テンソル分解を用いた位置情報プライバシーへの攻撃と対策

村上 隆夫† 渡辺 創†

†産業技術総合研究所セキュアシステム研究部門
305-8568 茨城県つくば市梅園 1-1-1 中央第 2
{takao-murakami, h-watanabe}@aist.go.jp

あらまし 人々の行動に対してマルコフ連鎖を仮定し，個人毎に学習した遷移行列を用いることで，位置情報プライバシーを暴露する様々な攻撃が提案されている．しかし，個人が過去に公開していた位置情報が少ない場合，遷移行列を学習するためのデータが不足する．本稿ではまず，少量の学習データから遷移行列を頑健に推定するため，個人毎の遷移行列の集合を「テンソル」と見做し，テンソル分解を用いて遷移行列を学習する手法を提案する．次に，公開された位置情報を基に，その後の時刻における位置を予測する攻撃を考え，その対策として攻撃成功確率を一定値以下に抑えつつ，曖昧領域サイズを最小化する位置情報の曖昧化手法を提案する．タクシーの移動系列データを用いた評価実験を行い，これらの提案手法の有効性を示す．

Location Privacy Attacks Using Tensor Factorization and Defenses against the Attacks

Takao Murakami† Hajime Watanabe†

†Research Institute for Secure Systems (RISEC),
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568, Japan
{takao-murakami, h-watanabe}@aist.go.jp

Abstract Recent studies have proposed various attacks against location privacy, such as location prediction attacks and tracking attacks, using a Markov Chain transition matrix for each user. However, when a user has disclosed only a small amount of location information in the past, the training data can be extremely sparse. In this paper, we first address this data sparsity problem by regarding a set of transition matrices as a "tensor" and adopting tensor factorization. Then we consider an attack which predicts a location after one disclosed location, and propose a region merging method which can minimize the region size as an optimal defense. The experimental results using the dataset of taxi traces show the effectiveness of our proposals.

1 はじめに

近年，スマートフォンやカーナビゲーションシステムなどが普及し，目的地への経路検索や，周辺の飲食店などの POI（Point of Interest）検索といった，様々な位置情報サービス（LBS: Location-based Service）が広く利用されている．SNS（Social Networking Service）においても，ユーザが自身の位置情報を公開し，周辺にいる知人の位置情報を取得するサービス [1] が注目を集めている．また，ユーザの移動系列情報（位置を時系列に並べたデータ）が大量に

データセンター側に蓄積されているのを背景として，この巨大なデータ（位置情報ビッグデータ）を道路混雑状況の分析 [2] などに利活用することも期待されている．

このようなサービスはユーザに有益な情報をもたらす一方で，ユーザのプライバシーが暴露され得るという問題が指摘されている．例えば，ユーザが公開した位置情報や移動系列情報を基に，自宅や通院している病院などが特定される可能性があり，さらには秘密にしておきたい趣味嗜好や交友関係なども知られてしまう恐れがある．このようなプライバシーは「位置情報プライバシー」（Location Privacy）と呼ばれ，その攻撃，対策，安全性の評価尺度などに関する研究が幅広く行われている [3]．

位置情報プライバシーを暴露する代表的な攻撃として，マルコフ連鎖（Markov Chain）に基づく手法が近年研究されている [4, 5, 6, 7]．この手法では，まず人々が移動可能な領域を計 M 個の領域 $x_1, \cdots, x_M$ に分割することで（或いは人々が良く訪れる計 M 個の POI のみを対象とすることで），位置情報を離散化する．次に，予め時間間隔（10 分，1 時間など）を定めた上で人々の行動にマルコフ性を仮定し，領域 x_i（$1 \leq i \leq M$）から次の時刻に領域 x_j（$1 \leq j \leq M$）に遷移する確率で構成される $M \times M$ の遷移行列を学習する．この遷移行列を用いることで，例えばユーザが公開した位置情報を基に，その後の時刻における位置（領域 ID）を予測する「位置予測攻撃」（Location Prediction Attack）[4, 5] が可能となる．

さらに，文献 [6, 7] は攻撃対象とする個人毎に遷移行列を 1 つずつ学習する手法を提案している．これにより，個人の行動の特徴（歩行速度，良く利用する経路など）を考慮した上で，位置予測攻撃を行うことが可能となる．また，個人 ID の削除や位置情報の曖昧化などの加工を施した上で公開された移動系列情報から，個人と具体的な位置情報を特定する「トラッキング攻撃」（Tracking Attack）[7] も可能となる．この各個人の遷移行列は，その個人が過去に公開した位置情報などを用いて予め学習しておく．

しかしながら，一般的に個人が普段から公開している位置情報は多くない（例えば 1 日に数回など）．文献 [6, 7] では各個人の遷移行列を独立に学習しているが，この場合，各遷移行列の学習データが不足する恐れがある．このデータ不足問題に関する議論は，筆者らの知る限り行われていない．位置情報プライバシーに対する攻撃と対策は表裏一体の関係にあるため [7]，攻撃の議論が十分でないということは，対策の議論も十分でないことを意味している．

以上を考慮し，本研究では個人毎の遷移行列を少量の学習データから推定する手法を提案し，それを用いた攻撃への対策も合わせて提案する．

本研究の貢献：

本研究の貢献は以下のとおりである．

- まず，各個人から得られる少量の学習データから遷移行列を頑健に推定するため，個人毎の遷移行列の集合を「テンソル」と見做し，テンソル分解（Tensor Factorization）[8, 9, 10, 11] を用いて個人毎の遷移行列を学習する手法を提案する（第 3.2 節）．
- 次に，個人毎の遷移行列を用いた具体的な攻撃として位置予測攻撃を考え，その最適な対策として，攻撃成功確率を一定値以下に抑えつつ，曖昧領域サイズを最小化する位置情報の曖昧化手法を提案する（第 4 章）．
- 以上の提案手法に対して，タクシーの移動系列データ（CRAWDAD[12]）を用いた評価実験を行い，有効性を示す（第 5 章）．

2 関連研究

2.1 個人毎の遷移行列を用いた位置情報プライバシーへの攻撃

本章では，個人毎の遷移行列を用いた攻撃に関する文献 [6, 7] とその問題点を詳述する．まず，以降で用いる記号を定義する．攻撃対象とする計 N 人のユーザの集合を $\mathcal{U} = \{u_1, \cdots, u_N\}$ とし，移動可能な計 M 個の領域（或いは POI）の集合を $\mathcal{X} = \{x_1, \cdots, x_M\}$ とする．また，時刻は予め定められた間隔で区切って離散化し，整数値で表すものとする（即ち，時刻の集合は $\mathbb{Z}$）．ユーザ $u_n \in \mathcal{U}$ が領域 $x_i \in \mathcal{X}$ から次の時

刻に領域 $x_j \in \mathcal{X}$ に遷移する確率を $p_{n,i,j}$ とし，ユーザ $u_n \in \mathcal{U}$ の遷移行列を P_n とする．また，N 以下の自然数の集合を $[N]$ と表記する．

文献 [6, 7] は，個人毎の遷移行列 $\{P_n | n \in [N]\}$ を用いた位置情報プライバシーへの攻撃を提案している．具体的には位置予測攻撃やトラッキング攻撃など様々な攻撃が可能だが，ここでは例として位置予測攻撃について詳述する．

位置予測攻撃：

ユーザ $u_n \in \mathcal{U}$ が，時刻 $t \in \mathbb{Z}$ において領域 $x_i \in \mathcal{X}$ にいたことを公開したとする．攻撃者はこれを基に，このユーザがその後の時刻 $t+c \in \mathbb{Z}$（c は自然数）においてどの領域に滞在していたかを予測する．ユーザ $u_n \in \mathcal{U}$ が時刻 $t \in \mathbb{Z}$ において滞在する領域を表す確率変数を $X_{n,t}$ とする．これを用いると，ユーザ u_n の時刻 t における領域が x_i であったときに，時刻 $t+c$ における領域が $x_j \in \mathcal{X}$ であるという事後確率は，

$$\Pr(X_{n,t+c} = x_j | X_{n,t} = x_i) \tag{1}$$

と表せる．式 (1) を簡単に $\alpha_c(x_j)$ と表記すると，これは Forward アルゴリズム [13] より，ユーザ u_n の遷移行列 P_n を用いて，再帰的に

$$\begin{aligned} \alpha_\tau(x_j) &= \Pr(X_{n,t+\tau} = x_j | X_{n,t} = x_i) & (2) \\ &= \sum_{\rho=1}^{M} \alpha_{\tau-1}(x_\rho) p_{n,\rho,j} & (3) \end{aligned}$$

と計算できる．但し，$1 \leq \tau \leq c$ であり，

$$\alpha_0(x_j) = \begin{cases} 1 & (x_j = x_i \text{ のとき}) \\ 0 & (x_j \neq x_i \text{ のとき}) \end{cases} \tag{4}$$

である．

位置予測攻撃では，計 M 個の各領域に対する事後確率 $\{\alpha_c(x_j) | j \in [M]\}$ を求め，ユーザ u_n の時刻 $t+c$ における領域を予測する．例えば，事後確率 $\alpha_c(x_j)$ の最も大きい領域 x_j を予測結果として出力する方法がある．これはベイズ識別 [14] として知られ，事後確率が正しく求まるという仮定の下，攻撃成功確率（即ち，予測結果 x_j が正しい確率）を最大化できる．或いは事後確率 $\alpha_c(x_j)$ の大きい順に M 個の領域をソートし，その上位 $L(\leq M)$ 個を候補として出力する方法もある（$L = 1$ のときはベイズ識別）．

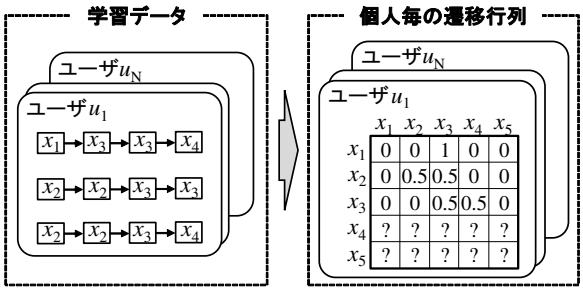

図 1: 最尤推定により学習された個人毎の遷移行列（$M = 5$）.「?」は遷移確率が不明な要素.

2.2 問題点

文献 [6, 7] では，個人毎の遷移行列を独立に学習している．しかしながら，この学習方法では個人が過去に公開していた位置情報が少ない場合に，各遷移行列の学習データが不足する．

例えば，文献 [6] では最尤推定により個人毎の遷移行列を学習しているが [1]，この場合における学習データ不足の問題を，図 1 を用いて説明する．この例では，ユーザ u_1 の遷移行列の学習に使用できるデータは，4 つの位置情報からなる移動系列情報 3 つのみである（領域数は $M = 5$）．これらの移動系列からは，領域 x_4 或いは領域 x_5 から次の領域への遷移が観測されないため，これを基に最尤推定を行うと，領域 x_4 或いは領域 x_5 からの遷移確率が不明となる．文献 [7] も，個人毎の遷移行列を独立に学習しているため，同様である．

従って，文献 [6, 7] はこのような場合に高い精度で位置情報プライバシーへの攻撃を行うことができない，という問題を抱えている．

3 テンソル分解を用いた個人毎の遷移行列の学習

本研究では，前節で述べた学習データ不足の問題を解決するため，テンソル分解 [8, 9, 10, 11]

[1] 文献 [7] では，学習用の移動系列情報のうち幾つかの位置が欠損している場合を考慮し，この欠損位置を Gibbs サンプリングによって推定しながら，ベクトル $\mathbf{p_{n,i}} = (p_{n,i,1}, \cdots, p_{n,i,M})$ が従う分布を学習している．しかし，この欠損位置がない場合に $\mathbf{p_{n,i}}$ の最頻値は最尤推定値と一致するため，本稿では最尤推定の場合のみを考える．

に着眼する．テンソルとは，ベクトルを1次元テンソル，行列を2次元テンソルとして含む一般的な概念であり，多次元配列として解釈できるものである [8]．本研究では，個人毎の遷移行列の集合をユーザ（User），領域（From Region），次の時刻における領域（To Region）の3軸で構成される3次元テンソルと見做し，これを「遷移確率テンソル」と呼ぶ（図2参照）．

第3.1節でテンソル分解について説明し，続く第3.2節でテンソル分解を用いた遷移確率テンソルの学習法を提案する．

3.1 テンソル分解

テンソル分解は，テンソルを低ランクな因子行列や低次元の因子ベクトルなどを用いて分解することで近似するものである．これにより，少量の学習データから，テンソルを（未観測な要素を含めて）頑健に推定することが可能となる．テンソルの分解手法としては様々なものがあり，Tucker分解やCP分解が代表的であるが [8]，本稿ではPITF（Pairwise Interaction Tensor Factorization）[9, 10] に着眼する．これはTucker分解やCP分解よりも簡潔なモデルであり，より良い性能を実現できる [9] ものとして，近年注目されている手法である．

以下，PITFについて詳述する．ここでは遷移確率テンソルと同様にUser，From Region，To Regionの3軸で構成される（但し，各要素は確率ではなく実数値をとる）テンソル $\mathcal{A} \in \mathbb{R}^{N\times M\times M}$ を例として考え，その第 (n,i,j) 要素を $a_{n,i,j} \in \mathbb{R}$ とする．PITFでは，$a_{n,i,j}$ が以下の式で表されるようにテンソル $\mathcal{A}$ を分解する．

$$\begin{aligned}&\hat{a}_{n,i,j}\\ &= \langle \mathbf{u}_n^{(a)}, \mathbf{v}_i^{(a)} \rangle + \langle \mathbf{u}_i^{(b)}, \mathbf{v}_j^{(b)} \rangle + \langle \mathbf{u}_j^{(c)}, \mathbf{v}_n^{(c)} \rangle \quad (5)\\ &= \sum_{k=1}^{K} u_{n,k}^{(a)} v_{i,k}^{(a)} + \sum_{k=1}^{K} u_{i,k}^{(b)} v_{j,k}^{(b)} + \sum_{k=1}^{K} u_{j,k}^{(c)} v_{n,k}^{(c)} \quad (6)\end{aligned}$$

但し，$\hat{a}_{n,i,j}$ は $a_{n,i,j}$ のPITFによる近似値であり，$\mathbf{u}_n^{(a)}, \mathbf{v}_i^{(a)}, \mathbf{u}_i^{(b)}, \mathbf{v}_j^{(b)}, \mathbf{u}_j^{(c)}, \mathbf{v}_n^{(c)} \in \mathbb{R}^K$ は K 次元因子ベクトルである（K は通常 N や M より小さな値となるように予め決めておく）．また，$u_{n,k}^{(a)}, v_{i,k}^{(a)}, u_{i,k}^{(b)}, v_{j,k}^{(b)}, u_{j,k}^{(c)}, v_{n,k}^{(c)}$ をモデルパラメータと呼び，その集合を $\Theta = \{u_{n,k}^{(a)}, v_{i,k}^{(a)}, u_{i,k}^{(b)}, v_{j,k}^{(b)}, u_{j,k}^{(c)}, v_{n,k}^{(c)} | n \in [N], i,j \in [M], k \in [K]\}$ とする．

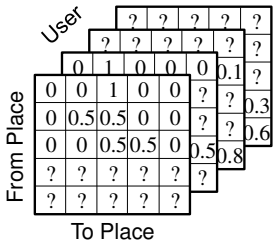

図 2: 遷移確率テンソル

このようにPITFでは，テンソルにおける任意の2軸間の相互作用を，因子ベクトルの内積という形でモデル化する．即ち，$\{\mathbf{u}_n^{(a)} | n \in [N]\}$ と $\{\mathbf{v}_i^{(a)} | i \in [M]\}$ はUserとFrom Region間の相互作用，$\{\mathbf{u}_i^{(b)} | i \in [M]\}$ と $\{\mathbf{v}_j^{(b)} | j \in [M]\}$ はFrom RegionとTo Region間の相互作用，$\{\mathbf{u}_j^{(c)} | j \in [M]\}$ と $\{\mathbf{v}_n^{(c)} | n \in [N]\}$ はTo RegionとUser間の相互作用をそれぞれモデル化する．このように分解することで，似た因子ベクトル（即ち，似たユーザ，似た領域）同士で影響を及ぼし合いながらテンソルの学習を行うことが可能となり，未観測な要素の推定もできるようになる．

3.2 遷移確率テンソルの学習

前節では実数値から構成されるテンソルを分解する例を述べたが，図2のような遷移確率で構成されるテンソルを分解した研究例は少なく，筆者らの知る限りでは文献 [10] 程度である．文献 [10] は，本やCDなどの商品の推薦に向けて，個人毎の商品の遷移行列からなるテンソルを分解している．しかし，ここでは商品の順位付けを目的としており，遷移確率の値そのものは求めていない．一方で，我々が考えている位置情報プライバシーへの攻撃では，第2.1節の式 (3) のように，遷移確率 $p_{n,i,j}$ を用いる必要がある．

そこで，本研究ではテンソル分解を用いて遷移確率テンソルの各要素の値を学習する手法を提案する．但し，遷移確率はTo Regionに対する和が1となっており（即ち，$\sum_j p_{n,i,j} = 1$），この制約の下で遷移確率の値を直接学習するのは困難である．従って，本研究では学習データから

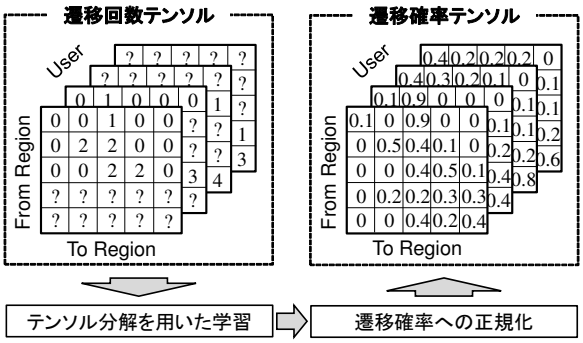

図 3: 提案する遷移確率テンソルの学習法

各遷移の回数を数えることで求めた「遷移回数テンソル」を分解して各要素を学習した後，To Region に対する和が 1 となるように正規化することで遷移確率テンソルを求める（図 3 参照）．

以下，提案手法を説明する．ここでは第 3.1 節で説明した PITF を分解手法として用いる．学習データから各遷移の回数を数えることで求めた遷移回数テンソルを，第 3.1 節のテンソル $\mathcal{A}$ に当てはめる（即ち，$a_{n,i,j}$ はユーザ $u_n \in \mathcal{U}$ の領域 $x_i \in \mathcal{X}$ から領域 $x_j \in \mathcal{X}$ への遷移回数）．本研究では，遷移回数は非負値であることを考慮し，この制約の下，以下の式で示される正規化二乗誤差 [11] を最小化するモデルパラメータ集合 $\hat{\Theta}$ を求める．

$$\hat{\Theta} = \arg\min_{\Theta \geq 0} \sum_{(n,i,j) \in \mathbb{D}} (a_{n,i,j} - \hat{a}_{n,i,j})^2 + \lambda ||\Theta||_F^2 \quad (7)$$

但し，$\mathbb{D} = \{(n,i,j) | \sum_j a_{n,i,j} \geq 1\}$ である．

式 (7) の第 1 項は，ユーザ u_n の領域 x_i からの遷移が 1 つ以上存在する遷移回数テンソルの要素に対する二乗誤差の和である．即ち，対応する遷移確率が不明でない要素を用いて $\hat{\Theta}$ を求める．（その後，式 (6) より遷移確率が不明な要素を含めて各要素を求める．）また，第 2 項は過学習を防ぐためのもので，$||\cdot||_F^2$ はフロベニウスノルム（全要素の 2 乗和）である．$\lambda (\geq 0)$ は正規化係数と呼ばれ，通常 cross-validation によって学習データから自動決定する [11]．$\Theta \geq 0$ は，各モデルパラメータが非負値であることを意味し，これにより遷移回数の推定値 $\hat{a}_{n,i,j}$ の非負性が保証される（式 (6) 参照）．

式 (7) の最適化問題を厳密に解くのは困難なため，本研究では近似解を求める手法として代表的な ANLS（Alternating Nonnegative Least Square）[11] を用いる．ANLS は，ある 1 つのモデルパラメータに関して，残りのパラメータを固定しながら非負値という制約下で最適化問題を解き，これを各モデルパラメータが収束するまで繰り返す手法である．式 (6) を式 (7) に当てはめ，残りのパラメータを固定して式 (7) を解くことで，以下の更新式を得る．

$$u_{n,k}^{(a)} \leftarrow \frac{\left[\sum_{\mathbb{D}_n} (a_{n,i,j} - \hat{a}_{n,i,j} + u_{n,k}^{(a)} v_{i,k}^{(a)}) v_{i,k}^{(a)}\right]_+}{\sum_{\mathbb{D}_n} (v_{i,k}^{(a)})^2 + \lambda} \quad (8)$$

$$v_{i,k}^{(a)} \leftarrow \frac{\left[\sum_{\mathbb{D}_i} (a_{n,i,j} - \hat{a}_{n,i,j} + u_{n,k}^{(a)} v_{i,k}^{(a)}) u_{n,k}^{(a)}\right]_+}{\sum_{\mathbb{D}_i} (u_{n,k}^{(a)})^2 + \lambda} \quad (9)$$

$$u_{i,k}^{(b)} \leftarrow \frac{\left[\sum_{\mathbb{D}_i} (a_{n,i,j} - \hat{a}_{n,i,j} + u_{i,k}^{(b)} v_{j,k}^{(b)}) v_{j,k}^{(b)}\right]_+}{\sum_{\mathbb{D}_i} (v_{j,k}^{(b)})^2 + \lambda} \quad (10)$$

$$v_{j,k}^{(b)} \leftarrow \frac{\left[\sum_{\mathbb{D}_j} (a_{n,i,j} - \hat{a}_{n,i,j} + u_{i,k}^{(b)} v_{j,k}^{(b)}) u_{i,k}^{(b)}\right]_+}{\sum_{\mathbb{D}_j} (u_{i,k}^{(b)})^2 + \lambda} \quad (11)$$

$$u_{j,k}^{(c)} \leftarrow \frac{\left[\sum_{\mathbb{D}_j} (a_{n,i,j} - \hat{a}_{n,i,j} + u_{j,k}^{(c)} v_{n,k}^{(c)}) v_{n,k}^{(c)}\right]_+}{\sum_{\mathbb{D}_j} (v_{n,k}^{(c)})^2 + \lambda} \quad (12)$$

$$v_{n,k}^{(c)} \leftarrow \frac{\left[\sum_{\mathbb{D}_n} (a_{n,i,j} - \hat{a}_{n,i,j} + u_{j,k}^{(c)} v_{n,k}^{(c)}) u_{j,k}^{(c)}\right]_+}{\sum_{\mathbb{D}_n} (u_{j,k}^{(c)})^2 + \lambda} \quad (13)$$

但し，$\mathbb{D}_n = \{(i,j) | \sum_j a_{n,i,j} \geq 1\}$，$\mathbb{D}_i = \{(n,j) | \sum_j a_{n,i,j} \geq 1\}$，$\mathbb{D}_j = \{(n,i) | \sum_j a_{n,i,j} \geq 1\}$ であり，$[x]_+ = \max(x, 0)$ である．

式 (8) により $\{\mathbf{u}_n^{(a)} | n \in [N]\}$ を更新し，次に式 (9) により $\{\mathbf{v}_i^{(a)} | i \in [M]\}$ を更新する．同様のことを式 (13) まで行い，各モデルパラメータが収束するまで式 (8)-(13) を繰り返して学習を行う．尚，モデルパラメータの初期値については，全て 0 に設定すると式 (8) の分母の値が小さくなり，その後，各モデルパラメータが大きく振動して収束が遅くなるため，避けるべきである．これまで様々な初期化方法が提案されているが [15]，この中で，区間 $(0, 1)$ の実数値をランダムに選ぶ方法がシンプルでかつ最も広く用いられているため，本稿でもこれを用いる．

提案する遷移確率テンソルの学習アルゴリズムをまとめると，以下のとおりである．

1. 式 (8)-(13) を各モデルパラメータが収束するまで繰り返すことで，Θ を学習する．
2. 式 (6) より遷移回数テンソルを求め，To Region に対する和が 1 となるように正規化して遷移確率テンソルを求める．

4　位置予測攻撃への最適な対策

本研究では，個人毎の遷移行列を用いた攻撃として位置予測攻撃を考え（第 2.1 節参照），その対策として位置情報の曖昧化（Precision Reducing）[7] を考える．これは，複数の領域を 1 つのグループ（以後，曖昧領域）に纏め，その曖昧領域の ID のみを公開することで，その後の位置の予測を困難にする手法である．

ここでは簡単のため，縦 2^B 個 × 横 2^B 個（B は自然数）の領域を考え，公開する領域の縦方向と横方向の ID をそれぞれ B ビットのバイナリ系列で表現したときの下位 b ($b \in \{0, 1, \cdots, B\}$) ビットをそれぞれ削除することを考える（図 4 参照）．削除するビット数 b を増やすほど曖昧領域サイズが大きくなり，その後の位置の予測が困難になるが，ユーザが利用できるサービスの質は低下する（尚，$b = B$ のときは位置情報を公開しないのと等価）．即ち，プライバシーと有用性の間には，トレード・オフの関係がある．

本研究では，このトレード・オフを最適化する曖昧化手法を提案する．具体的には，攻撃者が 1 つの領域を予測結果として出力する攻撃の成功確率（予測結果が正しい確率）を要求値 $\overline{\alpha}$ ($0 \leq \overline{\alpha} \leq 1$) 以下に抑えつつ，必要な削除ビット数 b（曖昧領域サイズ）を最小化する手法を提案する．

ユーザ $u_n \in \mathcal{U}$ が時刻 $t \in \mathbb{Z}$ において領域 $x_i \in \mathcal{X}$ にいることを公開したい一方で，時刻 $t + c \in \mathbb{Z}$（c は自然数）に領域 $x_j \in \mathcal{X}$ にいることは秘密にしたいものとする．領域 x_i の下位 $b \in \{0, 1, \cdots, B\}$ ビットを削除した曖昧領域を $\tilde{x}_i^{(b)}$ と表記する．このとき，提案手法は以下の最適化問題を解く．

$$\begin{aligned} &\textbf{Minimize} \quad b \\ &\textbf{subject to} \quad \alpha_c^{(b)}(x_j) \leq \overline{\alpha} \end{aligned} \tag{14}$$

但し，$\alpha_c^{(b)}(x_j)$ は曖昧領域 $\tilde{x}_i^{(b)}$ が得られた後の，時刻 $t + c$ における領域 x_j に対する事後確率

$$\alpha_c^{(b)}(x_j) = \Pr(X_{n,t+c} = x_j | X_{n,t} = \tilde{x}_i^{(b)}) \tag{15}$$

である．これは Forward アルゴリズムより，ユーザ u_n の遷移行列 P_n を用いて，再帰的に

$$\alpha_\tau^{(b)}(x_j) = \sum_{\rho=1}^{M} \alpha_{\tau-1}^{(b)}(x_\rho) p_{n,\rho,j} \tag{16}$$

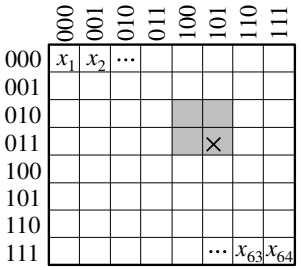

図 4: 位置情報の曖昧化（$B = 3$，$b = 1$，×印が実際の位置，灰色の部分が曖昧領域）

と計算できる．但し，$1 \leq \tau \leq c$ であり，

$$\alpha_0^{(b)}(x_j) = \begin{cases} \dfrac{\pi_{n,j}}{\sum_{x_\rho \in \tilde{x}_i^{(b)}} \pi_{n,\rho}} & (x_j \in \tilde{x}_i^{(b)} \text{ のとき}) \\ 0 & (x_j \notin \tilde{x}_i^{(b)} \text{ のとき}) \end{cases} \tag{17}$$

である．$\pi_{n,j}$ はユーザ u_n が領域 x_j にいるという事前確率である．$\{\pi_{n,j} | j \in [M]\}$ は一様分布を仮定する，或いは遷移行列 P_n から定常確率を求めるなどの方法を用いて予め定めておく．

提案手法は，削除ビット数 b を 0 から $B-1$ まで 1 ずつ増やしながら，事後確率 $\alpha_c^{(b)}(x_j)$ を式 (16)(17) により計算し，$\overline{\alpha}$ 以下になった時点でそのときの b を採用することで，式 (14) の最適化問題を解く．但し，$b = B-1$ でも $\alpha_c^{(b)}(x_j) \leq \overline{\alpha}$ とならない場合は，$b = B$ を採用する（即ち，位置情報 x_i を公開しない）．削除ビット数 b を決定した後，曖昧領域 $\tilde{x}_i^{(b)}$ を出力する．

このように式 (14) の最適化問題を解くことによる効果を説明する．攻撃者がベイズ識別により位置予測攻撃を行った（即ち，事後確率 $\alpha_c^{(b)}(x_j)$ の最も大きい領域 x_j を予測結果とした）とき，事後確率が正しく求まるという仮定の下で攻撃成功確率は最大となり，その値は $\alpha_c^{(b)}(x_j)$ となる [14]．従って，式 (14) を解くことで，どのような攻撃に対しても，その成功確率を $\overline{\alpha}$ 以下に抑えることが可能となり，その条件下で削除ビット数 b を最小化することが可能となる．

ここで重要なのは，対策者（曖昧化を施す者）は式 (14) の事後確率 $\alpha_c^{(b)}(x_j)$ を正しく求められれば，この最適化を実現できる点である．事後確率 $\alpha_c^{(b)}(x_j)$ を正しく求めるためには，ユーザ u_n の遷移行列 P_n を正しく求める必要があるが，これは攻撃者が学習したものと同じである必要

はない．例えば，位置情報を公開するサービス提供者側で，(公開していない) ユーザ u_n の過去の位置情報を沢山持っている場合には，それを用いて遷移行列 P_n を学習した上で曖昧化を施してもよい．この遷移行列の適切な学習により，プライバシーと有用性の両立が可能となる．

5 評価実験

5.1 実験条件

第 3.2 節および第 4 章の提案手法の有効性を検証するため，タクシーの移動系列データ (CRAWDAD[12]) を用いた評価実験を行った．このデータは，サンフランシスコにおける計 536 台 ($N = 536$) のタクシーの位置情報を，30 日間に渡って収集したものである．

このデータから約 10 分間隔で位置情報を取り出し，各タクシーにつき計 10 個の位置情報からなる移動系列 1 個を学習用に，計 10 個の位置情報からなる移動系列 10 個を評価用に用いた．また，学習用の移動系列と評価用の移動系列の間には 1 日の時間間隔を設けた．領域については縦 8 個 × 横 8 個の領域 ($M = 64$，$B = 3$) を考え，縦方向と横方向のぞれぞれにおいて，学習用と評価用の全位置情報 (計 $536 \times 10 \times 11$ 個) の出現頻度が一様 (即ち，$1/8 = 12.5[\%]$) となるように境界を定めた．

第 3 章で説明した PITF のモデルパラメータの次元数は $K = 16$ とした ($K = 1, 2, 4, 8, 16, 32, 64, 128$ と試し，$K = 4$ まで予測精度が上がり，$K = 32$ 以降は下がることを確認しているが，本稿では紙面の都合上，詳細を割愛する)．また，各モデルパラメータの初期値としては，区間 $(0, 1)$ の実数値をランダムに選んだ．式 (7) の正規化係数 λ については学習用の全遷移 (計 536×9 個) をランダムに 10 分割し，cross-validation によって自動決定した．第 4 章の式 (17) で用いる事前確率 $\pi_{n,j}$ については，一様分布を仮定して $\pi_{n,j} = 1/64$ と設定した．

5.2 実験結果

まず，個人毎の遷移行列をテンソル分解によって学習する手法 (第 3.2 節) の有効性を調べるため，各遷移行列を独立に学習する場合 [6, 7] との比較を行った．具体的には，最尤推定を用いて各遷移行列を学習し，遷移確率 $p_{n,i,j}$ が不明な場合には一様分布を仮定して $p_{n,i,j} = 1/64$ とする手法と比較した．

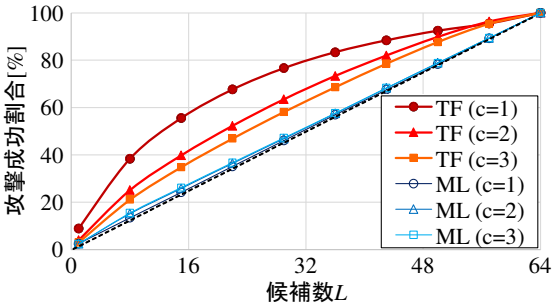

図 5: 位置予測攻撃の評価結果 (TF : テンソル分解，ML : 最尤推定，点線 : ランダムな推定)

図 5 に，位置予測攻撃における候補数 $L(\leq 64)$ (第 2.1 節参照) と攻撃成功割合 (L 個の候補の中に正解が含まれていた割合) との関係を示す．但し，「TF」はテンソル分解 (提案手法)，「ML」は最尤推定の結果であり，点線はランダムに推定した場合 (random guess) の精度である．また，予測する時刻としては $c = 1, 2, 3$ (即ち，約 $10, 20, 30$ 分後) の 3 通りを試した．

最尤推定の精度が random guess とほぼ同じであることが分かる．これは，学習データが非常に少なく (1 ユーザあたり，遷移行列の要素数が $M^2 = 4096$ 個なのに対して，学習用の遷移数は 9 個)，最尤推定では不明な遷移確率が多数出現したためと考える．これに対して，テンソル分解を用いた場合には大幅な精度向上を実現している (例えば，$L = 16, c = 1$ のとき，最尤推定の約 3 倍の精度である約 60[%] を実現)．テンソル分解によって，個人毎の遷移行列を効率的に学習できた効果が現れたものと考えている．尚，予測時刻が後になるにつれて (c の増加に伴い)，精度が下がっている．しかし，$c = 3$ のときも依然として random guess より高い精度を実現しており，テンソル分解を用いた位置予測攻撃の脅威が見てとれる．

次に，提案する曖昧化手法 (第 4 章) の有効性を調べるため，削除ビット数 $b \in \{0, 1, 2, 3\}$ (曖昧領域サイズ) を常に固定する手法との比

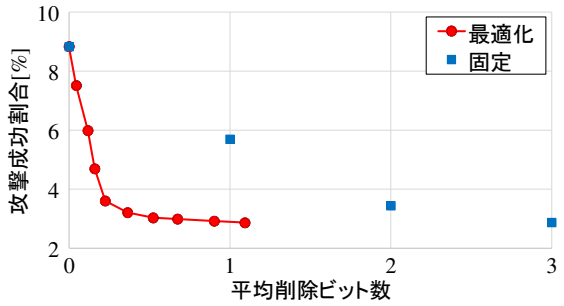

図 6: 曖昧化手法の評価結果（$L = 1, c = 1$）

較を行った．図 6 に，$L = 1, c = 1$ のときにおける平均削除ビット数（削除ビット数 b の全評価用データに対する平均）と攻撃成功割合との関係を示す．但し，「最適化」は提案手法の結果であり，ここでは攻撃者と同じ遷移行列を用い，攻撃成功確率の要求値 α を様々な値に変えたときに描く曲線を記している．また，「固定」は削除ビット数 b を常に固定する手法の結果である．

提案手法の方が，平均削除ビット数を大幅に低減できていることが分かる．例えば，提案手法では削除ビット数を常に $b = 3$ と固定する（即ち，位置情報を常に削除する）場合と同じ攻撃成功割合を保ったまま，平均削除ビット数を約 1.1 ビットまで減らせている．これは，提案手法ではユーザ毎，位置毎に曖昧領域サイズを最適化できたためと考えている．尚，このときの削除ビット数 $b \in \{0, 1, 2, 3\}$ の割合を調べたところ，それぞれ 57[%]，7.2[%]，5.9[%]，30[%] であった（即ち，半分以上は曖昧化を施していない）．以上により提案手法の有効性が示された．

6 まとめ

本稿ではテンソル分解を用いて個人毎の遷移行列を学習する手法を提案した後，位置予測攻撃への対策として曖昧領域サイズを最小化する位置情報の曖昧化手法を提案した．これらの提案手法と実験結果は，位置情報プライバシーに関する議論を一段と深めるという大きな意義に繋がったものと考える．

今後は，テンソル分解を用いた学習法のトラッキング攻撃 [7] などへの適用や，その対策を検討していく予定である．

謝辞 本研究に関して，産業技術総合研究所の兼村厚範氏，赤穂昭太郎氏，筑波大学の日野英逸氏より有益なコメントを頂いたので感謝する．

参考文献

[1] A. Vaccari, “Introducing A New Optional Feature Called Nearby Friends,” [Online] Available: http://newsroom.fb.com/news/2014/04/introducing-a-new-optional-feature-called-nearby-friends, 2014.
[2] 増田有孝, “ビッグデータを活用した高精度の道路交通情報サービス”, IT ソリューションフロンティア, vol.29, no.3, pp.16–19, 2012.
[3] C. Bettini et al., “Privacy in Location-Based Applications: Research Issues and Emerging Trends,” Springer, 2009.
[4] A. Y. Xue et al., “Destination Prediction by Sub-trajectory Synthesis and Privacy Protection against Such Prediction,” Proc. 2013 IEEE International Conference on Data Engineering (ICDE’13), pp.254–265, 2013.
[5] K. Minami and N. Borisov, “Protecting Location Privacy against Inference Attacks,” Proc. 9th ACM workshop on Privacy in the electronic society (WPES’10), pp.123–126, 2010.
[6] S. Gambs et al., “De-anonymization Attack on Geolocated Data,” Proc. 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom’13), pp.789–797, 2013.
[7] R. Shokri et al., “Quantifying Location Privacy,” Proc. 2011 IEEE Symposium on Security and Privacy (S&P’11), pp.247–262, 2011.
[8] T. G. Kolda and B. W. Bader, “Tensor Decompositions and Applications,” SIAM Review, vol.51, no.3, pp.455–500, 2009.
[9] S. Rendle and L. Shmidt-Thieme, “Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation,” Proc. 3rd ACM International Conference on Web Search and Data Mining (WSDM’10), pp.81–90, 2010.
[10] S. Rendle et al., “Factorizing Personalized Markov Chains for Next-Basket Recommendation,” Proc. 19th International Conference on World Wide Web (WWW’10), pp.811–820, 2010.
[11] J. Kim, “Nonnegative Matrix and Tensor Factorization, Least Squares Problems, and Applications,” Ph.D. Thesis, Georgia Institute of Technology, 2011.
[12] M. Piorkowski et al., “CRAWDAD data set epfl/mobility (v. 2009-02-24),” Downloaded from http://crawdad.org/epfl/mobility.
[13] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” Proc. IEEE, vol.77, no.2, pp.257–286, 1989.
[14] R. O. Duda et al., “Pattern Classification,” Wiley-Interscience, 2000.
[15] R. Albright et al., “Algorithms, Initializations, and Convergence for the Nonnegative Matrix Factorization,” SAS Technical Report, pp.1–18, 2014.