Automatic User Identification Method across Heterogeneous Mobility Data Sources

Wei Cao[1,2], Zhengwei Wu[2], Dong Wang[1], Jian Li[1], Haishan Wu[2*]

[1] Institute for Interdisciplinary Information Sciences, Tsinghua University
[2]Big Data Lab, Baidu Inc
{cao-w13@mails, wang-dong12@mails, lijian83@mail}.tsinghua.edu.cn
{wuzhengwei, wuhaishan}@baidu.com

***Abstract*—With the ubiquity of location based services and applications, large volume of mobility data has been generated routinely, usually from heterogeneous data sources, such as different GPS-embedded devices, mobile apps or location based service providers. In this paper, we investigate efficient ways of identifying users across such heterogeneous data sources. We present a MapReduce-based framework called *Automatic User Identification* (AUI) which is easy to deploy and can scale to very large data set. Our framework is based on a novel similarity measure called the *signal based similarity* (SIG) which measures the similarity of users' trajectories gathered from different data sources, typically with very different sampling rates and noise patterns. We conduct extensive experimental evaluations, which show that our framework outperforms the existing methods significantly. Our study on one hand provides an effective approach for the mobility data integration problem on large scale data sets, i.e., combining the mobility data sets from different sources in order to enhance the data quality. On the other hand, our study provides an in-depth investigation for the widely studied human mobility uniqueness problem under heterogeneous data sources.**

I. Introduction

Ubiquitous location based services and applications have enabled people to use GPS-embedded devices for navigation, travel planning and geolocation information sharing in their daily life. Such mobility data is now collected routinely at a very large scale. The large volume of mobility data gives rise to new opportunities for discovering patterns and characteristics of human mobility behaviors. An increasing number of researches empowered by mobility data has emerged recently. Meanwhile, mining of such mobility data also shows great potentials in various industrial and commercial applications, including traffic analysis [1]–[3], travel recommendation [4]–[11], location-based social network [12]–[17], geographical searching [18], [19] etc.

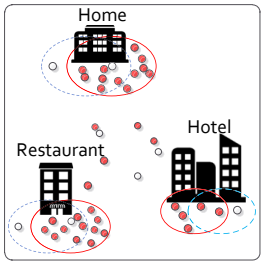
In real applications, mobility data is usually generated from heterogeneous data sources, such as different GPS-embedded devices, mobile apps, or LBS providers etc. In this paper, we aim to study the efficient approach of identifying users from mobility datasets collected from heterogeneous sources. We first present two motivations of our work. On one hand, an effective user identification algorithm is the fundamental ingredient of the commonly faced mobility data integration problem [20]–[22], which aims to improve the quality and density of the data by fusing multiple mobility data sets collected from different sources. On the other hand, as the *human mobility uniqueness* problem being widely studied recently [23]–[25], our work provides an in-depth investigation for the uniqueness of human mobility under heterogeneous data sources.

A closely related topic is *user similarity search* where the goal is to retrieve a subset of users with similar spatio-temporal patterns. Essentially, we can think of the user identification problem as a special case of user similarity search problem. For each trajectory, we retrieve the most similar ones from another data source and identify whether they belong to the same person.

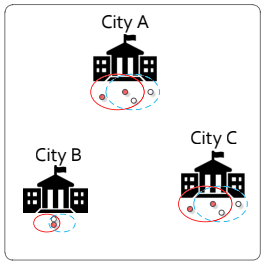
Despite the fact that the user similarity search has been studied quite extensively [5], [26]–[29], not much work has investigated the mobility data collected from heterogeneous data sources, which is much more complicated. In our experiments, the data is collected from various mobile apps, including the navigation data, map queries, geo-tagged records in social platforms etc (see Section VI for details). To make our exposition more concrete, we first illustrate some distinct features of our data and our objective.

- The most distinctive feature we have found about the trajectories is the sheer variety of the sampling rates among different sources. For example, trajectories of GPS navigation data are usually sampled at a very high rate whereas the geo-tags or map queries are sampled with an extremely low rate. Even in the same data source, the utilization frequencies could vary drastically during different time intervals. On the other hand, most prior work uses approximately uniformly (and often densely) sampled data [5], [26], [27], [30].
- For some sparse trajectories, it is almost impossible to infer any information of user movement. For example, for the geo-tagged check-in data in our data set, each user only generated one GPS record every 2.63 days on average. Most of prior work measures the trajectory similarities based on the movement behaviors of the users such as the speed, move direction, spatial-temporal closeness (i.e., two trajectories are similar only if they appeared in approximately the same place at approximately the same time) [26], [30]–[34]. However, such features are not available in our data due to the extreme sparsity.
- The trajecotries of the users with close relationship usually have a significant overlap. For example, we investigate the mobility data of several students who study in University T. Most of their trajectories lie on their department buildings and dormitories. Such

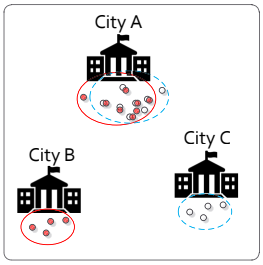
*Corresponding author.



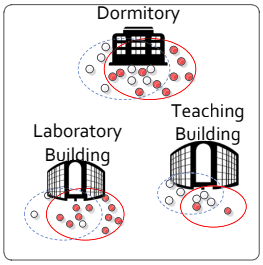
(a) Trajectories of the same person which are sampled at very different rates.



(b) Trajectories of the same person which co-occurred in several places far apart from each other.



(c) Trajectories of the same person which are disjoint in several cities.



(d) Trajectories of two school mates which have significant overlap.

Fig. 1. Four typical cases observed in the real dataset

overlap renders it difficult to distinguish them. However, few prior work investigates user identification problem under such case.

- For different data sources, the trajectories can be temporally disjoint. For example, two data sets of the same group of anonymized users with inconsisten user id, one is collected at January and another is collected at February. Especially, for the businessmen, they may go to several different cities during several months which makes it difficult to measure their similarities.

To provide some intuitions for the readers and to illustrate the challenges, we show an example in Example 1.

Example 1: In Figure 1, we show four typical cases observed in the real datasets (there are many other cases or combinations of those cases, that are impossible to list exhaustively).

Each of (a)(b)(c) represents two trajectories (from different data sources) that belong to the same person and (d) shows the trajectories that belong to two schoolmates.

- *In (a), the white trajectory is sampled at a much lower rate than the red trajectory but they both occurred in several fixed places frequently. Such co-occurrences are significant for identifying the same person, especially when they take place far apart from each other, such as several different cities, as shown in (b).*
- *However, in (c), we observed from the data that the trajectories of the same person can be also disjoint in several cities. Such case often happens when the trajectories are temporally disjoint as well. However, as they co-occurred significantly in one city (city A), we can still identify that they belong to the same person.*
- *In (d), as the trajectories in the same campus have significant overlap, it is hard to distinguish the users from their schoolmates. Nevertheless, the red trajectories occurred more in the laboratory building while the white trajectory tends to go to the teaching building more. Such pattern enables us to identify them uniquely.*■

The above features make our user identification problem very different from the previous trajectory similarity problems [5], [26]–[28], [30], [33]–[36] in that the trajectories we deal with are sampled at very different rates, and extremely noisy.

To address the challenge, we propose a MapReduce-based framework, called *Automatic User Identification* (AUI), which is based on a novel trajectory similarity measure. AUI is easy to deploy and can scale to very large data set. We summarize our technical contributions below:

- We formulate the user identification problem over large scale heterogeneous mobility datasets and we present a MapReduce-based frame called AUI.
- We design an effective filtering strategy based on the MapReduce-based framework. With the filtering strategy, for each trajectory we only need to compare it with a small number of candidates. The filtering strategy is the foundation of that AUI can scale to very large data sets.
- We design a novel similarity measure called the *signal based similarity* (SIG) by considering the frequencies of the co-occurrences and the locations where they took place. Since our data is collected from many different data sources, we do not assume any property of the mobility data (e.g., sample rate, time span). We show that compared with the existing measures, our measure can handle the extremely noisy cases effectively. The experiment result shows that our measure is more robust and accurate for our user identification problem with heterogeneous data sources.
- We adopt a rejection strategy in order to reduce the mis-identification cases. Many application scenarios are highly sensitive to misidentification, i.e. a few erroneous cases could lead to serious consequences. Our strategy enhances the accuracy of the framework significantly.
- We evaluate our framework by 6 experiments of different cases. For the easiest case (31511 users in China), we achieve an accuracy of 99.94%. For the hardest case (14115 college students who study in the same campus), we achieve an accuracy of 90.09% whereas the best existing method only achieves an accuracy of 61.38% in this case.

II. Formulation and Overview

The problem studies the user identification across heterogeneous data sources. We first present several useful definitions.

Definition 1 (Trajectory): A trajectory $T = \{p_1, p_2, \ldots, p_{|T|}\}$ is a temporally ordered sequence of spatio-temporal points. Each point p_i is associated with three attributes x, y, t where the pair (x, y) [1] indicates the coordinate of p_i and t indicates the timestamp when p_i was recorded.

Here we stress that the trajectories in our data could be extremely sparse, i.e., there may be less than one spatio-temporal point per day in average.

Definition 2 (Mobility data set): A mobility data set D is defined as a collection of trajectories. Each trajectory $T \in D$ is associated with an id $T.id$.

Definition 3 (Matching trajectory): Suppose two trajectories T_A, T_B are collected from different mobility data sets. If T_A and T_B are generated from the same user, then we call T_B a matching trajectory of T_A.

Our problem is defined as follow. Given two mobility data sets D_A and D_B (usually collected from two different data sources), for each $T_A \in D_A$, our goal is to identify whether there exists $T_B \in D_B$ which is the matching trajectory of T_A. Moreover, it is guaranteed that for each T_A, there exits at most one matching trajectory in D_B. We do not assume any other property of the mobility data set. Thus, the sampling rates of the trajectories could be very high or extremely low. Furthermore, D_A and D_B could be temporally disjoint, i.e., they are collected at different time intervals.

We explain the reason why we can achieve user identification across heterogeneous mobility data sources. Montjoye et al. [23] showed that the human mobilities are highly unique. Each individual has her/his own mobility pattern. People tend to visit the places where they often visited in the past. Even for the users with very close relationship, they still have noticeable different mobility patterns as we illustrated in Fig 1(d). Such uniqueness of human mobility allows us to identify trajectories that belong to the same user from different sources.

To handle the extremely sparse trajectories (which makes it hard to infer any mobility pattern from the daily data), we accumulate the trajectories during a long time interval (for example, the trajectories during 3 months). Thus, by accumulating the historical mobility data, it is possible to infer the mobility pattern of a user such as the places he/she tends to visit. Fig. 2 shows an example in our data set where the trajectory is collected from February to May in 2015 with 5 points per day in average. By accumulating all these points, we can clearly see that this user usually stay at home and the workplace. Moreover, he/she had visited Wangfujing and Beijing Botanical Garden one day in the past 3 months. Motivated by this, we consider both the frequency of spatial co-occurrences (i.e., two trajectories co-occurred at approximately the same location) under different granularities and the locations where they took place. We define our trajectory similarity mainly based on these factors.

We first present an overview of AUI. Our framework consists of three stages: the pre-processing stage, the multi-resolution filtering stage and the verification stage. All of the stages are implemented on the MapReduce frame. In the pre-processing stage, we first perform data compression by transforming each trajectory into consecutive transitions between a set of *stay points*, i.e., the locations where the user stays for a while rather than just passing by. Since for large scale data sets, pairwise comparison is rather expensive, in the multi-resolution filtering stage, we partition the map into multiple grids with multi-resolution. For each trajectory $T_A \in D_A$, we select a small subset of trajectories in D_B which co-occurred frequently in multiple cells with T_A as the candidate set of T_A. Finally, as the trajectories we deal with are sampled at very different rates and extremely noisy, in the verification stage, we evaluate each candidate with the signal based similarity which can handle such cases effectively and we carefully select the matching trajectory. We further extend our similarity measure and adopt an effective rejection strategy in the verification stage to reduce the misidentification cases (Section V).

[1] We use the mercator coordinate in our experiment.

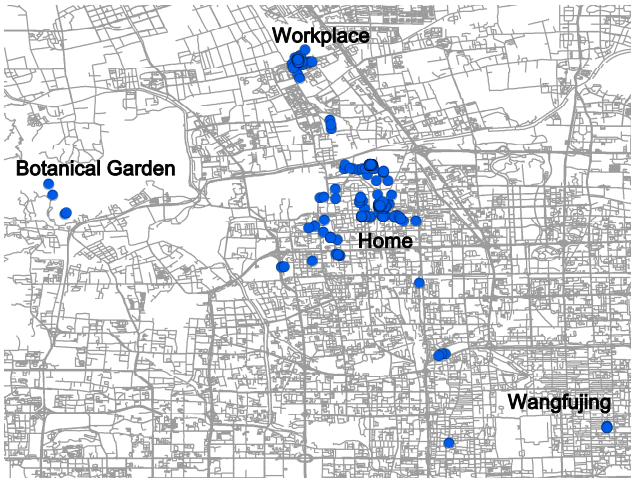


Fig. 2. A trajectory in our data.

III. Pre-Processing

The pre-processing stage transforms each trajectory into a set of stay points, i.e., the location points that the user stay for a while rather than passing by. Thus stay points usually carry the particular semantic meanings such as the building they live or the park they went [27]. The pre-processing stage consists of map-only jobs. For each trajectory $T \in D_A/D_B$, the map function takes $T.id$ as the key and T as the value. We use a similar method as in [27] to pre-process the trajectories. For each trajectory in a mobility data set, we split it into consecutive segments. A segment is defined as a series of continuous location points sharing the same mobility status, such as stay, move and pass-by. For example, a user drove for 2 hours to a shopping mall and spent 3 hours in the mall, then drove to another place. Such trajectory can be split into 3 segments: driving to the mall, staying at the mall and driving to another place. For each segment under stay status, we use the geometric center of the segment as the corresponding stay point. We then use a series of stay points as the compressed trajectory. A location point is considered as under stay status if the user spent more than Δ_t^l minutes but less than Δ_t^u minutes around this point within a distance of Δ_d meters. Comparing with [27], we set an additional upper bound Δ_t^u here (typically 12 hours). The reason is that in some sparse trajectories the time gap between two consecutive location points can exceed several days, which renders it difficult to infer the user's mobility status during the gap. For such cases, we regard the latter location point as the start point of a new segment. Furthermore, we stress that the pre-processing stage mainly effect on the non-sparse trajectories. For the extremely sparse trajectories, almost

Algorithm 1 Pre-Processing

Map: $(\langle T.id, T\rangle)$
1: $S \leftarrow \{\}$, $P \leftarrow \{p_1\}$;
2: **for** $i = 2 \ldots |T|$ **do**
3: $p_b \leftarrow P[1]$;
4: $t \leftarrow p_i.t - p_b.t$;
5: $d \leftarrow \mathrm{Dis}(p_i, p_b)$;
6: **if** $d \geq \Delta_d$ or $t \geq \Delta_t^u$ **then**
7: **if** $t \geq \Delta_t^l$ **then**
8: $sp.loc \leftarrow \mathrm{MeanCoordinate}(P)$; ▷ stay point
9: $sp.cnt \leftarrow |P|$; ▷ number of around points
10: $S.\mathrm{add}(sp)$; ▷ add the stay point into S
11: **end if**
12: $P \leftarrow \{p_i\}$; ▷ start a new segment
13: **else**
14: $P.\mathrm{add}(p_i)$; ▷ add p_i into current segment
15: **end if**
16: **end for**
17: $\mathrm{emit}(\langle T.id, S\rangle))$

a single spatio-temporal point can represent a stay point. See Algorithm 1 for the pseudo-code.

IV. Multi-Resolution Filtering

Since the pair-wise comparison is expensive, especially for the large data sets, in the multi-resolution filtering stage, for each $T_A \in D_A$, we only select a small subset from D_B as its candidates. The multi-resolution filtering stage contains two phases. In the first phase, for each $T_A \in D_A$, we gather the "significant" co-occurrences with T_A. In the second phase, we evaluate the gathered co-occurrences and select a small subset of trajectories as the candidate set of T_A.

Formally, in the first phase, we partition the map into N grids with different granularities, i.e., the side length of cells in the grid. We use $G_1, \ldots, G_N$ to denote these grids. The map function takes the trajectory id $T.id$ as the key and the corresponding stay points S as the value. For each G_i, we enumerate the stay point $sp \in S$ and emit the key-value pairs $\langle c, \langle T.id, sp.cnt\rangle\rangle$, which indicates that $T.id$ occurred in a cell c for $sp.cnt$ times. Here c represents a tuple $\langle cx, cy, sz\rangle$ where $\langle cx, cy\rangle$ is the coordinate of the cell in the grid and sz is the corresponding side length. Note that since different grids are partitioned into different granularities, the tuple $\langle cx, cy, sz\rangle$ indicates a specific cell uniquely. The trajectory ids occurred in the same cell thus must be shuffled into the same reduce task.

In the reduce stage, each reduce function takes a key value pair $\langle c, list(\langle T.id, T.cnt\rangle)\rangle$ as input. If the length of $list(\langle T.id, T.cnt\rangle)$ exceeds m_c, we simply drop the cell c. Here m_c is a parameter to be specified (see Section VI for details). Such cells usually correspond to the "common places" of the users such as the subway stations, which are not significant for the identification. Dropping such cells on one hand accelerates the algorithm. On the other hand, in Section VI, we show that it enhances the performance of our algorithm. For the rest of the cells, the reduce function splits $list(T.id)$ into two groups, the ids from D_A and the ids from D_B, denoting as $list(T_A.id)$ and $list(T_B.id)$. (To ensure the source of trajectory id is distinguishable, we add different special marks to the trajectory ids in different data sources.) Each pair $(T_A.id, T_B.id)$ from the two lists is a candidate pair. For each candidate pair, we emit a key-value pair $\langle T_A.id, \langle T_B.id, c, o\rangle\rangle$ which indicates that $T_A.id$ and $T_B.id$ co-occurred in the cell c for o times. Here the co-occurred frequency o is obtained by $\min\{T_A.cnt, T_B.cnt\}$.

Algorithm 2 Multi Resolution Filtering

Map: $(\langle T.id, S\rangle)$
1: **for** $i = 1 \ldots N$ **do**
2: $sz \leftarrow$ the cell size of G_i
3: **for** $sp \in S$ **do**
4: $cx, cy \leftarrow$ the cell coordinate that sp lies in G_i
5: $c \leftarrow (cx, cy, sz)$
6: $\mathrm{emit}(\langle c, \langle T.id, sp.cnt\rangle\rangle)$
7: **end for**
8: **end for**

Reduce: $(\langle c, list(\langle T.id, T.cnt\rangle)\rangle)$
9: **if** $len(list(\langle T.id, T.cnt\rangle)) \leq m_c$ **then**
10: $l_A \leftarrow$ items of $list(\langle T.id, T.cnt\rangle)$ where $T.id \in D_A$
11: $l_B \leftarrow$ items of $list(\langle T.id, T.cnt\rangle)$ where $T.id \in D_B$
12: **for** $\langle T_A.id, T_A.cnt\rangle \in l_A$ **do**
13: **for** $\langle T_B.id, T_B.cnt\rangle \in l_B$ **do**
14: $o \leftarrow \min\{T_A.cnt, T_B.cnt\}$ ▷ co-occurrences
15: $\mathrm{output}(T_A.id, \langle T_B.id, c, o\rangle)$
16: **end for**
17: **end for**
18: **end if**

//second phase
Map: $(\langle T_A.id, \langle T_B.id, c, o\rangle\rangle)$
19: $\mathrm{emit}(\langle T_A.id, \langle T_B.id, c, o\rangle\rangle)$

Reduce: $(\langle T_A.id, list(\langle T_B.id, c, o\rangle)\rangle)$
20: **for** $\langle T_B.id, c, o\rangle \in list(\langle T_B.id, c, o\rangle)$ **do**
21: $sz \leftarrow$ the size of c
22: Increase the ranking score of $T_B.id$ by $(r_{sz} \cdot o)$
23: **end for**
24: $\mathcal{I} \leftarrow$ top Q trajectory ids with largest ranking scores
25: **for** $T_B.id \in \mathcal{I}$ **do**
26: Merge the items of $T_B.id$ into one key-value pair $\langle T_A.id, \langle T_B.id, list(\langle c, o\rangle)\rangle\rangle$
27: $\mathrm{output}(\langle T_A.id, \langle T_B.id, list(\langle c, o\rangle)\rangle\rangle)$
28: **end for**

In the second phase, the map function simply emits its input $\langle T_A.id,\ \langle T_B.id, c, o\rangle\rangle$. Thus, the key-value pairs with the same $T_A.id$ are shuffled into the same reduce task and each reduce task takes the key-value pair $\langle T_A.id, list\,(\langle T_B.id, c, o\rangle)\rangle$ as the input. We evaluate each $T_B.id$ in $list\,(\langle T_B.id, c, o\rangle)$ with a ranking score. Formally, for each co-occurrence with $T_B.id$ in a cell with side length sz, we increase the ranking score of $T_B.id$ by a parameter r_{sz}. The finer granularity they co-occurred, the larger ranking score we increase. We select the top Q ids with the largest ranking scores as the candidate set of T_A. For each candidate $T_B.id$, we merge the related co-occurrences into one key-value pair $\langle T_A.id, \langle T_B.id, list(\langle c, o\rangle)\rangle\rangle$. See Algorithm 2 for the pseudo code.

We show a running example in Fig. 3. The grid in Fig. 3 is partitioned into small cells. The side length of each cell is 20 meters. The circle points are the stay points of user id_A and the triangle points are the stay points of user id_B. In the map stage of the first phase, for each cell we emit the users who occurred in this cell with corresponding frequency. By shuffling and grouping the keys, in the reduce stage we output

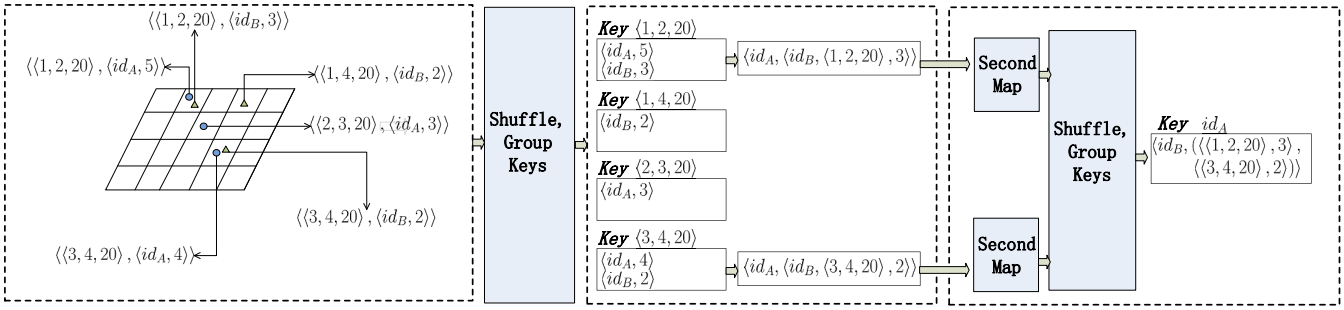


Fig. 3. Running example of the multi-processing stage

the co-occurrences of id_A. Finally, in the second phase of the multi-resolution stage, we select the candidates of id_A and merge the related co-occurrences.

V. Verification Stage

In the verification stage, we evaluate each candidate of T_A and carefully select the matching trajectory. We first present the signal based similarity in Section V-A which is the foundation of the verification. Next, we present the algorithm of verification in Section V-B.

A. Signal Based Similarity

Intuitively, the signal based similarity takes a pair of trajectories as input and observes the co-occurrences sequentially. Each co-occurrence is regarded as a "signal" which indicates that two trajectories might belong to the same user. The similarity is the final signal when the whole sequence is processed. The stronger the final signal is, the more similar the two trajectories are. We distinguish two kinds of signals, the *observed signal* and the *stimulus signal*. The observed signal is directly calculated by the co-occurrences in each cell. The strength of the observed signal increases as the frequency of co-occurrences increases. However, such co-occurrences may be affected by some "kernel places". For example, in Figure 4, the two trajectories co-occurred frequently at the company, as a result, they are also frequently observed co-occurring at the nearby bus station. We refer to the company in this example as a kernel place. To capture such spatial correlation feature, we introduce the stimulus signal. We assume that there exist several *kernel cells* initially. Each kernel cell emits a positive stimulus signal. Each stimulus signal spreads out spatially from the kernel cell with an attenuation factor $\alpha < 1$. We consider the observed signal as the superposition of the decaying stimulus signals. The signal based similarity extracts the kernel cells and recovers the strengths of the stimulus signals from the observed signals. The final signal is calculated by considering both the strength of stimulus signals and the distances between the kernel cells.

Formally, suppose we are given two trajectories $T_A \in D_A$ and $T_B \in D_B$. We consider each grid G_i separately. We use (c_1, o_1), ..., (c_m, o_m) to denote the observed co-occurrences in G_i. Here c_k represents the k-th co-occurred cell and o_k represents its corresponding frequency. For simplicity, we do not require a specific order here. We first calculate the observed signal in each cell. Note that the frequency of co-occurrences has the following diminishing marginal utility property: when many co-occurrences at a specific cell have been observed, further co-occurrences at the same cell can only contribute

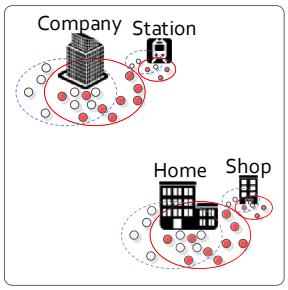


Fig. 4. The company and the home represent two kernel places.

limited information. To capture such property, we use a sigmoid function $f_s(o_k; \eta, \gamma)$ to calculate the observed signal in c_k:

$$\mathsf{ob}(c_k) = f_s(o_k) = \frac{\eta}{1 + e^{-\gamma o_k}} - \frac{\eta}{2} \tag{1}$$

where η, γ are two parameters to be specified (see Section VI for details).

To recover the stimulus signals efficiently, we simply process the observed signals sequentially to approximate the stimulus signals. For the cell c_k, we assume that it is only affected by the cells $c_1, \ldots, c_{k-1}$. Note that the stimulus signals we obtained may be different under different orders of cells. However, in the experiments, we find a specific order rarely affects the accuracy for user identification. Specifically, we use $\mathsf{st}(c_k)$ to denote the stimulus signal in cell c_k. Initially, we have that $\mathsf{st}(c_1) = \mathsf{ob}(c_1)$. For any $k > 1$, we have

$$\mathsf{st}(c_k) = \max \left\{ \begin{array}{l} \mathsf{ob}(c_k) - \sum_{l<k} \mathsf{st}(c_l) \cdot \alpha^{\mathrm{Dis}_{\mathrm{grid}}(c_k, c_l)} \\ 0 \end{array} \right. . \tag{2}$$

where $\mathrm{Dis}_{\mathrm{grid}}$ is the distance metric defined on the grid. In our algorithm, we use Euclidean distance of cell centers scaled by the side length of the cell as $\mathrm{Dis}_{\mathrm{grid}}$. Thus, the kernel cells are the cells with non zero stimulus signal. We use K to denote the indices of kernel cells, i.e., $K = \{k : \mathsf{st}(c_k) > 0\}$.

Next, we take the distances between the kernel cells into consideration. We use $\mathsf{md}(c_k)$ to denote the minimum distance from cell c_k to the previous kernel cells, i.e.,

$$\mathsf{md}(c_k) = \min_{(l<k) \wedge (l \in K)} \mathrm{Dis}_{\mathrm{grid}}(c_k, c_l).$$

Algorithm 3 Signal Based Similarity for T_A and T_B

for $i = 1 \dots N$ **do**
  $K \leftarrow \{\}$
  for $k = 1 \dots m$ **do**
    $o \leftarrow$ co-occurrences in cell $c_k \in C$;
    $\mathsf{ob}(c_k) = f_s(o)$;
    if $k == 1$ **then**
      $\mathsf{st}(c_1) \leftarrow \mathsf{ob}(c_1)$
    else
      $\mathsf{st}(c_k) \leftarrow \max\{0, \mathsf{ob}(c_k) - \sum_{l \in K} \mathsf{st}(c_l) \cdot \alpha^{\mathrm{Dis}_{\mathrm{grid}}(c_k, c_l)}\}$;
    end if
    if $\mathsf{st}(c_k) > 0$ **then**
      K.add(k);
    end if
  end for
  $\mathsf{sig}_i = \mathsf{st}(c_1) + \sum_{k \in K \setminus \{1\}} \mathsf{st}(c_k) \cdot (1 + f_d(\mathsf{md}(c_k)))$
end for
$\mathsf{SIG} = \sum_{i=1}^{N} \beta_i \cdot \mathsf{sig}_i$;

Algorithm 4 Verification

Map: $(\langle T_A.id, \langle T_B.id, list(\langle c, o\rangle)\rangle\rangle)$
calculate the signal based similarity SIG
calculate the weighted jaccard similarity WJS
if $\mathsf{SIG} \geq \theta_{\mathsf{SIG}}$ and $\mathsf{WJS} \geq \theta_{\mathsf{WJS}}$ **then**
  emit$(\langle T_A.id, \langle T_B.id, \mathsf{SIG}, \mathsf{WJS}\rangle\rangle)$
end if

Reduce: $(\langle T_A.id, list(\langle T_B.id, , \mathsf{SIG}, \mathsf{WJS}\rangle)\rangle$
if $\exists T_B^*.id \in list(T_B.id)$ dominate all the others **then**
  output$(\langle T_A.id, T_B^*.id\rangle)$
end if

Similarly, we define a sigmoid function $f_d(\mathsf{md}(c_k); \lambda, \mu)$ which has the same form as f_s with parameters λ and μ to be specified. Then, the signal in the grid G_i is:

$$\mathsf{sig}_i = \mathsf{st}(c_1) + \sum_{k \in K \setminus \{1\}} \mathsf{st}(c_k) \cdot (1 + f_d(\mathsf{md}(c_k))) \quad (3)$$

The equation 3 essentially captures the feature that we showed in Fig. 1(c), i.e., it is significant to observe the co-occurrences taking far apart from each other.

Finally, we sum the signals of all the grids. We set a weight parameter β_i for the G_i. Again, the finer granularity G_i is, the larger weight parameter β_i becomes. Thus, the signal based similarity is defined as:

$$\mathsf{SIG} = \sum_i \beta_i \cdot \mathsf{sig}_i \quad (4)$$

See Algorithm 3 for the pseudo code.

B. Verification

In the verification stage, each map function takes a key-value pair $\langle T_A.id, \langle T_B.id, list(\langle c, o\rangle)\rangle\rangle$. Note that $list(\langle c, o\rangle)$ contains all the co-occurrences in all grids. Thus, we can directly calculate the signal based similarity from $list(\langle c, o\rangle)$.

To verify the candidates, a feasible way is to measure each candidate of T_A with the signal based similarity and select the most similar one as the matching trajectory. In the experiment section, we show that such strategy achieves a reasonable performance. However, many application scenarios are highly sensitive to misidentification, i.e. a few erroneous cases could cause serious consequence. It is necessary to adopt a rejection strategy, i.e., to refuse to identify the trajectories that may lead to mistakes. In Section VI, we show that it is difficult and inaccuracy to obtain a rejection strategy with only the signal based similarity. Motived by this, we adopt an effective rejection strategy with another similarity measure called the weighted Jaccard similarity which is applied in combination with the signal based similarity. The weighted Jaccard similarity was first proposed by Ioffe et al. [37]. It measures the similarity between two weighted set. For our problem, we regard each trajectory as a set of cells it has visited. The weight of each cell is the corresponding visiting frequency. Thus, for a given pair of trajectories, it measures the similarity of the places they visited and the corresponding frequencies. For a specific grid G_i , we use v_c^A (v_c^B resp.) to denote the frequency of T_A (T_B resp.) being observed at cell c. The similarity of T_A and T_B on grid G_i is defined as:

$$\mathsf{wjs}_i = \frac{\sum_c \min\{v_c^A, v_c^B\}}{\sum_c \max\{v_c^A, v_c^B\}}. \quad (5)$$

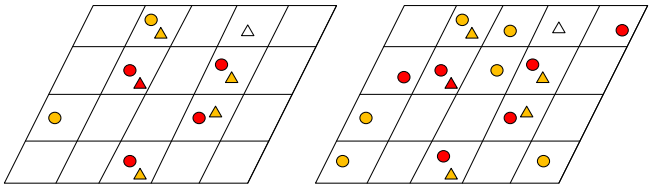


Fig. 5. Example of the weighted Jaccard similarity.

We show an example in Fig. 5. The points in the same shape (circle or triangle) indicate a specific id. We use the red color to indicate that the user occurred in this cell for 5 times. Similarly, we use yellow color to represent the frequency of 3 and the white color to represent the frequency of 1. Thus, the weighted Jaccard similarity in the left figure is $17/27 = 0.63$ and the weighted Jaccard similarity in the right figure is $17/49 = 0.35$.

Note that the term $\min\{v_c^A, v_c^B\}$ is exactly the frequency of co-occurrence of T_A and T_B in cell c which is already contained in $list(\langle c, o\rangle)$, all we need to do is calculate the term $\max\{v_c^A, v_c^B\}$. Re-writing the denominator as $\sum_c v_c^A + \sum_c v_c^B - \sum \min\{v_c^A, v_c^B\}$, we find that the first two terms are exactly the accumulated frequency of all the stay points which can be obtained easily at the beginning. Thus, the weighted Jaccard similarity on G_i also can be calculated from $list(\langle c, o\rangle)$. Similarly, we use $\mathsf{WJS} = \sum_i \beta_i \cdot \mathsf{wjs}_i$ as the weighted Jaccard similarity of T_A and T_B.

We first calculate the signal based similarity and the weighted Jaccard similarity for all the candidates. Next, we set two *small* thresholds θ_{SIG} and θ_{WJS} to filter out the dissimilar candidates. If a candidate $T_B.id$ has its signal based similarity smaller than θ_{SIG} or its weighted Jaccard similarity smaller than θ_{WJS}, we filter it out from the candidate set. After the filtering, if there exists a unique maxima in the candidate set, we return it as the matching trajectory. Otherwise, we reject the identification. A candidate is called a maxima if there do not exist another candidate with both of the two similarities greater than it. For

example, suppose we have 3 candidates with $(\mathsf{SIG}, \mathsf{WJS})$ equal to $(12.7, 0.4), (13.1, 0.3), (10, 0.35)$ respectively. Then the first two candidates are both maxima. See Algorithm 4 for the pseudo code.

We stress that when the mobility data sets are sampled in very different rates, the weighted Jaccard similarity becomes very low which leads to a great error. However, in the experiment section, we show that by combining with the signal based similarity and setting an extremely small threshold θ_{WJS}, AUI achieves a great performance.

VI. Experiment Evaluation

In this section, we conduct extensive experiments on real mobility data sets to demonstrate the performance of the proposed algorithm. We first describe our experiment setting in Section VI-A then we continue by presenting the effects of the parameters in Section VI-B. We further compare our algorithm with the existing methods in Section VI-C. Finally, in Section VI-D we give a discussion of our experimental results.

A. Experiment Setting

Our data set is collected from users who shared location data using different mobile apps of Baidu Inc. The data sources include the navigation data, map queries, geo-tagged records in social platforms etc. During our experiment, all the user id were anonymized by hashing. The trajectories of the same user have the consistent hashing id which we use it as the ground truth.

We distinguish two kinds of data sets, *the dense mobility data set* and *the sparse mobility data set*. The sources of dense mobility data sets contains the navigation data and the GPS location data. The sources of sparse mobility data sets contains map queries, geo-tagged check-in data etc. Thus, the trajectories in the dense mobility data are sampled at a very high rates whereas the trajectories in the sparse mobility data are sampled at a low rates.

We randomly select 31511 users in China and extract the corresponding mobility data from the dense mobility data set, denoting as *CN-Dense*. The time span of CN-Dense is from August, 2014 to February 2015. Among these users, 6396 users can be found in the sparse mobility set. We extract the mobility data of these 6396 users from the sparse mobility set, denoting as *CN-Sparse*. For the first experiment, we use CN-Sparse and CN-Dense as D_A and D_B respectively as we described in Section II.

Next, to evaluate our algorithm in the case that the trajectories are temporally disjoint, we split CN-Dense into two parts. The first part is from August 2014 to November 2014, denoting as *CN-Part1* and the second part is from December 2014 to February 2015, denoting as *CN-Part2*. Note that some trajectories only appeared during one of the time intervals. Thus, after splitting, the number of trajectories of each data set can be less than 31511. Furthermore, only part of the trajectories in D_A have the matching trajectories.

We further select 4323 company employees who work in Company B and 14115 college students who study in University T. We design another 4 experiments with the same method we used in our first two experiment settings.

To pre-process the data, we set $\Delta_d = 100$m and $[\Delta_t^l, \Delta_t^u] = [0.5\text{h}, 12\text{h}]$ (Algorithm 1). Thus, a location point is considered as a stay point if the user stays around the point within 100 meters for more than half an hour but less than 12 hours. We compare the average number of location points each trajectory contains before and after the pre-processing. The details as shown in Table I.

We stress that the first two experiments (CN-Sparse vs. CN-Dense and CN-Part1 vs. CN-Part2) form the easiest two cases, since these users are randomly chosen from a large population distributed on a vast spatial domain, which renders it easy to uniquely identify them even under coarse granularity. The last two experiments (UT-Sparse vs. UT-Dense and UT-Part1 vs. UT-Part2), form the hardest two cases since these students all study and live in the same campus and it is difficult to distinguish any of them from their schoolmates.

All the algorithms are implemented in Python and ran under streaming mode of Hadoop system (Release 2.6.0). All the experiments are conducted on a Hadoop Cluster with 10 nodes. Each node corresponds to a computer with a Intel Xeon E312 CPU of 2 cores (2.1GHz for each core) and a 8G memory.

B. Effects of Parameters

Before we show the experiment results, we first give two useful definitions.

Definition 4 (hitting rate): Given a specific trajectory $T_A \in D_A$ and its candidate set, if the matching trajectory of T_A is contained in its candidate set, we say we "hit" T_A. Suppose there are H trajectories in D_A which is hit. Then the *hitting rate* of D_A is defined as $H/|D_A|$.

Definition 5 (coverage, accuracy): Suppose we use a single similarity measure S to identify the users. For each $T_A \in D_A$, denote its candidate set as $\mathcal{C}$. If

$$\max_{T_B.id \in \mathcal{C}} \mathrm{S}(T_A, T_B) \leq \theta$$

where θ is a given threshold, then we refuse to identify T_A. Otherwise, we select the candidate with the largest similarity as the matching trajectory. Suppose there are Rej trajectories which we refuse to identify and there are Cor trajectories which are correctly identified. Then the *coverage* is defined as $1 - \frac{Rej}{|D_A|}$ and the *accuracy* is defined as $\frac{Cor}{|D_A| - Rej}$.

Note that Definition 5 defines the coverage of a single similarity measure. For AUI, since it utilizes a pair of similarity measures and selects the matching trajectory only if it is the unique maxima. The coverage of AUI is essentially associated with the "cohesion" of the data sets and can be determined automatically, i.e., the coverage would be much lower if users within the data sets are spatially correlated or exhibit a high degree of social homophily.

Our default parameter setting is presented in Table II. To illustrate the effects of different parameters, each time *we only change one parameter and keep the others unchanged.*

1) Effects of Q: Recall that in the multi-resolution filtering stage, for each $T_A \in D_A$, we select a candidate set of size Q (Algorithm 2). If Q is too small, it is easy to miss the matching trajectories in their candidate set. However, the results show that only choosing $Q = 20$ is enough to achieve the hitting rates from 91.61% to 99.66%. A higher hitting rate can be obtained by choosing larger value of Q, which would increase the time cost of the verification stage at the same time, as shown in Table III.

TABLE I. DESCRIPTION OF THE DATA SETS

Data set	Number of trajectories	Average points per trajectory before compression	Average points per trajectory after compression	Time span (Year-Month-Day)
CN-Sparse	6396	81.31	20.47	14-08-01 to 15-02-28
CN-Dense	31511	2303.12	196.65	14-08-01 to 15-02-28
CN-Part1	27019	1063.70	92.78	14-08-01 to 14-11-30
CN-Part2	30853	1441.28	117.65	14-12-01 to 15-02-28
CB-Sparse	888	101.11	29.09	15-03-01 to 15-05-31
CB-Dense	4323	1012.71	170.07	15-03-01 to 15-05-31
CB-Part1	3014	512.58	84.02	15-03-01 to 15-04-15
CB-Part2	3770	714.07	120.37	15-04-16 to 15-05-31
UT-Sparse	1992	101.61	27.06	15-03-01 to 15-05-31
UT-Dense	14115	776.04	166.52	15-03-01 to 15-05-31
UT-Part1	9695	374.66	79.53	15-03-01 to 15-04-15
UT-Part2	12497	544.49	116.10	15-04-16 to 15-05-31

TABLE II. DEFAULT PARAMETER SETTING

variable	value	source
N	2 (with side lengths $= \{20m, 200m\}$)	Algorithm 2
r_{20}, r_{200}	$0.625, 0.375$	Algorithm 2
m_c	2000	Algorithm 2
Q	20	Algorithm 2
η	16	Equ. (1)
γ	0.2	Equ. (1)
α	0.4	Equ. (2)
λ	50	Equ. (3)
μ	$1/4000$	Equ. (3)
β_1, β_2	$0.8, 0.2$	Equ (3)

TABLE III. HITTING RATE OF EACH EXPERIMENT

Experiment	$Q = 3$	$Q = 20$	$Q = 50$
CN-Sparse vs. CN-Dense	96.17%	97.62%	97.62%
CN-Part1 vs. CN-Part2	98.0%	98.82%	98.74%
CB-Sparse vs. CB-Dense	90.50%	99.66%	99.66%
CB-Part1 vs. CB-Part2	83.14%	91.61%	93.50%
UT-Sparse vs. UT-Dense	85.14%	99.10%	99.10%
UT-Part1 vs. UT-Part2	78.01%	91.88%	94.22%

2) Effects of N: In the multi-resolution filtering stage, we partition the map into N grids with different granularities (Algorithm 2). To illustrate the effects of N, we use 3 experiments with different parameter settings. In Setting A, we only have one grid. The side length of each cell is 20m. In Setting B, we use the default setting as shown in Table II. In Setting C, we partition the map into 4 grids. The side lengths equal to 20m, 50m, 100m, 200m respectively and we set the weight parameters β as $\{0.6, 0.4, 0.2, 0.1\}$ from the finest granularity to the coarsest granularity. The results are shown in Table IV. Each item in Table IV corresponds to a pair *(coverage, accuracy)*.

From the experimental results, it is easy to see that with multiple granularities, our algorithm achieves a much better performance. The reason is that the finer granularity only captures the significant co-occurrences of two trajectories, such as the co-occurrences in the same building. However, due to the noise of the mobility data and the location error of GPS devices, it is hard to identify the users with such limited information. As we can see, in Setting A, the algorithm rejects the most of the identifications for all of the experiments. Especially, for the experiment CN-Sparse vs. CN-Dense, the coverage is only 28.92%. However, by utilizing multiple granularities with proper weight parameters, we successfully identify 92.01% of the trajectories with the accuracy of 99.74% for the experiment CN-Sparse vs. CN-Dense in Setting C, which is much better than the other settings.

3) Effects of m_c: As we explained in Section IV, the cells with a large population usually correspond to some "common places". Eliminating such cells (more than m_c users being observed in this cell) on one hand accelerates the algorithm, on the other hand enhances the performance of the algorithm.

We evaluate the effects of m_c on the hardest experiment UT-Part1 vs. UT-Part2 (the other experiments are not sensitive to m_c since the trajectories are distributed on a vast spatial domain). We compare the coverage, the corresponding accuracy and the running time under different values of m_c. The results are shown in Table. V.

From Table V we can see that, the coverage increases with the increase of m_c. If m_c is too small (e.g. $m_c = 50$ in our experiment), the algorithm drops the most of the cells and only retain the sparsely-populated cells. Of course, observing the co-occurrences in those sparsely-populated cells is very significant for user identification. However, the coverage in such case is extremely low. Furthermore, we find that the accuracy of each experiment does not vary much when m_c varies. Such property echoes that AUI determines the coverage automatically according to the cohesion of the data set.

It is notable that the complexity of the reduce stage in the first phase of the multi-resolution filtering is $O(\min\{m_c, |T_c|\}^2)$ where T_c is the trajectories ids occurred in the current cell.

TABLE IV. EFFECTS OF N

Experiment	Setting A (20m)	Setting B (20m, 200m)	Setting C (20m, 50m, 100m, 200m)
CN-Sparse vs. CN-Dense	$(28.92, 100.00)$	$(59.72, 99.94)$	$(92.01, 99.74)$
CN-Part1 vs. CN-Part2	$(59.46, 99.95)$	$(88.35, 99.80)$	$(92.01, 99.78)$
CB-Sparse vs. CB-Dense	$(52.70, 99.78)$	$(73.31, 97.39)$	$(81.98, 95.60)$
CB-Part1 vs. CB-Part2	$(57.84, 89.32)$	$(70.66, 91.36)$	$(70.22, 92.01)$
UT-Sparse vs. UT-Dense	$(56.32, 97.77)$	$(71.18, 90.20)$	$(62.60, 94.47)$
UT-Part1 vs. UT-Part2	$(49.88, 88.25)$	$(60.81, 90.09)$	$(58.40, 92.51)$

TABLE V. EFFECTS OF m_c

m_c	coverage (%)	accuracy (%)	time (s)
50	47.08	88.45	33
200	56.48	88.27	66
800	57.75	88.19	164
2000	60.81	90.09	521

Thus, a large value of m_c leads to a high time complexity as well.

4) Effects of α: As we presented in Section V-A, in the signal based similarity, each kernel cell emits a stimulus signal which spreads out spatially with an attenuation factor α. Since α is only related with the signal based similarity, to show the effects of α, we evaluate the algorithm performance by only utilizing the signal based similarity under the coverage equal to 90%. We set α to be 0 and $1 - 10^{-30}$ respectively. Such two values correspond to that the stimulus signal does not spread out at all and the stimulus signal does not decrease almost respectively. The results are shown in Table VII.

Such results essentially reflect the effects of the "kernel cells". When $\alpha = 0$, the cells are independent (we do not distinguish the kernel cells in such case). As we can see by taking the relations of the cells into consideration (setting a proper $\alpha > 0$), the accuracy increases a lot in the hard experiments (UT-Sparse vs. UT-Dense, UT-Part1 vs. UT-Part2).

5) Effects of Distance Values between Kernel Cells: As we mentioned in Section V, the distance values between the kernel cells play an important role in the signal based similarity. To illustrate the effects of taking the distance into consideration, we evaluate our algorithm with 3 experiments with different settings. Recall that the signal based similarity in Equ.(3) is calculated by

$$\mathsf{sig}_i = \mathsf{st}(c_1) + \sum_{k \in K \setminus \{1\}} \mathsf{st}(c_k) \cdot (1 + f_d(\mathsf{md}(c_k))).$$

In Setting A, we set $f_d(x) = 0$, i.e., the signal based similarity is unrelated with the distance values between the cells. We use the default setting as in Table II as Setting B. In Setting C, we replace the sigmoid function with a linear function $f_d(x) = x$ to eliminate the diminishing marginal utility property. Again, we compare the accuracies under the coverage equal to 90%. The results are shown in Table VII

TABLE VI. EFFECTS OF α

Experiment	accuracy(%) $\alpha = 0$	accuracy(%) $\alpha = 0.4$	accuracy(%) $\alpha \approx 1$
CN-Sparse vs. CN-Dense	98.66	98.64	69.43
CN-Part1 vs. CN-Part2	99.58	99.59	98.68
CB-Sparse vs. CB-Dense	94.25	94.37	76.50
CB-Part1 vs. CB-Part2	79.63	80.08	73.00
UT-Sparse vs. UT-Dense	87.06	93.15	69.47
UT-Part1 vs. UT-Part2	69.18	72.34	47.50

TABLE VII. EFFECTS OF DISTANCES

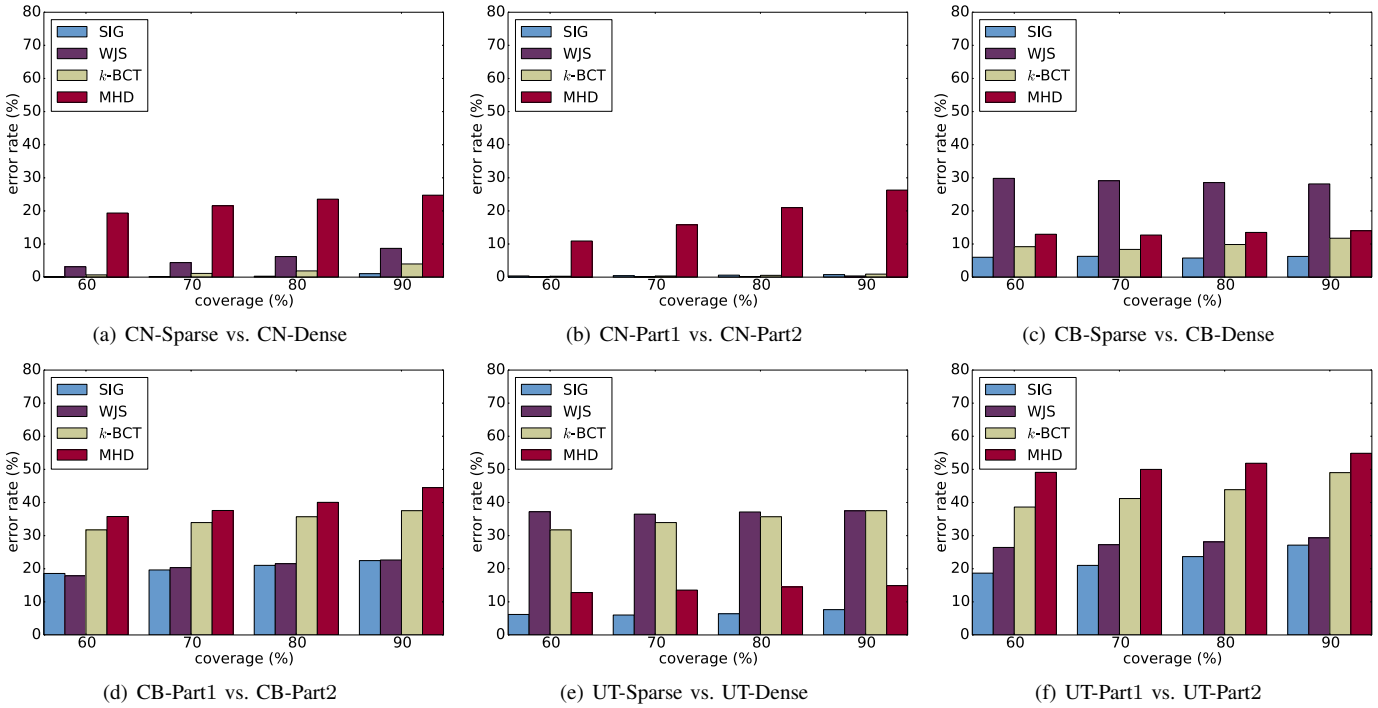
Experiment	Setting A	Setting B	Setting C
CN-Sparse vs. CN-Dense	98.32	98.64	98.52
CN-Part1 vs. CN-Part2	99.48	99.59	98.86
CB-Sparse vs. CB-Dense	87.23	94.37	86.36
CB-Part1 vs. CB-Part2	79.94	80.08	63.05
UT-Sparse vs. UT-Dense	76.00	93.15	86.14
UT-Part1 vs. UT-Part2	68.53	72.34	57.79

From Table VII, we can see that for the two easiest experiments, the distances between the cells do not effect on the accuracy much. However, for the hardest two experiments, the accuracy increases dramatically when the distance values and the diminishing marginal utility property is taking into consideration.

C. Comparisons with Other Algorithms

In this section, we use the default setting as shown in Section VI-B and compare our algorithm with the other existing methods.

We first compare the performance of the signal based similarity with the existing measures. As we mentioned in Section I, since our trajectories we deal with are sampled at very different rate, the sparse trajectories may contain less than one point per day. Even the time span of the trajectories can



(a) CN-Sparse vs. CN-Dense (b) CN-Part1 vs. CN-Part2 (c) CB-Sparse vs. CB-Dense

(d) CB-Part1 vs. CB-Part2 (e) UT-Sparse vs. UT-Dense (f) UT-Part1 vs. UT-Part2

Fig. 6. Error rates of different similarity measures under the same coverage

TABLE VIII. PERFORMANCE COMPARISON FOR AUI

Experiment	AUI	SIG	WJS	k-BCT	MHD
CN-Sparse vs. CN-Dense (coverage = 59.72%)	99.94%	99.84%	99.08%	98.80%	78.13%
CN-Part1 vs. CN-Part2 (coverage = 88.35%)	99.80%	99.57%	99.41%	98.20%	70.39%
CB-Sparse vs. CB-Dense (coverage = 73.31%)	97.39%	97.53%	70.87%	91.59%	87.37%
CB-Part1 vs. CB-Part2 (coverage = 70.66%)	91.36%	81.67%	80.43%	65.93%	62.40%
UT-Sparse vs. UT-Dense (coverage = 72.84%)	95.63%	95.20%	76.02%	71.48%	87.32%
UT-Part1 vs. UT-Part2 (coverage = 60.81%)	90.09%	80.96%	74.97%	61.38%	50.89%

be disjoint. Most prior work is not available for our problem setting. Here we compare with two feasible existing similarity measures. [2]

Adelfio et al. [38] extended Hausdorff distance and proposed the *modified Hausdorff distance* (MHD). For two trajectories T_A and T_B, MHD is defined as:

$$\text{MHD}(T_A, T_B) = \frac{1}{|T_A|} \sum_{p_a \in T_A} \min_{p_b \in T_B} \text{Dis}(p_a, p_b)$$

Chen et al. [35] present a similarity measure called k *Best Connected Trajectories* (k-BCT). Specifically, for two trajectories T_A and T_B, k-BCT is defined as:

$$k\text{-BCT}(T_A, T_B) = \sum_{p_a \in T_A} e^{-\min_{p_b \in T_B} \text{Dis}(p_a, p_b)}.$$

To make a fair comparison, we adjust the threshold value for each measure and compare the error rates under the same coverage. The results are shown in Figure 6.

[2]These methods are available but not designed for our setting.

We then show the performance evaluation of AUI. We set the thresholds $\theta_{\text{SIG}} = 12$ and an extremely small threshold $\theta_{\text{WJS}} = 0.005$. In Table VIII, we label the coverage of AUI and compare the performance for the other similarity measures under the same coverage.

D. Discussions

Based on the experiment results, we can easily see the advantages of the signal based similarity and AUI in our problem.

For all the 6 experiments, the signal based similarity outperforms other measures significantly. As we can see, the weighted Jaccard similarity performs well when the trajectories are sampled at similar rates (Figure 6(b), Figure 6(d), Figure 6(f)). However, it causes high error rate when the trajectories are sampled at very different rates (Figure 6(a), Figure 6(c), Figure 6(e)). The reason is that the weighted Jaccard similarity essentially measures the similarity of the spatio-temporal point sets. When the trajectories are sample at very different rates, the point sets of the same user can become totally distinct. On the other hand, MHD and k-BCT capture the geometrical distance features between the trajectories. These two measures

are not too sensitive to the sample rates. Nevertheless, when the trajectories have significant overlaps, it is hard to identify the users by only using the distance features (Figure 6(d), Figure 6(e), Figure 6(f)). Furthermore, we stress that for the trajectories distributed on a vast spatial domain, there usually exist several points which are far from each other (e.g., the spatio-temporal points in city B and city C in Figure 1(c)). For such case, the modified Hausdorff distance can be very large which is easy to lead to mistakes (Figure 6(a) and Figure 6(b)).

Despite the signal based similarity out performs the other measures for all the experiments. For real applications, it is hard to assign a proper threshold value or determine the coverage in advance. For example, by setting $\theta_{\mathrm{SIG}} = 0.95$, it is enough to achieve an accuracy of 97.90% under the coverage of 90% for the experiment CN-Sparse vs. CN-Dense. However, for the experiment UT-Part1 vs. UT-Part2, we need to set the threshold θ_{SIG} equal to 10.95 to achieve the same coverage and the accuracy is only 71.40%. AUI could determine the coverage according to the cohesion of the data set. As show in Table 4, for the easiest case (CN-Part 1 vs. CN-Part 2), AUI achieves the coverage of 88.35% and the corresponding accuracy of 99.80%. For the hardest cases (UT-Sparse vs. UT-Dense and UT-Part1 vs. UT-Part2), AUI covers only 72.84% and 60.81% of the trajectories but much higher accuracies (95.63% and 90.09%) than the other measures under the same coverage.

VII. Related Work

The studies of user identification focus on the concept of uniqueness in human mobility. It has been shown that people tend to visit places where they visited regularly in the past, which we refer to as "significant places" [13], [23]–[25]. Zang and Bolot [25] showed that given each individual's top n significant places with highest visiting frequencies, we can uniquely identify a small subset of users from a very large-scale anonymized dataset. Montjoye et al. [23] showed that the human mobility retain highly unique even if we coarsen the location points. We stress that these work investigated the uniqueness or the user identification problem in a single mobility data set, i.e., given several historical location points which are already contained in the data set and retrieve the trajectories that match the given points. Rossi et al. [36] presented a technique for identifying users with previously unseen data, i.e., the location points that are not included in the original data set used for model training. We point out that in [36], the unseen data in their experiment is *sampled without replacement* from the original data set which differs from our problem. As we showed in Section VI, their method does not handle the user identification across heterogeneous data sources effectively. Furthermore, in a related work, Crandall et al. [13] used sparse geo-tags in social platforms to infer the social ties between different users. They showed that the strength of social ties is highly correlated with geographical co-occurrences of the users under various spatial granularities.

The user similarity search problem aims to retrieve similar user items from the entire database, based on the personal traits extracted from their past behaviors, in our case, their spatio-temporal mobility patterns. As we claimed in Section I, user identification can be regarded as a special case of user similarity search. An essential ingredient in most methods is to define the similarity measure between a pair of user trajectories. Considerable amount of definitions have been proposed in the past. Most of them are extensions or variants of traditional methods, including Dynamic Time Warping (DTW) [32], Longest Common Subsequence (LCSS) [26], Edit Distance on Real Penalty (ERP) [31], Edit Distance on Real Sequences (EDR) [30], etc. In addition, Li et at. [27] proposed a hierarchical graph based similarity measurement (HGSM) which takes into account both sequential and hierarchical property of geographic locations in user trajectories. These measures utilized the temporal closeness of trajectories, i.e., two trajectories are similar if they co-occurred in approximately the same place at approximately the time. Chen et al. [35] defined a trajectory similarity measure called k *Best Connected Trajectories* (k-BCT) based on the spatial distance and the order constraint in trajectory. Since k-BCT aims to search trajectories from a database using a small set of locations as queries, directly deploying it could lead to high error rate when the trajectories have a significant overlap as we showed in Section VI. We refer the interested readers to a recent survey [39] for more details about user similarity search. In the recent research [33], Ranu et al. studied the similarity measure of trajectories under inconsistent sampling rates. They formulated a robust distance function called *Edit Distance with Projections (EDwP)* to match trajectories under inconsistent and variable sampling rates. However, their method needs to infer the movement of users such as the speed which is not available in our problem.

VIII. Conclusion

In this paper, we study the method of identifying users from heterogeneous data sources. In our problem, the trajectories in mobility data that we deal with are sampled at very different rates and extremely noisy. To address this challenge, we formulate the user identification problem over large scale heterogeneous mobility data sets and present a MapReduce-based framework called *Automatic User Identification* (AUI). AUI is based on a novel similarity measure called the *signal based similarity*. We conduct extensive experiments and show that the signal based similarity significantly outperforms the existing similarity measures for user identification problem. Furthermore, we adopt a rejection strategy to reduce misidentification when the application scenarios are sensitive to error cases. The experimental results show that our rejection strategy can further improve the accuracy of our framework.

Acknowledgement

This work was supported in part by the National Basic Research Program of China grants 2015CB358700, 2011CBA00300, 2011CBA00301, and the National NSFC grants 61033001, 61361136003.

References

[1] B. Krogh, O. Andersen, E. Lewis-Kelham, N. Pelekis, Y. Theodoridis, and K. Torp, "Trajectory based traffic analysis," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2013, pp. 536–539.

[2] P. H. Li, M. L. Yiu, and K. Mouratidis, "Historical traffic-tolerant paths in road networks." ACM Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL), 2014.

[3] I. Hefez, Y. Kanza, and R. Levin, "Tarsius: A system for traffic-aware route search under conditions of uncertainty," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 517–520.

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[5] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, and P. Kalnis, "User oriented trajectory search for trip recommendation," in *Proceedings of the 15th International Conference on Extending Database Technology*. ACM, 2012, pp. 156–167.

[6] J.-D. Zhang, C.-Y. Chow, and Y. Li, "Lore: Exploiting sequential influence for location recommendations," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014, pp. 103–112.

[7] E. H.-C. Lu, C.-Y. Chen, and V. S. Tseng, "Personalized trip recommendation with multiple constraints by mining user check-in behaviors," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 209–218.

[8] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou, "Crowdplanner: A crowd-based route recommendation system," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 1144–1155.

[9] Z. Chen, H. T. Shen, and X. Zhou, "Discovering popular routes from trajectories," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 2011, pp. 900–911.

[10] H. Su, K. Zheng, K. Zeng, J. Huang, S. Sadiq, N. J. Yuan, and X. Zhou, "Making sense of trajectory data: A partition-and-summarization approach," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 963–974.

[11] J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized route recommendation using big trajectory data," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 543–554.

[12] E. Spertus, M. Sahami, and O. Buyukkokten, "Evaluating similarity measures: a large-scale study in the orkut social network," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 678–684.

[13] D. J. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg, "Inferring social ties from geographic coincidences," *Proceedings of the National Academy of Sciences*, vol. 107, no. 52, pp. 22 436–22 441, 2010.

[14] P. Bouros, D. Sacharidis, and N. Bikakis, "Regionally influential users in location-aware social networks," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014, pp. 501–504.

[15] Y. Kanza, E. Kravi, and U. Motchan, "City nexus: discovering pairs of jointly-visited locations based on geo-tagged posts in social networks," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014, pp. 597–600.

[16] W. Kang, A. K. Tung, F. Zhao, and X. Li, "Interactive hierarchical tag clouds for summarizing spatiotemporal social contents," in *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*. IEEE, 2014, pp. 868–879.

[17] J. Jiang, H. Lu, B. Yang, and B. Cui, "Finding top-k local users in geo-tagged social media data." ICDE, 2015.

[18] L. Chen, G. Cong, X. Cao, and K.-L. Tan, "Temporal spatial-keyword top-k publish/subscribe," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 255–266.

[19] B. Zheng, N. J. Yuan, K. Zheng, X. Xie, S. Sadiq, and X. Zhou, "Approximate keyword search in semantic trajectory database," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 975–986.

[20] R. Karam, F. Favetta, R. Kilany, and R. Laurini, "Integration of similar location based services proposed by several providers," in *Networked Digital Technologies*. Springer, 2010, pp. 136–144.

[21] R. Karam and M. Melchiori, "Improving geo-spatial linked data with the wisdom of the crowds," in *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. ACM, 2013, pp. 68–74.

[22] S. Vert and R. Vasiu, "Relevant aspects for the integration of linked data in mobile augmented reality applications for tourism," in *Information and Software Technologies*. Springer, 2014, pp. 334–345.

[23] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific reports*, vol. 3, 2013.

[24] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[25] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 2011, pp. 145–156.

[26] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Data Engineering, 2002. Proceedings. 18th International Conference on*. IEEE, 2002, pp. 673–684.

[27] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. ACM, 2008, p. 34.

[28] H. Wang and K. Liu, "User oriented trajectory similarity search," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*. ACM, 2012, pp. 103–110.

[29] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Mining user similarity from semantic trajectories," in *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*. ACM, 2010, pp. 19–26.

[30] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 491–502.

[31] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 792–803.

[32] B.-K. Yi, H. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Data Engineering, 1998. Proceedings., 14th International Conference on*. IEEE, 1998, pp. 201–208.

[33] S. Ranu, P. Deepak, A. D. Telang, P. Deshpande, and S. Raghavan, "Indexing and matching trajectories under inconsistent sampling rates," in *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE, 2015, pp. 999–1010.

[34] E. Frentzos, K. Gratsias, and Y. Theodoridis, "Index-based most similar trajectory search," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007, pp. 816–825.

[35] Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie, "Searching trajectories by locations: an efficiency study," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 255–266.

[36] L. Rossi and M. Musolesi, "Spatio-temporal techniques for user identification by means of gps mobility data," *EPJ Data Science*, vol. 4, pp. 1–16, 2015.

[37] S. Ioffe, "Improved consistent sampling, weighted minhash and l1 sketching," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 246–255.

[38] M. D. Adelfio, S. Nutanong, and H. Samet, "Similarity search on a large collection of point sets," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 132–141.

[39] Y. Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, p. 29, 2015.