problem.*

Proof: Let ϕ_0 denote the ranking function based on the posterior probability. Since the target function (3.1) is a random variable, we consider its expectation conditioned on the observations, and have:

$$E(\sum_{i=1}^{N} X(\xi_i, v)\phi(\xi_i)|r(\boldsymbol{V})) = \sum_{i=1}^{N} P(X(\xi_i, v) = 1|r(\boldsymbol{V}))\phi(\xi_i).$$

Without loss of generality, we assume $P(X(\xi_1, v) = 1|r(\boldsymbol{V})) \geq P(X(\xi_2, v) = 1|r(\boldsymbol{V})) \geq ... \geq P(X(\xi_N, v) =$

$1|r(\boldsymbol{V}))$. On the other hand, according to SIMP algorithm, we have $\phi_0(\xi_i) = i$. Thus, we have $\phi_0(\xi_1) \leq \phi_0(\xi_2) \leq ... \leq \phi_0(\xi_N)$. Combining these two inequalities and applying the rearrangement inequality, for an arbitrary ranking function ϕ_1, we have:

$$\sum_{i=1}^{N} P(X(\xi_i, v){=}1|r(\boldsymbol{V}))\phi_1(\xi_i) \geq \sum_{i=1}^{N} P(X(\xi_i, v){=}1|r(\boldsymbol{V}))\phi_0(\xi_i).$$

In another form, it can be expressed as:

$$E(\sum_{i=1}^{N} X(\xi_i, v)\phi_1(\xi_i)|r(\boldsymbol{V})) \geq E(\sum_{i=1}^{N} X(\xi_i, v)\phi_0(\xi_i)|r(\boldsymbol{V})).$$

Thus, the ranking function ϕ_0 generated by the algorithm minimizes the target function $\sum_{i=1}^{N} X(\xi_i, v)\phi(\xi_i)$ of the set-wise identity matching problem, which proves its optimality.■

4.3 Approximation Algorithm. The ranking function based on posterior probability is optimal. However, calculating it based on (4.11) suffers from high computational complexity growing exponentially with the size of $\boldsymbol{V}$. In order to solve this problem, we apply the three approximation methods as follows:

- *Ignoring non-adjacent IDs:* It is unreasonable to link two IDs of which the trajectories have no co-location. Thus, we limits the problem to the neighbor of v, *i.e.*, $\mathcal{N}(v)$, which significantly reduces the size of $\boldsymbol{V}$. For these IDs with a large number of neighbors, we implement further approximation methods to them.
- *Ignoring the denominator:* From (4.11), we can observe that the denominator is independent with ξ. Thus, we alternatively rank the candidate ID sets only based on the numerator of (4.11). By this way, only partitions in $\mathcal{P}(\boldsymbol{V}, \xi \cup v)$ need to be considered.
- *Reducing feasible partitions:* We further reduce feasible partitions in $\mathcal{P}(\boldsymbol{V}, \xi \cup v)$ to reduce computational complexity. Specifically, we use the constant parameter N_{max} to represent the maximum accepted $|\boldsymbol{V}|$. If $|\boldsymbol{V}| \geq N_{max}$, all IDs in $\boldsymbol{V} \backslash \{v \cup \xi\}$ are considered to belong to different users.

Algorithm 1 RS(v, $\mathcal{N}(v)$, ξ)

```
Input: The target ID $v$, its neighborhood $\mathcal{N}(v)$ and a set of IDs $\xi \subseteq \mathcal{N}(v)$.
Output: The probability-based ranking score $\text{RS}(v, \mathcal{N}(v), \xi) = P(X(\xi, v){=}1|r(\boldsymbol{V}))$.
Initialize: $D_{sum} \leftarrow 0$; $D_{target} \leftarrow 0$; $f \leftarrow 0$;
if $|\mathcal{N}(v)| > N_{max}$ then
    $f = 1$;
if $f = 0$ then
    for $p \in \mathcal{P}(\mathcal{N}(v))$ do
        $D_{sum} = D_{sum} + D(p)$;
        if $\exists U \in p$ s.t. $\xi \cup v \subseteq U$ then
            $D_{target} = D_{target} + D(p)$;
    $\text{CP}(\xi, v) = D_{target}/D_{sum}$.
else
    $p = \{\xi \cup v\} \bigcup \{\{w\}|w \in \boldsymbol{V} \backslash \{v \cup \xi\}\}$;
    $\text{CP}(\xi, v) = D(p)$.
```

Based on above approximation methods, the computational complexity is reduced from $O(2^{|\boldsymbol{V}|})$ to $O(|\boldsymbol{V}|)$. Based on them and (4.11), we propose an algorithm to calculate the confidence score, which is described in Algorithm 1. Given the target ID v and its neighborhood $\mathcal{N}(v)$, if $|\mathcal{N}(v)| \leq N_{max}$, the confidence score is computed by traversing all partitions in $\mathcal{P}(\boldsymbol{V})$ according to (4.11). Otherwise, the two proposed approximation methods are adopted to reduce the computational complexity. Based on Algorithm 1, we obtain the ranking function ϕ of ID sets, where ID sets with higher confidence probability are given higher rankings.

5 Performance Evaluation

We evaluate our system against two state-of-the-art pair-wise matching algorithms using two ground-truth datasets. Now we introduce the utilized datasets, evaluation metrics and experiment results.

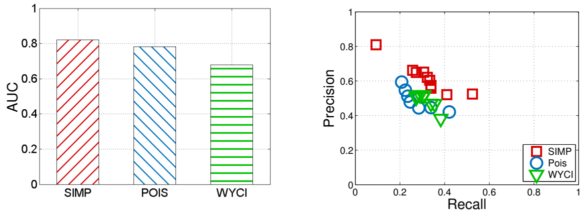
5.1 Datasets. We have two *ground-truth* datasets for performance evaluation, including a new ISP dataset (4 services) and an existing dataset from a prior work (2 services).

ISP Dataset: This dataset is collected by a large ISP in China, which covers 412,455 ground-truth users and their 815,117 IDs in 4 representative online services including instant messenger (QQ), online social network (Weibo), e-commerce (Taobao) and online review site (Dianping) during the full month of November 2015. Details are shown in Table 2. It records users' accessing activities via broadband network, which are associated to a physical locations, *e.g.*, a WiFi access point or a broadband interface. For simplicity, we refer them as access points (AP). Each record represents the user's login action in a given service, characterized by "service name", "ID", "AP name", and "timestamp". There are 31 million total records (on average 38.2 records per ID).

The ground-truth is also provided by the same ISP, which collect users' online IDs via the cellular networks that are associated with the same device with unique cellular identifier. Note that all the IDs and cellular identifiers have been fully anonymized (hashed

Table 2: Four services in the ISP dataset.

Service	Type	# of IDs
QQ	Instant messaging (IM)	725,621
Taobao	E-commerce (EC)	7,545
Weibo	Online social network (OSN)	2,545
Dianping	Online review (OR)	79,403



(a) AUC (b) Precision and Recall

Figure 2: Performance on IDs with one-to-one relation (IM vs. OR).

bit string) without any user PII or meta data. The real IDs were never made available to or utilized by us. In addition, the usage of anonymized datasets is approved by the ISP and our institution.

Twitter-Foursquare: On Foursquare, users may display their Twitter account information, which makes it possible to obtain the ground-truth mapping between Twitter IDs and Foursquare IDs. This dataset is collected by Zhang et al. [21]. In total, it contains 385 users with location check-ins on both sides (770 online IDs), and totally 24,556 location check-ins collected from both Twitter and Foursquare.

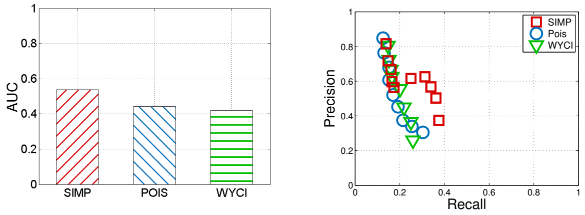
5.2 Baseline Algorithms. We compare our algorithm with two state-of-the-art algorithms as follows:

POIS: Riederer et al. [13] assume the number of visits of each user to a location follows a Poisson distribution, and an action (*e.g.* login) on each service occurs independently with Bernoulli distribution. Based on the ranking score derived from this model, it finds the maximum weighted matching of IDs as the results. In addition, it filters out IDs by the “eccentricity” factor ϵ, which is defined as the threshold for the weight gap between the best and second-best IDs.

WYCI: Rossi et al. [14] use the frequency of user login activities in different locations to approximate the probability of location visiting by $P(l|r(v)) = \frac{N_l^v+\alpha}{\sum_{l\in L} N_l^v+\alpha|\boldsymbol{L}|}$, where N_l^v is the number of login records of v at location l. $\alpha > 0$ is the smoothing parameter and $|\boldsymbol{L}|$ is the number of locations. Then, it iteratively finds ID u to maximize the probability $\prod_{(l,t)\in r(u)} P(l|r(v))$.

Both algorithms are designed for ID matching between two services, and we apply them by matching multiple services one by one. For a given ID in one service, they produce a ranked list of the matched IDs (with ranking scores).

5.3 Evaluation Metrics. We use standard metrics including precision, recall and AUC to evaluate the algorithm performance with some adjustments. More specifically, for a target ID v, POIS or WYCI produces *a ranked list of IDs*: $[u_1, u_2, ..., u_k]$, where u_i is the i_{th} ID and k is the number of matched IDs. Our system SIMP produces *a ranked list of sets*: $[U_1, U_2, ..., U_k]$, where each item U_i is a set of IDs ($U_i = \{u_{i1}, ..., u_{is}\}$). For comparison, we need to either convert a set-list to an ID-list, or the other way around.



(a) AUC (b) Precision and Recall

Figure 3: Performance on IDs with one-to-one relation (Twitter vs. Foursquare).

ID List Evaluation: First, we convert the set-list generated by our algorithm to an ID-list, by setting the set size as 1 ($|U_i| = 1$). That means for each set in the list $[U_1, U_2, ..., U_k]$, we choose the top one to form a new ID list, and then compute the metrics using standard precision, recall, and AUC.

Precision & Recall: Given the list of matched IDs, precision is the fraction of IDs that truly belong to the same user v. Recall is the fraction of v's actual IDs that are correctly matched in this list [12].

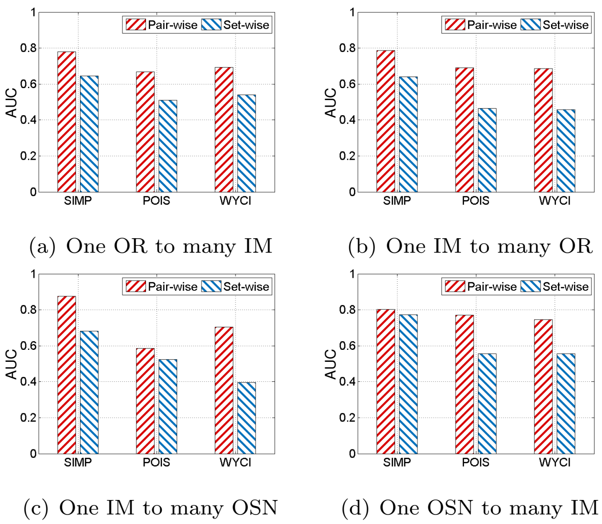
AUC pair-wise: AUC refers to “Area under ROC curve” [3, 15] where ROC curve is plotting the true positive rate (TPR) against the false positive rate (FPR). AUC is essentially evaluating the quality of a ranking, and its value equals to the probability of ranking a randomly chosen positive instance higher than a randomly chosen negative one,

$$AUC = \frac{\sum_{i=1}^{n_0}(n_0 + n_1 - r_i) - n_0(n_0+1)/2}{n_0 n_1},$$

where r_i is the rank of i_{th} positive instance, and n_0 and n_1 are the number of positive and negative instances, respectively. Here, “positive” means the matched ID is correct based on ground-truth. We set $k = 10$ IDs in the list, and thus $n_0 + n_1 = k = 10$. We refer this AUC (for pair-wise matching algorithms) as *AUC pair-wise*.

Set List Evaluation: Clearly, converting the set-list to an ID-list diminishes the key benefits of our system. Our system may correctly match all the IDs in the top candidate set, but had to shrink the set size to 1 for the comparison. Thus, we introduce a method to convert the ID list into set list (applied to results of POIS and WYCI), and use a AUC to evaluate the ranking quality of the set list. The basic idea is to group IDs into sets (of pre-defined size), and then we rank these sets based on the highest ranked ID in each set. For example, for a given ID list $[u_1, u_2, , ..., u_3]$, we can convert them to a set list (with set size = 2) as $[\{u_1, u_2\}, \{u_1, u_3\}, \{u_2, u_3\}]$.

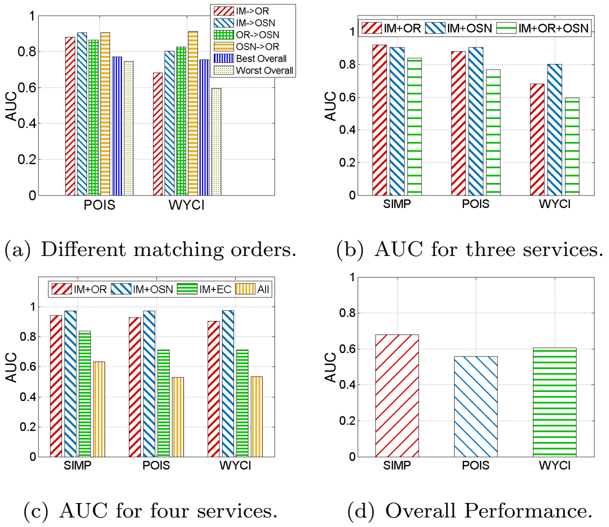
AUC set-wise: For a given list of sets, we also use AUC to evaluate the ranking quality. We follow the same definition of AUC to calculate the probability of ranking a randomly chosen “positive set” higher than



(a) One OR to many IM (b) One IM to many OR

(c) One IM to many OSN (d) One OSN to many IM

Figure 4: Performance on IDs with one-to-many relation, with different combinations of services.



(a) Different matching orders. (b) AUC for three services.

(c) AUC for four services. (d) Overall Performance.

Figure 5: Performance on IDs of multiple services and overall performance.

a randomly chosen "negative set", where "positive set" means that *all* the IDs in this set belong to the same user as the target ID v. If any ID is incorrect, the set is a negative one.

5.4 Experiment and Results. We evaluate our system by experiments in different cases with IDs of one-to-one relation (one ID per service) and of one-to-many relation (multiple IDs per service and multiple services).

One-to-One Relation: We first select users who only have one IM identity and one OR identity as the ground-truth, and evaluate the performance of different algorithms for IDs with one-to-one relation. The results are shown in Figure 2. Specifically, Figure 2(a) shows our algorithm (SIMP) has a higher AUC than two state-of-the-art algorithms. Figure 2(b) shows the precision-recall trade-off by adjusting parameters (*e.g.*, θ_s for SIMP, ϵ in POIS). Our algorithm achieves better performance.

In Figure 3, we conduct the same experiments using the Twitter-Foursquare dataset, whose quality is found to be worse than the ISP data — check-in data is much more sparse. The overall AUC becomes lower than that of Figure 2, but the trend is still consistent: our algorithm outperforms the baselines by 0.1 in AUC. The largest precision gain is about 0.32. Note that this is comparing the ID-list metrics. In this case, the set size shrinks to 1, which is to our disadvantage.

One-to-Many Relation: Then, we select users with IDs of one-to-many relation, *e.g.*, users with one IM identity and multiple OR identities, and evaluate the performance of our system using both pair-wise and set-wise AUC. The results are shown in Figure 4. From the results, we can have three key observations. First, our algorithm significantly outperforms the baselines over different combinations of services and AUC metrics. Second, the performance of our algorithm is more *consistent* over different services. The baselines, in contrary, have a larger variance in the AUC. Third, the gaps between our algorithm and baselines are larger for set-wise AUC than pair-wise AUC. This indicates the advantage of our algorithm in finding multiple IDs in one service simultaneously. The highest gain over baselines is 0.2 in terms of AUC (pair-wise), when matching IM to OSN.

Multiple Services: Next, we examine the performance of our algorithm in linking IDs over multiple services by selecting users who have three IDs (*i.e.*, one IM, one OR and one OSN) as the ground-truth. We examine the impact of *number of services* and *matching order*. In this particular experiment, we use IM identity as the target ID, and find the other two IDs belonging to the same user. There are three possible matching sequences: (IM-OR, IM-OSN), (IM-OR, OR-OSN), (IM-OSN, OSN-OR). We perform each sequence and obtain the set-wise AUC shown in Figure 5.

From Figure 5(a) and (b), we find that *number of services* has a significant influence on the two baseline algorithms. Both POIS and WYCI have a clear performance degradation from 2-service matching to 3-service matching, where the average AUC difference is 0.16. In addition, the *matching order* also matters, particularly for WYCI. In this case, the AUC difference between the best and worse sequences can be as large as 0.2. This confirms our design intuition, that the pair-wise matching has fundamental limitation to scale-up to multiple services. On the other hand, as we can observe from Figure 5(b), performance degradation of our proposed algorithm is only 0.05 in terms of AUC from 2-service matching to 3-service matching, while it is 0.16 on average for two baseline algorithms. It indicates that our algorithm is much less sensitive to the matching order nor the number of services.

Figure 5(c) shows the result by extending the experiment scope to users who have four IDs (*i.e.*, one ID for each service). We can observe that our algorithm consistently outperforms baselines under these settings. The advantage is more obvious for multi-service matching. Finally, we discard all constraints and evaluate the performance of different algorithms. The results are shown in Figure 5(d). From the results, we can observe that our algorithm outperforms other algorithms with performance gap of over 0.1 in terms of AUC.

Summary. The evaluation results show that our proposed system outperforms the state-of-the-art algorithms in different aspects, particularly for IDs of many-to-many relation across multiple services. Its AUC beats baselines by 0.1 in overall performance, and by 0.2 in many-to-many ID matching, demonstrating the effectiveness of our proposed system.

6 Conclusions

In this work, we propose an ID-linking algorithm across multiple services by modeling the spatial-temporal locality of user activities. We propose a novel contact graph and an optimal Bayesian-based inference method to link IDs across services. Our system solves a number of open problems in multi-service ID linking, including service and identity multiplicity and heterogeneous data quality. Extensive experiments on large scale and real-world datasets demonstrate the effectiveness of our system. In addition, we will release parts of the dataset and all the code, which is available at our github repository*. We believe it paves the way toward solving ID linkage problems in practice.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China under 61621091 and 61673237, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology, and the NSF grant CNS-1717028.

References

[1] O. Goga, H. Lei, S. H. K. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira, *Exploiting innocuous activity for correlating users across sites*, in Proc. WWW, 2013.

[2] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. P. Gummadi, *On the reliability of profile matching across large online social networks*, in Proc. KDD, 2015.

[3] D. J. Hand and R. J. Till, *A simple generalisation of the area under the roc curve for multiple class classification problems*, Machine learning, 45 (2001), pp. 171–186.

[4] D. Irani, S. Webb, C. Pu, and K. Li, *Modeling unintended personal-information leakage from multiple online social networks*, Internet Computing, 15 (2011), pp. 13–19.

[5] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, *Towards ip geolocation using delay and topology measurements*, in Proc. IMC, 2006.

[6] N. Korula and S. Lattanzi, *An efficient reconciliation algorithm for social networks*, PVLDB, 7 (2014), pp. 377–388.

[7] S. Kumar, R. Zafarani, and H. Liu, *Understanding user migration patterns in social media*, in AAAI, 2011.

[8] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, *Hydra: Large-scale social identity linkage via heterogeneous behavior modeling*, in Proc. SIGMOD, 2014.

[9] X. Mu, F. Zhu, E. P. Lim, J. Xiao, J. Wang, and Z. H. Zhou, *User identity linkage by latent user space modelling*, in Proc. KDD, 2016.

[10] A. Narayanan and V. Shmatikov, *Robust de-anonymization of large sparse datasets*, in Proc. IEEE SP, 2008.

[11] Pewinternet, *Social Media Update 2016*, http://www.pewinternet.org/2016/11/11/social-media-update-2016/.

[12] D. M. Powers, *Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation*, (2011).

[13] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, *Linking users across domains with location data: Theory and validation*, in Proc. WWW, 2016.

[14] L. Rossi and M. Musolesi, *It's the way you check-in: identifying users in location-based social networks*, in Proc. COSN, 2014.

[15] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu, *User identity linkage across online social networks: A review*, ACM SIGKDD Explorations Newsletter, 18 (2017), pp. 5–17.

[16] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, *Measuring serendipity: connecting people, locations and interests in a mobile 3g network*, in Proc. IMC, 2009.

[17] M. Yan, J. Sang, T. Mei, and C. Xu, *Friend transfer: cold-start friend recommendation with cross-platform transfer learning of social knowledge*, in Proc. ICME, 2013.

[18] C. Yang, H. Yan, D. Yu, Y. Li, and D. M. Chiu, *Multi-site user behavior modeling and its application in video recommendation*, in Proc. ACM SIGIR, 2017.

[19] R. Zafarani and H. Liu, *Connecting users across social media sites: a behavioral-modeling approach*, in Proc. ACM SIGKDD, 2013.

[20] ———, *Finding friends on a new site using minimum information*, in Proc. SDM, 2014.

[21] J. Zhang, X. Kong, and P. S. Yu, *Transferring heterogeneous links across location-based social networks*, in Proc. WSDM, 2014.

*https://github.com/whd14/SIMP