A Succinct Model for Re-identification of Mobility Traces Based on Small Training Data

Takao Murakami
National Institute of Advanced Industrial Science and Technology (AIST)
Tokyo, Japan
takao-murakami@aist.go.jp

Abstract—Re-identification of mobility traces based on the Markov chain model has been widely studied to understand the risk of location privacy. It is well known that this model can re-identify the traces with very high accuracy when the amount of training data is large. However, the amount of training data can be very small in practice, since a user generally discloses only a small number of locations to the public. A state-of-the-art method in this scenario is to train the Markov chain model (transition matrices) via tensor factorization. The previous work has shown that this method outperforms a random guess even when the amount of training data is very small.

In this paper, we propose a succinct model for re-identification that outperforms the state-of-the-art method explained above. Our proposed method does not model a transition pattern (unlike the Markov chain model) but models a probability of being located in each region via matrix factorization. Then it re-identifies traces based on the JS (Jensen-Shannon) divergence between two probability distributions. We evaluate the proposed method using the Gowalla dataset, and demonstrate that the proposed method significantly outperforms the tensor factorization-based Markov chain model. We also demonstrate that the proposed method significantly outperforms a random guess even when only one single location is available per user as training data.

I. INTRODUCTION

With the widespread use of location-based services (LBS) such as location check-in, tagging, and POI (Point-of-interest) search, a great number of locations or mobility traces (time-series location data) have been collected into a database of the LBS provider. Although these data can be provided to a third party for Big data analysis [1] or made public to provide traffic information, there are some serious privacy issues. For example, the mobility traces may contain sensitive information such as homes and hospitals. There is also a security risk that a stalker exploits a victim’s location information [2]. Therefore, the LBS provider generally pseudonymize the mobility traces (i.e., replace user IDs with pseudonyms) before providing them to the third party (or publishing them). However, it is also argued that only the pseudonymization is not sufficient to anonymize the traces. A number of re-identification (a.k.a. de-anonymization) attacks [3]–[11] have been proposed in the literature to link the pseudonymized traces to the users.

A representative approach to re-identifying mobility traces is based on the Markov chain model [3]–[9] (other approaches include re-identification using home/workplace location pairs [10] and social network graph [11]). This approach first divides an area of interest into \( m \) regions \( x_1, \ldots, x_m \), and partitions time at a fixed interval (e.g., 30 minutes, one hour). Then it trains, for each of \( n \) target users \( u_1, \ldots, u_n \), an \( m \times m \) transition matrix \( Q_i \) (\( 1 \leq i \leq n \)), whose \((j,k)\)-th element is the transition probability of moving from region \( x_j \) to \( x_k \). Using the transition matrices \( Q_1, \ldots, Q_n \), it re-identifies each pseudonymized trace. Fig. 1 shows the overview of re-identification attacks based on the Markov chain model (we describe this model in Section II-B in more detail). It was shown that this model re-identifies users with high accuracy (e.g., 50% or more) when the amount of training data is very large [4], [5].

However, the amount of training data can be very small in practice, since a user generally discloses only a small number of locations to the public (e.g., by location check-in or tagging). To address this issue, a recent study [12] assumed a set of transition matrices as a third-order tensor, and proposed a method to train transition matrices via tensor factorization [13]. In this paper, we refer to the Markov chain model trained via tensor factorization as a tensor-factorization based Markov chain model. It was shown in [12] that this model outperforms a random guess in localization attacks, which specify an actual location of a user at a certain time, when the amount of training data is very small. It was also shown in [7]–[9] that this model outperforms a random guess in re-identification attacks.

In this paper, we reconsider a suitable model for re-identification from a small amount of training data. Generally, the size of the model should be small to avoid the over-fitting problem when the amount of training data is small [14]. In addition, consider an extreme case where only one single location is available per user as training data. The Markov chain model cannot be trained in this case, since no transition pattern is observed in the data. Therefore, we propose a new succinct model, which does not capture a transition pattern but captures a probability of being located in each region.

Our contributions are summarized as follows:

- We propose a succinct model for re-identification attacks. Our proposed method models a probability of being located in each region via matrix factorization, and re-identifies traces based on the JS (Jensen-Shannon) divergence between two probability distributions (Section III).

- We evaluate the proposed method using the Gowalla dataset [15]. The results showed that the proposed method...
significantly outperforms the tensor-factorization based Markov chain model [12], and significantly outperforms a random guess even when only one location is available per user as training data (Section IV).

II. PRELIMINARIES

In this section, we describe the previous work related to our work. We first describe the basic notations in Section II-A. We then explain re-identification attacks based on the Markov chain model and tensor factorization-based Markov chain model [12] in Sections II-B and II-C, respectively.

A. Notations

Let $\mathbb{R}$ be the set of real numbers and $\mathbb{R}_+$ be the set of non-negative real numbers. In addition, let $\mathbb{Z}_+$ be the set of non-negative integers and $\mathbb{N}$ be the set of natural numbers; i.e., $\mathbb{Z}_+ = \{0, 1, 2, \cdots \}$, $\mathbb{N} = \{1, 2, \cdots \}$.

In this paper, we consider $n \in \mathbb{N}$ target users and $m \in \mathbb{N}$ regions. Let $\mathcal{U} = \{u_1, \cdots, u_n\}$ be a set of target users, and $\mathcal{X} = \{x_1, \cdots, x_m\}$ be a set of regions. We model a discrete time as a positive integer; i.e., the set of times is $\mathbb{N}$.

B. Re-identification attacks based on the Markov chain model

Fig. 1 shows the framework for re-identification attacks based on the Markov chain model (which is based on the location-privacy framework introduced by [3]). We assume that anyone who obtains pseudonymized traces can be an adversary. However, we except for the LBS provider, since she has the original traces. For example, if the LBS provider provides the pseudonymized traces to a third party for analysis, the third party can be an adversary.

In the training phase, the adversary trains a transition matrix $Q_i \in [0, 1]^{m \times m}$ for each user $u_i \in \mathcal{U}$ ($1 \leq i \leq n$). Note that the amount of training data can be very small, since these data need to be linked to a user’s identity. Examples of such data include location data made public via location check-in or tagging, as described in Section I. In the re-identification phase, the LBS provider pseudonymizes mobility traces. Here we assume that the LBS provider does not obfuscate traces (e.g., adds noise to locations, generalizes locations) for simplicity. Then the adversary re-identifies each pseudonymized trace (i.e., determines whether it has been generated by $u_1, u_2, \cdots, u_n$) using transition matrices $Q_1, \cdots, Q_n$.

In this paper, we consider the Bayesian re-identification attack [7] as a concrete method to re-identify traces using the transition matrices $Q_1, \cdots, Q_n$. Let $l \in \mathbb{N}$ be the length of a pseudonymized trace and $X_t$ ($1 \leq t \leq l$) be a random variable representing a region of the pseudonymized trace at time $t$. We denote the pseudonymized trace by $X := (X_1, X_2, \cdots, X_l)$. Furthermore, let $H_i$ be a hypothesis that the pseudonymized trace has been generated by user $u_i$.

Given the pseudonymized trace $X$, the Bayesian re-identification attack computes a posterior probability $\Pr(H_i|X)$ for each hypothesis ($1 \leq i \leq n$) using the transition matrices $Q_1, \cdots, Q_n$ (for details of computing $\Pr(H_i|X)$, see [7]). Then it chooses user $u_i$ whose posterior probability $\Pr(H_i|X)$ is the largest as an identification result. More generally, it chooses $n'$ ($1 \leq n' \leq n$) candidates by sorting $n$ users $u_1, \cdots, u_n$ in descending order of $\Pr(H_i|X)$ and choosing the top $n'$ users.

The feature of the Bayesian re-identification attack is that it maximizes the re-identification accuracy (i.e., the probability that the correct answer is included in the $n'$ candidates) if the transition matrices $Q_1, \cdots, Q_n$ are accurately estimated.

C. Tensor factorization-based Markov chain model [12]

The Bayesian re-identification attack is optimal if the transition matrices $Q_1, \cdots, Q_n$ are accurately estimated. However, it is very challenging to estimate $Q_1, \cdots, Q_n$ in practice, since a user generally discloses only a small number of locations to the public in her daily life (e.g., uses location check-ins one or two times per day).

Fig. 1 shows an example of the case where the ML (Maximum Likelihood) estimation method is used to train $Q_1, \cdots, Q_n$. Here, “?” represents an unobserved element, which cannot be estimated from the ML estimation method. For example, since there is no transition from region $x_1$, $x_4$, and $x_5$ in the training trace of user $u_1$, the transition probabilities from $x_1$, $x_4$, and $x_5$ cannot be estimated using the ML estimation method. In addition, the remaining elements are obviously overfitted to the training trace (e.g., 0, 1). If $Q_1, \cdots, Q_n$ are not accurate, the adversary cannot accurately re-identify the traces.

To address this issue, a recent study [12] proposed a method to train $Q_1, \cdots, Q_n$ via tensor factorization. Specifically, it assumes $\{Q_1, \cdots, Q_n\}$ as a third-order tensor $A \in [0, 1]^{n \times m \times m}$ whose $(i, j, k)$-th element represents the $(j, k)$-th element of $Q_j$. Then it estimates $A$ from training traces of all users using tensor factorization [13]. Tensor factorization is a method to decompose a third-order tensor into low-rank matrices, and is known as a kind of dimensionality reduction method. By using tensor factorization, elements in $A$ (including unobserved elements) can be estimated from a small amount of training data with the help of other users. For more details, see [12].

It is shown in [7]–[9] that this method outperforms the ML estimation method and a random guess in re-identification attacks when the amount of training data is very small.
we compute a probability distribution \( p_i \) for each user \( u_i \) by normalizing each row in the substituted matrix.

In the re-identification phase (Fig. 3), we compute an empirical distribution \( p' \) from a pseudonymized trace \( (p'(x_j)) \) is an empirical probability of being located in region \( x_j \). Then we compute a distance \( d_i \in \mathbb{R}_+ \) between two probability distributions \( p_i \) and \( p' \) for each user \( u_i \) (1 \( \leq i \leq n \)). The distance \( d_i \) measures how “unlikely” the pseudonymized trace is generated from user \( u_i \) (i.e., the smaller \( d_i \) is, the more likely the pseudonymized trace is from \( u_i \)). As a distance measure between two probability distributions, we use the JS (Jensen-Shannon) divergence [17] (we describe the JS divergence in Section III-B in detail). Finally, we choose user \( u_i \) whose JS divergence is the smallest as an identification result. More generally, we choose \( n' \) (1 \( \leq n' \leq n \)) candidates by sorting \( n \) users \( u_1, \ldots, u_n \) in ascending order of the JS divergence and choosing the top \( n' \) users.

The feature of the proposed method lies at its simple model. In general, the size of the model should be small to avoid the over-fitting problem when the size of the training data is small [14]. In the Markov chain model, the number of elements in the transition matrices is \( m^2n \) in total. On the other hand, the size of \( p_1, \ldots, p_n \) in the proposed model is only \( mn \). The size of parameters in tensor factorization is \( (4m + 2n)k \) in [12], whereas the size of \( W \) and \( H \) in the proposed method is only \( (m + n)k \). In our experiments, we show that when there is only one training trace composed of ten locations per user (which is the same scenario as [12]), the proposed method outperforms the tensor factorization-based Markov chain model [12], which is the state-of-the-art re-identification model.

In addition, when only one single location is available per user as training data, the Markov chain model cannot be trained (as described in Section I). On the other hand, the proposed method works even in this case, since it does not model a transition pattern but models the probability of being located in each region. In our experiments, we show that the proposed model re-identifies a single location with much higher accuracy than a random guess.

### B. Algorithm

We now describe the proposed method in detail. Specifically, we describe details of NMF (Non-negative Matrix Factorization) and the JS (Jensen-Shannon) divergence in the proposed method. Then we summarize the proposed algorithm.

**Fig. 2. Proposed method in the training phase.**

**III. A SUCINCT MODEL FOR RE-IDENTIFICATION ATTACKS**

We propose a succinct model for re-identification attacks that significantly outperforms the tensor-factorization based Markov chain model [12], and works well even in an extreme case where only one single location is available per user as training data. We first describe its overview in Section III-A. We then explain the proposed method in detail in Section III-B.

**A. Overview**

Figs. 2 and 3 show the overview of the proposed method. In the training phase (Fig. 2), we first compute a user-region matrix \( A \in \mathbb{Z}_{+}^{m \times n} \) from training traces. The \((i, j)\)-th element \( A_{i,j} \) of \( A \) is the number of times user \( u_i \) is in region \( x_j \). For example, \( A_{1,2} = 2 \) in Fig. 2, since user \( u_1 \) is in region \( x_2 \) at times 1 and 2. Based on \( A \), we compute a probability \( p_i(x_j) \in [0, 1] \) that user \( u_i \) is located in region \( x_j \) (1 \( \leq i \leq n, 1 \leq j \leq m \)). In other words, we compute a probability distribution \( p_i \) for each \( u_i \) (1 \( \leq i \leq n \)). The probability distribution \( p_i \) is used for re-identifying traces.

The simplest approach to computing \( p_i \) (1 \( \leq i \leq n \)) from \( A \) would be to use the ML (Maximum Likelihood) estimation method, which normalizes each row in \( A \) so that the summation of each row is one. The major drawback of this approach is that if an element in \( A \) is zero, the corresponding probability in \( p_i \) also becomes zero (i.e., overfitting problem). Consequently, if a pseudonymized trace has a number of locations not appeared in the training trace, the adversary may not accurately re-identify the pseudonymized trace using \( p_i \).

To overcome this problem, we use matrix factorization [16], which has been widely used for predicting missing ratings in item recommendation. Specifically, we regard zero elements in \( A \) as missing elements, and estimate these elements using matrix factorization. We factorize the user-region matrix \( A \) into two low-rank matrices \( W \in \mathbb{R}^{n \times k} \) and \( H \in \mathbb{R}^{k \times m} \) using NMF (Non-negative Matrix Factorization) [13] (we describe an algorithm for NMF in Section III-B in detail). Then we compute \( \hat{A} \in \mathbb{R}^{n \times m} \), which is an approximation of \( A \), by multiplying \( W \) by \( H^\top \); i.e., \( A = WH^\top \).

We substitute elements in \( \hat{A} \) for zero elements in \( A \). Finally, we compute a probability distribution \( p_i \) for each user \( u_i \) by normalizing each row in the substituted matrix.

In the re-identification phase (Fig. 3), we compute an empirical distribution \( p' \) from a pseudonymized trace \( (p'(x_j)) \) is an empirical probability of being located in region \( x_j \). Then we compute a distance \( d_i \in \mathbb{R}_+ \) between two probability distributions \( p_i \) and \( p' \) for each user \( u_i \) (1 \( \leq i \leq n \)). The distance \( d_i \) measures how “unlikely” the pseudonymized trace is generated from user \( u_i \) (i.e., the smaller \( d_i \) is, the more likely the pseudonymized trace is from \( u_i \)). As a distance measure between two probability distributions, we use the JS (Jensen-Shannon) divergence [17] (we describe the JS divergence in Section III-B in detail). Finally, we choose user \( u_i \) whose JS divergence is the smallest as an identification result. More generally, we choose \( n' \) (1 \( \leq n' \leq n \)) candidates by sorting \( n \) users \( u_1, \ldots, u_n \) in ascending order of the JS divergence and choosing the top \( n' \) users.

The feature of the proposed method lies at its simple model. In general, the size of the model should be small to avoid the over-fitting problem when the size of the training data is small [14]. In the Markov chain model, the number of elements in the transition matrices is \( m^2n \) in total. On the other hand, the size of \( p_1, \ldots, p_n \) in the proposed model is only \( mn \). The size of parameters in tensor factorization is \( (4m + 2n)k \) in [12], whereas the size of \( W \) and \( H \) in the proposed method is only \( (m + n)k \). In our experiments, we show that when there is only one training trace composed of ten locations per user (which is the same scenario as [12]), the proposed method outperforms the tensor factorization-based Markov chain model [12], which is the state-of-the-art re-identification model.

In addition, when only one single location is available per user as training data, the Markov chain model cannot be trained (as described in Section I). On the other hand, the proposed method works even in this case, since it does not model a transition pattern but models the probability of being located in each region. In our experiments, we show that the proposed model re-identifies a single location with much higher accuracy than a random guess.

**B. Algorithm**

We now describe the proposed method in detail. Specifically, we describe details of NMF (Non-negative Matrix Factorization) and the JS (Jensen-Shannon) divergence in the proposed method. Then we summarize the proposed algorithm.

**Fig. 3. Proposed method in the re-identification phase.**
In the training phase, we compute an approximation \( \hat{\mathbf{A}} \) of \( \mathbf{A} \) by factorizing \( \mathbf{A} \) into \( \mathbf{W} \) and \( \mathbf{H} \) using NMF. Let \( \Theta = \{ \mathbf{W}, \mathbf{H} \} \) be a set of parameters in NMF. NMF generally considers the following optimization problem:

\[
\min_{\Theta \geq 0} \sum_{(i,j) \in \Omega} (A_{i,j} - \hat{A}_{i,j})^2 + \lambda(||W||^2_F + ||H||^2_F),
\]

where \( \Omega \) is a set of all non-zero elements in \( \mathbf{A} \). \( || \cdot ||^2_F \) is the Frobenius norm (i.e., the square sum of all elements), and \( \lambda \geq 0 \) is a regularization parameter. The first term is the summation of squared errors between \( \mathbf{A} \) and \( \hat{\mathbf{A}} \) over \( \Omega \). The second term is called a regularization term, and is introduced to avoid the problem that \( \hat{\mathbf{A}} \) overfits \( \mathbf{A} \). The regularization parameter \( \lambda \) is generally determined by cross-validation [16]. “\( \Theta \geq 0 \)” means that all elements in \( \Theta \) are non-negative.

Since the optimization problem in (1) is not convex, it is hard to find an exact solution to (1) in practice. Therefore, we use ANLS (Alternating Non-negative Least Square) [13], [18] to find an approximate solution. ANLS solves for one parameter in \( \Theta \) while fixing the other parameters, and iterates it in a cyclic manner until convergence. Since the optimization problem in (1) is quadratic with regard to one parameter, it can be easily solved optimally. We do not explain details of the update formulae for lack of space (they are similar to [12]; see [12] for more details).

In the re-identification phase, we compute the JS divergence between two probability distributions \( p_i \) and \( p'_i \) for each user \( u_i \) (1 \( \leq \) \( i \) \( \leq \) \( n \)). The JS divergence is based on the KL (Kullback-Leibler) divergence (also called relative entropy), which is widely used in information theory [19]. The KL divergence \( D(p_i \| p'_i) \) between \( p_i \) and \( p'_i \) is given by:

\[
D(p_i \| p'_i) = \sum_{j=1}^{m} p_i(x_j) \log \frac{p_i(x_j)}{p'_i(x_j)}.
\]

It should be noted that the KL divergence \( D(p_i \| p'_i) \) becomes infinity when \( p_i > 0 \) and \( p'_i(x_j) = 0 \) for some region \( x_j \).

To avoid this, we use the JS divergence. The JS divergence \( \text{JSD}(p_i \| p'_i) \) between \( p_i \) and \( p'_i \) is given by:

\[
\text{JSD}(p_i \| p'_i) = \frac{1}{2} D(p_i \| m_i) + \frac{1}{2} D(p'_i \| m_i),
\]

where \( m_i = (p_i + p'_i)/2 \). The JS divergence \( \text{JSD}(p_i \| p'_i) \) is always finite, unlike the KL divergence.

In summary, the proposed algorithm is as follows:

**Training:**

1) Compute a user×region matrix \( \mathbf{A} \) from training traces.
2) Compute a parameter \( \Theta = \{ \mathbf{W}, \mathbf{H} \} \) in NMF by solving (1) via ANLS, and compute \( \hat{\mathbf{A}} = \mathbf{WH}^\top \).
3) Substitute elements in \( \hat{\mathbf{A}} \) for zero elements in \( \mathbf{A} \).
4) Compute each distribution \( p_i \) (1 \( \leq \) \( i \) \( \leq \) \( n \)) by normalizing each row in the substituted matrix.

**Re-identification:**

1) Compute a distribution \( p'_i \) from a pseudonymized trace.
2) Compute \( \text{JSD}(p_i \| p'_i) \) in (3) (1 \( \leq \) \( i \) \( \leq \) \( n \)).
3) Sort \( n \) users in ascending order of \( \text{JSD}(p_i \| p'_i) \), and choose the top \( n' \) (1 \( \leq \) \( n' \) \( \leq \) \( n \)) users as candidates.

**IV. EXPERIMENTAL EVALUATION**

**A. Experimental Set-up**

We performed experiments to evaluate the proposed method. In our experiments, we used the Gowalla dataset [15], which contains 6442890 check-ins of 196591 users all over the world. In our experiments, we used traces in New York and Philadelphia. We selected 164 users who had long traces (\( n = 164 \)), and extracted one training trace and nine testing traces (pseudonymized traces), each of which is composed of ten locations and has a time interval of greater than 30 minutes. Here we set the time interval between the training trace and the testing traces to be more than 30 days (i.e., the testing traces are generated more than 30 days after the training trace is generated). We eliminated the remaining users, because we had insufficient data to extract such ten traces for these users. We divided New York and Philadelphia into \( 20 \times 20 \) regions with regular intervals (\( m = 400 \)).

Using the training trace, we trained the model for re-identification. Here we compared the following two models:

**TF:** tensor factorization-based Markov chain model [12].

**Proposal:** proposed model (described in Section III).

For both TF and Proposal, we set the column \( k \) of \( \mathbf{W} \) and \( \mathbf{H} \) to be \( k = 16 \), and the regularization parameter \( \lambda \) to be \( \lambda = 10^{-3} \) in the same way as [12] (we also set \( \chi \) to be \( \chi \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} \} \) and confirmed that the performance was almost the same for each case).

Using the testing traces, we evaluated the performance of TF and Proposal. Specifically, we pseudonymized each testing trace, and re-identified the pseudonymized trace based on TF or Proposal. In TF, we chose \( n' \) (1 \( \leq \) \( n' \) \( \leq \) 164) candidates by performing the Bayesian re-identification attack [7] described in Section II-B. In Proposal, we chose \( n' \) candidates by sorting \( n \) users in ascending order of the JS divergence, as described in Section III. As a performance measure, we evaluated the attack success rate, which is the ratio of the number of successful attacks (i.e., attacks in which a correct user is included in the \( n' \) candidates) divided by the total number of testing traces (i.e., 164 \( \times \) 9 = 1476 traces in total).

Although there were ten locations in each training trace as described above, we also performed an experiment where only the first location in each training trace is available to the adversary (i.e., only one single location is available per user as training data). We evaluated only Proposal in this case, because the Markov chain model cannot be trained (as described in Section I). We also evaluated Proposal in the case where only the first location in each of training traces and testing traces is available to the adversary (i.e., the length of both training traces and pseudonymized traces is only one).
Fig. 4. Relationship between the number of candidates $n'$ and the attack success rate ($l_1$: number of locations in each training trace, $l_2$: number of locations in each testing trace, dot line: performance of a random guess).

**B. Experimental Results**

Fig. 4 shows the trade-off between the number of candidates $n'$ and the attack success rate, where $l_1$ and $l_2$ are the number of locations in a training trace and a pseudonymized trace, respectively. The dot line represents the performance of a random guess, which randomly chooses $n'$ candidates from 164 users. It can be seen from Fig. 4 that **Proposal** significantly outperforms TF in the case where $l_1 = l_2 = 10$. We consider this is because the size of the model is smaller in the proposed method, as described in Section III-A. It can also be seen that **Proposal** outperforms a random guess even when $l_1 = 1$, and that the attack success rate of **Proposal** is increased with increase in the length of the testing trace. We emphasize again that no transition pattern is observed in the training trace when $l_1 = 1$. The proposed model significantly outperforms a random guess even in this case, which demonstrates the effectiveness of the new model.

In our experiments, **Proposal** significantly outperforms TF and a random guess. It should be noted, however, that the attack success rate of **Proposal** is still not very high. For example, the attack success rate in the case where $l_1 = 1$, $l_2 = 10$, and $n' = 1$ is about 12%. This means that the adversary can uniquely re-identify traces with probability only about 12%. It can also be seen that the attack success rate in the case where $l_1 = 1$, $l_2 = 10$, $n' = 55$ is about 80%, which means that the adversary can reduce a list of candidates to only one third ($\approx 55/164$) with high accuracy. One way to significantly improve the attack success rate would be to use other auxiliary information (side knowledge). For example, suppose that a user disclosed the fact that she visited some department store (but did not specify which department store) via SNS. By combining this fact with the proposed method, the adversary may further narrow down the candidates.

In addition, the proposed model and the Markov chain model are completely different models. Thus, when $l_1 \geq 2$, the attack success rate might be increased by integrating these two models. As future work, we would like to develop such a fusion model. We would also like to develop a countermeasure against re-identification attacks based on such a model. For example, the combination of geo-indistinguishability and k-anonymity [20] might be an interesting candidate for it.

**V. Conclusion**

In this paper, we proposed a succinct model for re-identification based on small training data. Our experimental results showed that the proposed method significantly outperforms the tensor factorization-based Markov chain model [12], and significantly outperforms a random guess even when the length of a training trace is only one. Future work includes: (1) integrating the proposed model and the Markov chain model; (2) the development of a countermeasure against re-identification attacks using such a model.

**ACKNOWLEDGMENT**

The author would like to thank Yusuke Kawamoto (AIST) for technical comments on this paper.

**References**


