Local Differential Priacyによりプライバシーを考慮した位置情報分布推定

堀込　光[1]　菊池 浩明[1]

概要：スマートフォンの普及に伴い，サービス企業は，渋滞予測や混雑予想などのためにリアルタイムな位置情報を収集し公開している．しかし，こうしたサービスでは，個人の位置に関するプライバシが侵害されるの危険性がある．そこで，Google の RAPPOR などの Local Differential Privacy(LDP) 手法を用いて，個々 (Local) の真の位置情報を隠して正しく情報を集計し，プライバシを保護した人口分布推定が行われている．しかし，RAPPOR では，推定されている集計は最尤推定であり精度が十分ではない．そこで，本論文では，Expectation Maximizaion(EM) アルゴリズムを適用し，従来の RAPPOR よりも精度の高い推定方式を提案し，最尤推定よりも精度が高いことを報告する．

Privacy-Preserving Estimation Population Distribution using Local Differential Privacy

Hikaru Horigome[1]　Hiroaki Kikuchi[1]

1. はじめに

近年大幅に普及したスマートデバイスを私たちは常に所持している．これにより，サービス企業は人々のあらゆる行動を分析できるようになった．その統計情報は，防災や観光，交通など様々な分野で利用されている．例えば，株式会社 NTT ドコモが提供するモバイル空間統計 [1] では，年齢や性別についての 10 分おきの分布を提供しており，利用者は，任意の居住地を選択し人口分布をみることができる．しかし，事業者には全利用者の正確な位置履歴が管理されており，過失による情報漏洩や不正な内部犯行者によるプライバシ侵害の危険性がある．

個人情報を保護した情報出版技術の一つに Differential Privacy[2] がある．これは，収集した情報を公開する際に確率的なノイズを付加するなどして個人情報を保護する．確率的なノイズが用いられているので，値を曖昧にする匿名加工情報よりも統計的な値を算出するのに適している．しかし，匿名加工情報は，データは安全管理措置が適切である仮定の下で管理されており，プライバシ保護に関しての保証はない．そこで，Local Differential Pricacy[3] が注目されている．スマートデバイスからの情報を収集する際に確率的なノイズを付加してから収集するという技術である．これにより，サービス事業者でさえも真の値はわからない．

Local Differential Privacy の技術に関していくつかの手法が提案されている．Google は，2014 年に Erlingsson らによって提案された RAPPOR[4] という手法を，Apple は，2017 年に Privacy Count Mean Sketch(CMS)[5] という手法を提案し，情報を収集している．RAPPOR は，Randomize Response[6] に基づいており，真の値と偽の値を確率的に入れ替えることによりプライバシ情報を保護する．しかし，[4],[5] の先行研究では，真の値を推測する際，最尤推定法を用いているが，データの量が多い場合や値に偏りがあるときには誤差が大きい．

そこで，本研究では，EM アルゴリズム [7] を用いて人口を推定する方式を提案する．株式会社ナイトレイが公開する疑似人流データ [8] から抽出した 6,258 名のデータを用い，RAPPOR ベースの LDP により情報を収集し，この収集したデータから最尤推定と EM アルゴリズムにより東

[1] 明治大学 総合数理学部 先端メディアサイエンス学科
Department of Frontier Media Science, School of Interdisciplinary Mathematical Science, Meiji University

表 1　記号表

n	真の人口
n'	LDP により操作された人口
$\hat{n}$	最尤推定法による推定人口 ([4])
$n^{(*)}$	EM アルゴリズムによる推定人口 (提案)

京 23 区のそれぞれの人口を推測する．すべての安全指標 ϵ の値で EM アルゴリズムのほうが先行研究の RAPPOR の最尤推定法よりも誤差が小さいことを報告する．

2. Local Differential Privacy

2.1 定義

クライアントは確率メカニズム Q を通してデータ v を変化させて送信することで，サーバにクライアントの持つ真のデータを秘匿する．これによりユーザのプライバシーを保護する．Local Differential Privacy メカニズムは以下のように定義される．

定義 1. *[3]* Q を集合 V の要素 v を受けとって, 集合 Z の要素 z を出力する確率メカニズムとする. $\epsilon \geq 0$ においてハミング重み $\omega_H(v)$, $\omega_H(v')$ が 1 となる任意のペア $v, v' \in V$ と任意の部分集合 $S \subset Z$ に対して,Q が次の条件を満たす時, メカニズム Q は ϵ-*LDP* であるという.

$$Pr[Q(v) \in S] \leq e^{\epsilon} Pr[Q(v') \in S]$$

2.2 RAPPOR

RAPPOR は，2014 年に Erligsson らによって提案された手法である [4]．RAPPOR は，クライアント側が持っている情報の集合 V の要素 v_i に対して確率メカニズム Q を適用する．確率 p で出力 $z_i = v_i$，確率 $q = 1 - p$ で出力 $z_i = 1 - v_i$ とする．すなわち，

$$z_i = \begin{cases} v_i & w/p \quad p, \\ 1 - v_i & w/p \quad q. \end{cases}$$

例えば，あるユーザ 1 が $\boldsymbol{v} = (0, 1, 0, 0)$ という情報を持っており，別のユーザ 2 が $\boldsymbol{v'} = (0, 0, 1, 0)$ という情報を持っていたとする．ユーザ 1 の情報 $\boldsymbol{v}$ に確率アルゴリズム Q を通すことにより，出力 $\boldsymbol{z} = (0, 1, 0, 1)$ となったとすると，入力 $\boldsymbol{v} = (0, 1, 0, 0)$ という情報が出力 $\boldsymbol{z} = (0, 1, 0, 1)$ となる確率は，

$$Pr[Q(\boldsymbol{v}) = \boldsymbol{z}|\boldsymbol{v}] = p^3 q$$

となる．一方，ユーザ 2 の入力 $\boldsymbol{v'} = (0, 0, 1, 0)$ という情報が出力 $\boldsymbol{z} = (0, 1, 0, 1)$ となる確率は，

$$Pr[Q(\boldsymbol{v'}) = \boldsymbol{z}|\boldsymbol{v'}] = pq^3$$

となる．Kairouz らの binary mechanism[9] より，

$$\begin{cases} p = \frac{e^{\frac{\epsilon}{2}}}{1+e^{\frac{\epsilon}{2}}}, \\ q = \frac{1}{1+e^{\frac{\epsilon}{2}}} \end{cases}$$

とすると，

$$\frac{Pr[Q(\boldsymbol{v}) = \boldsymbol{z}|\boldsymbol{v}]}{Pr[Q(\boldsymbol{v'}) = \boldsymbol{z}|\boldsymbol{v'}]} = \frac{p^3 q}{pq^3} = e^{\epsilon}$$

となり，この確率メカニズム Q は，Local Differential Privacy を満たす．つまり，出力 $\boldsymbol{z} = (0, 1, 0, 1)$ のとき，この出力はユーザ 1 の入力 $\boldsymbol{v} = (0, 1, 0, 0)$ からなのか，ユーザ 2 の入力 $\boldsymbol{v'} = (0, 0, 1, 0)$ からなのか区別することができず，個人のデータを保護することができる．また，ϵ の値が小さいとき，データの変化量が大きくなり，安全性が高くなる．ϵ における確率 q の値を図 1 に示す．

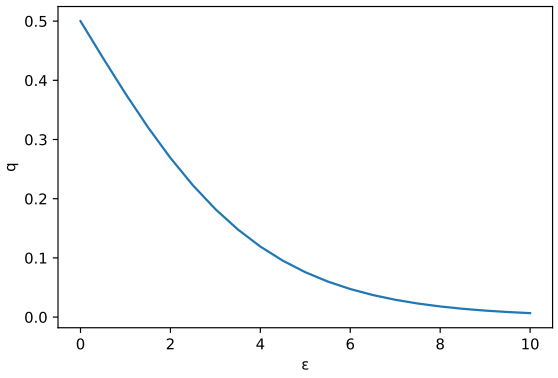

図 1　ϵ における確率 q

(証明) 入力 $\boldsymbol{v}$，$\boldsymbol{v'}$ を 23 ビットとする．$\boldsymbol{v}$，$\boldsymbol{v'}$ は 23 区のとこか 1 つについてのみ滞在しているデータであるため，それぞれある 1 ビットが 1，それ以外が 0 である．そのため，$\boldsymbol{v}$ と $\boldsymbol{v'}$ の異なるビット数 Δf は最大で 2 となる．

$$\Delta f = \sum_{k=1}^{23} ||\boldsymbol{v_i} - \boldsymbol{v'_i}||_1 \leq 2$$

また，出力 $\boldsymbol{z}$ と入力 $\boldsymbol{v}$，$\boldsymbol{v'}$ の異なるビット数をそれぞれ r，r' とすると，

$$\frac{Pr[Q(\boldsymbol{v}) = \boldsymbol{z}|\boldsymbol{v}]}{Pr[Q(\boldsymbol{v'}) = \boldsymbol{z}|\boldsymbol{v'}]} = \frac{p^{n-r} q^r}{p^{n-r'} q^{r'}} = (\frac{p}{q})^{r'-r}$$

となる．この時，

$$r' - r = \Delta f$$

であり，

$$\frac{Pr[Q(\boldsymbol{v}) = \boldsymbol{z}|\boldsymbol{v}]}{Pr[Q(\boldsymbol{v'}) = \boldsymbol{z}|\boldsymbol{v'}]} = (\frac{p}{q})^{\Delta f} = e^{\frac{\epsilon \Delta f}{2}} \leq e^{\epsilon}$$

となり，任意の $\boldsymbol{z} \in V$ について言えるので，ϵ-LDP の定義を満たす．

3. 提案手法

3.1 最尤推定

RAPPOR により 区$_i$ の真の人口 n_i を推定していく．最尤推定法による推定方法は以下の通りである．

区$_i$ の真の人口を n_i，ユーザ数を ℓ，RAPPOR アルゴリズムにより操作された値を加算して得られた i 区の人口を n_i' とする．確率 p，q を用いて，ある区 i のビットが実際に 1 であった n_i 人のうち n_ip 人が 1 を送信する確率が最も高い．また，i 区におらず，そのビットが実際は 0 であった $(\ell - n_i)$ 人のうち $(\ell - n_i)q$ 人が 1 と送信する．従って，

$$n_i' = n_ip + (\ell - n_i)q$$

が成り立ち，n_i について解くと最尤値 $\hat{n}$ は，

$$\hat{n} = \frac{n_i' - \ell q}{p - q}$$

となり，n_i を推測できる．

また，n 人のうち h 人が 1 を送信したとすると，n' を得る条件付き確率分布，すなわち，n_i' のユーザ数 ℓ に対する割合は，

$$Pr[n_i'|n,h] = \begin{pmatrix} n \\ h \end{pmatrix} p^h q^{n-h} + \begin{pmatrix} \ell - n \\ n' - h \end{pmatrix} p^{\ell-n-n'+h} q^{n'-h}$$

となる．図 2 に $h = np$ として求めた 14:00 の新宿区の人口 n' の確率分布を示す．

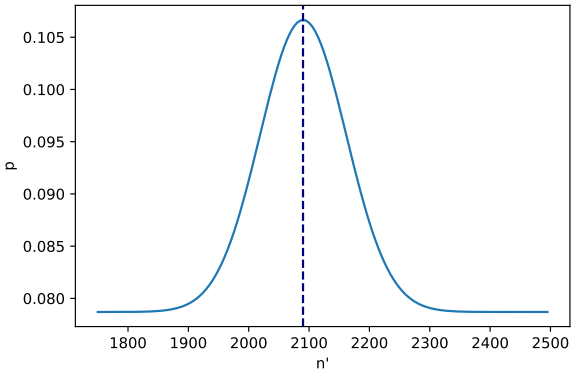

図 2 確率分布 $Pr[n'|n,h](\epsilon = 0.5, n = 505, \ell = 4640)$

3.2 EM アルゴリズム

EM アルゴリズムは，ベイズの定理に基づいて，確率パラメータを求める手法である．この手法では，E(Expectation)-step と M(Maximization)-step を推定値が収束するまで反復することにより，推定する．区$_i$ の全人口に対する割合を θ_i，各時刻におけるユーザ数を ℓ，RAPPOR アルゴリズムにより算出された 区$_i$ の人口を n_i' とする．E-step と M-step を以下に示す．

3.2.1 E-step

各ユーザ $u \leq \ell$ と 区$_i \leq 23$ について，処理を行う．ユーザ u の 区$_i$ における入力を $\boldsymbol{v_i} = (v_1, \ldots, v_{23}) \in \{0,1\}^{23}$，出力を $\boldsymbol{z_i} = (z_1, \ldots, z_{23}) \in \{0,1\}^{23}$ とする．θ_i の初期値 $\theta^{(0)_1} = \cdots = \theta_{23}^{(0)} = \frac{1}{23}$ とする．

入力が $v_i = 1$ だった場合，出力が z_i である条件付き確率は，

$$Pr[z_i|v_i = 1] = \frac{Pr[z_i, v_i = 1]}{Pr[v_i = 1]}$$

となる．また，ベイズの定理より，出力が z_i であった場合，入力 $v_i = 1$ である確率は，

$$\theta_{i,u}^{(k)} = Pr[v_i = 1|z_i] = \frac{Pr[z_i|v_i = 1]Pr[v_i = 1]}{\sum_{j=1}^{23} Pr[z_i|v_j = 1]Pr[v_j = 1]} = \frac{Pr[z_i|v_i = 1]\theta_i^{(k-1)}}{\sum_{j=1}^{23} Pr[z_j|v_j = 1]\theta_j^{(k-1)}}$$

となる．

3.2.2 M-step

すべてのユーザで E-step の計算が終わったら，$\theta_i^{(k)}$ をすべてのユーザの平均とする．

$$\theta_i^{(k)} = \frac{1}{\ell}\sum_{u=1}^{\ell} \theta_{i,u}^{(k-1)}$$

$\theta_i^{(k)}$ が収束して $\theta_i^{(k)} - \theta_i^{(k-1)} \leq \epsilon_2$ となるまで，E-step と M-step を反復する．その収束値 $\theta_i^{(*)}$ を用いて，

$$n_i^{(k)} = \ell\theta_i^{(k)}$$

で推定する．反復回数 k における推定人口 $n_i^{(k)}$ は図 3 のように変化する．ϵ の値が小さいほど収束するまでの反復回数 k は大きくなる．

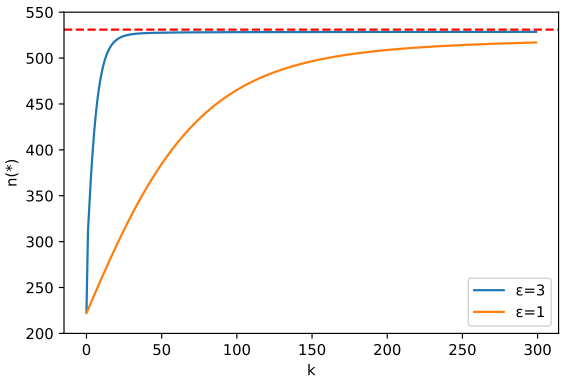

図 3 反復回数 k による推定値の変化

3.2.3 数値例

ユーザ 1 の入力 $\boldsymbol{v_1}$= $(v_{1,1}, v_{1,2}, v_{1,3}, v_{1,4})$ をハミング重み $\omega_H(v) = 1$ の 4 ビットのベクトルとし，出力 $\boldsymbol{z_1}$ = $(1,0,1,0)$ であるとする． それぞれの人口の割合

$\theta_1^{(0)} = \ldots = \theta_4^{(0)} = \frac{1}{4}$，$p = 0.6$，$q = 0.4$ とする．このとき，出力 $\boldsymbol{z_1}$ の入力が $v_{1,i} = 1$ である確率はそれぞれ，$Pr[\boldsymbol{z_1}|v_{1,1} = 1] = p^3q$ ，$Pr[\boldsymbol{z_1}|v_{1,2} = 1] = pq^3$ ，$Pr[\boldsymbol{z_1}|v_{1,3} = 1] = p^3q$ ，$Pr[\boldsymbol{z_1}|v_{1,4} = 1] = pq^3$ となり，$\theta_{1,i}^{(1)}$ を計算する．

$$\begin{aligned}\theta_{1,1}^{(1)} &= Pr[v_{1,1} = 1|\boldsymbol{z_1}] \\ &= \frac{p^3q\theta_1^{(0)}}{p^3q\theta_1^{(0)} + pq^3\theta_2^{(0)} + p^3q\theta_3^{(0)} + pq^3\theta_4^{(0)}} \\ &= \frac{\frac{1}{4}p^3q}{\frac{1}{4}p^3q + \frac{1}{4}pq^3 + \frac{1}{4}p^3q + \frac{1}{4}pq^3} = 0.34615\end{aligned}$$

同様に $\theta_{1,2}^{(1)}$，$\theta_{1,3}^{(1)}$，$\theta_{1,4}^{(1)}$ を計算する．

各ユーザについて同様に $\theta_{u,i}^{(1)}$ を求め，その平均を $\theta_i^{(1)}$ とする．これを $\theta_i^{(k)}$ が収束するまで反復させる．

4. 実験

4.1 実験目的

LDP アルゴリズムにおける最尤推定と EM アルゴリズムの精度を明らかにする．

4.2 収集データ

本研究では，株式会社ナイトレイより無料公開されている疑似人流データ [8] を使用する．このデータには 6,258 人の一日の位置情報が格納されている．ユーザの位置情報は緯度，経度で示されている．

この緯度，経度から Goole Map API を用いて 8:00 から 3 時間ごとのユーザの位置する市町村区を求めた．そして，各時刻における東京 23 区に属するユーザを収集した．

表 2 は，各時間ごとの東京 23 区に属している全人口である．また，区の番号を 8:00 における人口順とし，区番号 1 から 12 を表 3 に示す．表 4 では，各時刻の人口の平均値，最大値，最小値を示している．

そして，図 4 は，中野区，江東区，中央区，文京区における時間ごとの人口推移である．東京 23 区の 8:00 と 14:00 のヒートマップを図 6，図 7 に示す．

表 2 時間ごとの 23 区の人口

時間	人数
8:00	2,957
11:00	3,922
14:00	4,640
17:00	4,793
20:00	4,300
23:00	3,283

各時間ごとにそれぞれのユーザの位置情報をもとに位置情報の入力 $\boldsymbol{v}$ を作成していく．東京 23 区を列名とし，その時間にユーザが位置していた区を 1 とし，それ以外を 0 となる行列 $\boldsymbol{v}$ (23 行 1 列) を作成する．その行列 $\boldsymbol{v}$ に

表 3 各時刻の 23 区の人口 ℓ

番号	時間	8:00	11:00	14:00	17:00	20:00	23:00
n_1	世田谷区	295	331	367	403	368	317
n_2	新宿区	278	414	505	531	454	304
n_3	港区	267	393	509	479	416	284
n_4	渋谷区	262	394	533	532	479	351
n_5	千代田区	186	381	506	496	476	248
n_6	杉並区	165	209	227	246	187	188
n_7	中野区	154	117	116	133	141	142
n_8	江東区	126	119	108	121	97	117
n_9	中央区	121	177	216	188	148	118
n_{10}	大田区	114	140	121	118	125	115
n_{11}	文京区	98	166	181	197	206	143
n_{12}	品川区	98	147	182	173	147	99

表 4 各時刻の 23 区の平均人口と人口の最大値，最小値

時間	8:00	11:00	14:00	17:00	20:00	23:00
平均	128.57	170.52	201.74	208.39	186.96	142.74
最大値	295	414	533	532	479	351
最小値	25	35	36	34	45	29

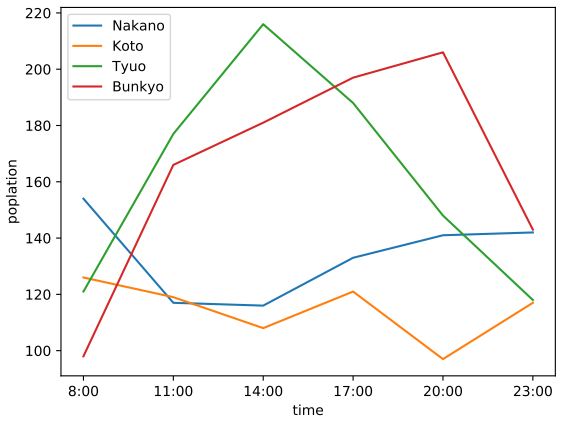

図 4 各区の人口推移

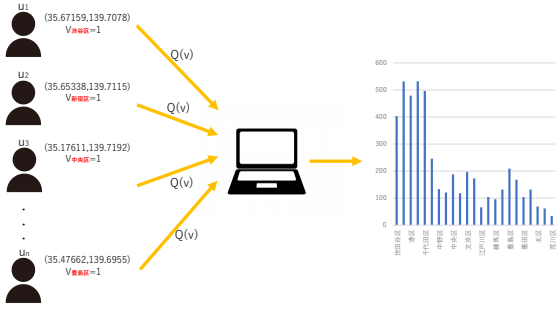

図 5 システム構成図

RAPPOR アルゴリズムを適用し，送信する．収集したデータに対し，最尤推定法と EM アルゴリズムを適用し実際の値を推測し，誤差を求める．例えば，ユーザがある時間に中野区にいるとすると入力 $\boldsymbol{v}$ は，

$$v = (n_{\text{世田谷区}}, n_{\text{中野区}}, n_{\text{渋谷区}}, \ldots, n_{\text{葛飾区}}, n_{\text{江戸川}}) = (0, 1, 0, \ldots, 0, 0)$$

となる．このデータを RAPPOR アルゴリズムを適用すると，出力 $\boldsymbol{z}$ は，

$$z = (n_{\text{世田谷区}}, n_{\text{中野区}}, n_{\text{渋谷区}}, \ldots, n_{\text{葛飾区}}, n_{\text{江戸川}}) = (1, 0, 0, \ldots, 1, 0)$$

となったとする．

各ユーザから $\boldsymbol{z}$ を出力し，出力 $\boldsymbol{z}$ の各区の和 $n'_i = \sum_{u=1}^{\ell} z_{i,u}$ を求める．この n' から最尤推定法と EM アルゴリズムを用いて各区の実際の人口 $\hat{n}$ と $n^{(*)}$ を推定していく．システム構成図を図 5 に示す．

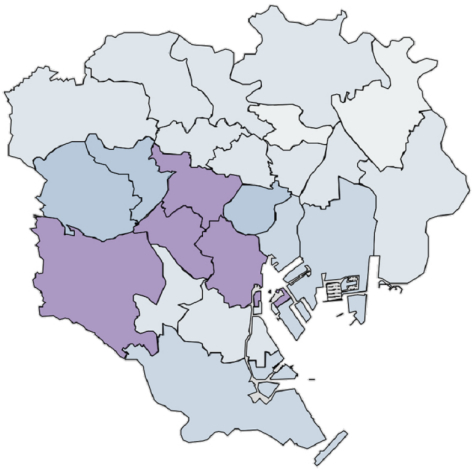

図 6 8:00 における人口分布

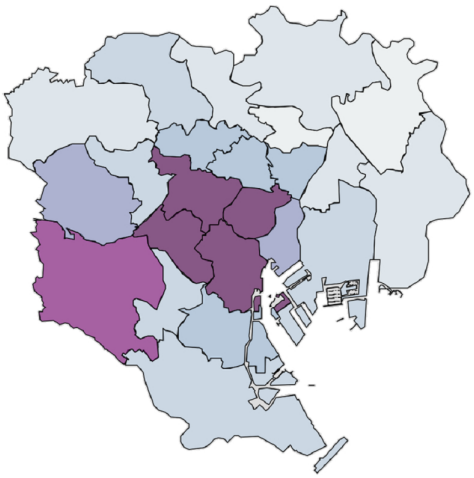

図 7 14:00 における人口分布

4.3 実験

最尤推定法と EM アルゴリズムを用いて RAPPOR アルゴリズムで収集したデータから推定した人口と実際の人口との誤差を ϵ の値を変化させて求めた．図 8 に真の人口 n_i を示す．この人口 n_i に RAPPOR を適用すると図 9 のようになり，n'_i に EM アルゴリズムを用いて人口を推測すると図 10 のようになる．誤差を平均絶対誤差 MAE として，次のようにして求める．

(1) 実際の人口データから各ユーザごとに滞在する区を 1，それ以外を 0 とする入力 $\boldsymbol{v}$ を作成する．
(2) 入力 v 用いて ϵ の値を 0.5 から 5 まで 0.5 ずつ変化させながら，RAPPOR を適用し出力 $\boldsymbol{z}$ を求める．
(3) 各ユーザの出力 $\boldsymbol{z}$ から各区の人口 n'_i を求める．
(4) (3) で求めた人口 n' に最尤推定法と EM アルゴリズムを適用し，各区の実際の人口 $\hat{n}$ と $n^{(*)}$ と推定する．
(5) 各区ごとに推定した人口と実際の人口の差の絶対値を求め，その和を S とする．
(6) (2) から (5) を 10 回行い，S の平均 $\hat{S}$ を求める．

S は，東京都の区の数は 23 で真の 区$_i$ の人口を n_i，推定した各区の人口を $\hat{n}_i$，$n_i^{(*)}$ とすると，

$$S = \sum_{i=1}^{23} |n_i - \hat{n}_i|$$

として求める．ϵ の値を変化させた各時間の誤差 $\hat{S}$ を実験結果を図 15 から図 20 に示す．

また，17:00 における $\epsilon = 0.1$ の際の最尤推定と EM アルゴリズムにおける推定人口は図 11 のようになる．図 12 には，各区の真の人口とそれぞれの推定人口の散布図を示している．図 11，図 12 ともに実験を 10 回行い，各推定 $\hat{n}, n^{(*)}$ の平均したものである．図 13 は EM アルゴリズムの，図 14 は最尤推定の真の人口に対する平均絶対誤差 MAE の散布図である．表 5 には，ϵ ごとの各時刻の誤差 S の平均を示す．

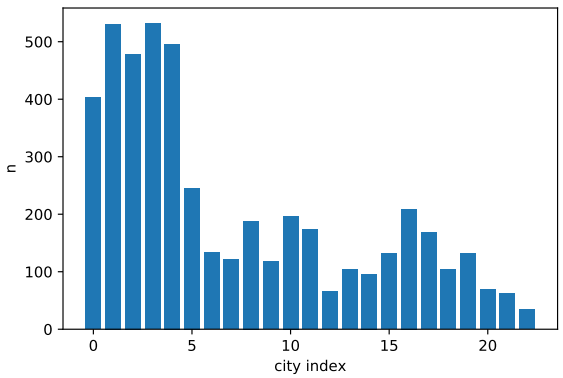

図 8 各区の真の人口 n_i

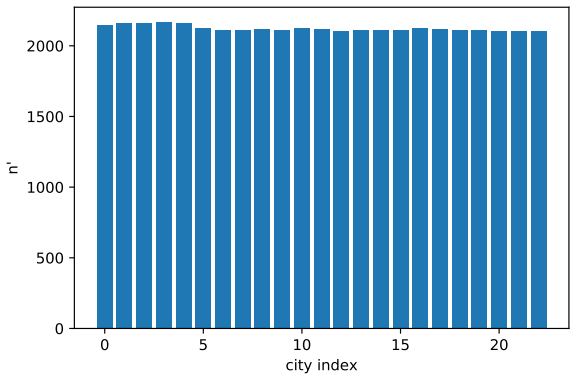

図 9　RAPPOR における人口 $n_i'(\epsilon = 0.5)$

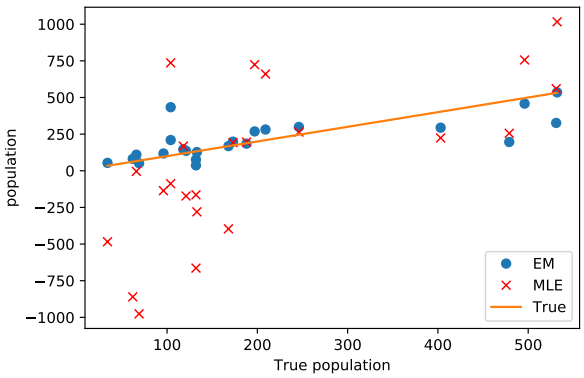

図 12　最尤推定と EM アルゴリズムによる推定人口

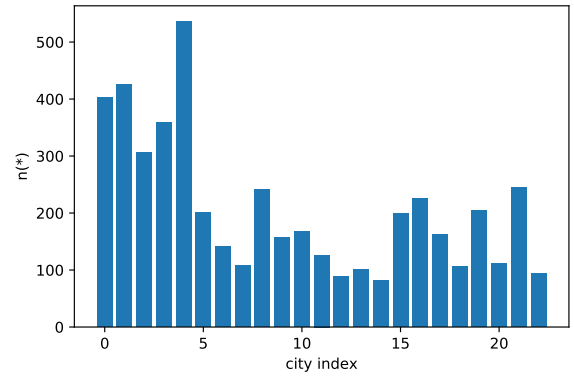

図 10　EM アルゴリズムによる推定人口 $n_i^{(*)}$

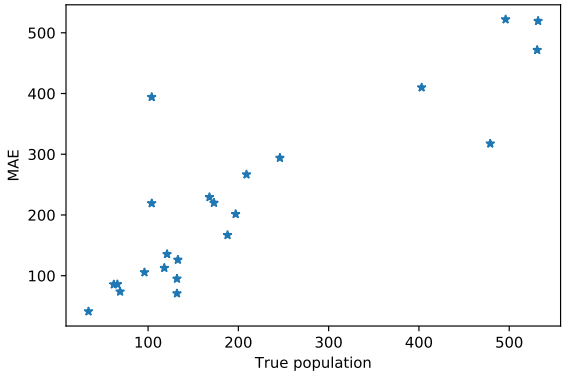

図 13　EM アルゴリズムによる真の人口に対する MAE

図 11　最尤推定と EM アルゴリズムによる各区の推定

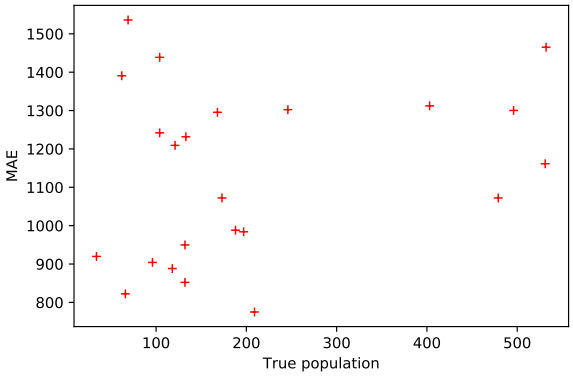

図 14　最尤推定による真の人口に対する MAE

4.4　考察

図 8 と図 15 から図 20 より，最尤推定に比べ EM アルゴリズムでは，すべての ϵ の値において MAE が小さかった．17:00 において $\epsilon = 0.5$ のとき，最尤推定と EM アルゴリズムの MAE の差は 1,871.515 であり，他の ϵ の MAE の差よりも大きい．$\epsilon \leq 0.5$ のときには，ϵ が小さくなるにつれ両推定の MAE の差は大きくなっていくと考えられる．また，最尤推定による MAE と真の人口との相関係数は 0.255 であったのに対し，EM アルゴリズムによる推定の MAE と真の人口との相関係数は 0.870 であった．つまり，EM アルゴリズムによる推定では，人口数と誤差には相関があると言え，図 13 に示すように，人口が多い区では誤差が大きくなる．

5.　おわりに

本研究では，Local Differential Privacy のアルゴリズムのひとつである RAPPOR を用いて疑似人流データからデータ収集を行い，最尤推定法と EM アルゴリズムを用い

表 5　ϵ における各時刻の平均絶対誤差 S

ϵ	EM	MLE
0.5	2971.31	4885.28
1.0	1846.69	2256.53
1.5	1404.14	1614.74
2.0	982.69	1072.24
2.5	780.58	849.41
3.0	628.41	710.34
3.5	524.80	601.56
4.0	407.92	503.13
4.5	349.73	434.63
5.0	287.42	363.16

て人口推定を行った．EM アルゴリズムを用いて推定を行う際，ϵ の値が小さいと反復回数が大きくなる．データ数が大きくなれば，収束値のわずかな違いで推定値に大きな影響を与える．収束回数の適切な決定が今後の課題である．

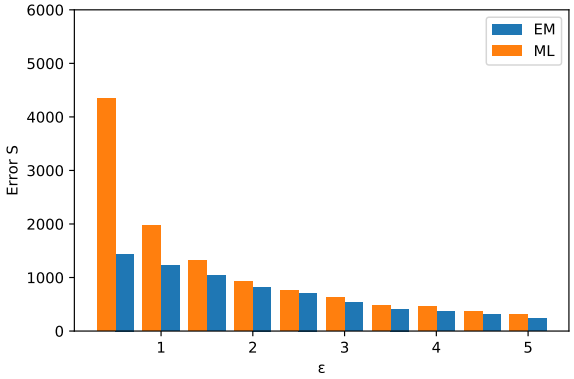

図 15　8:00 の誤差 S

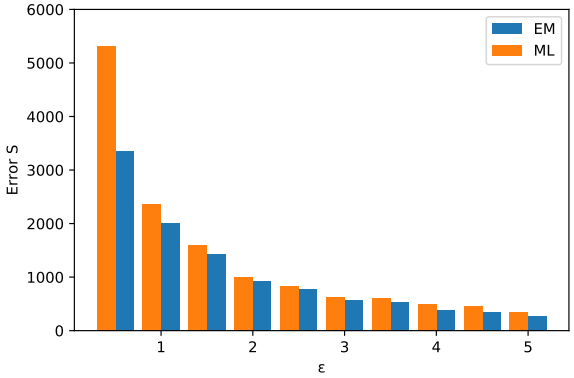

図 16　11:00 の誤差 S

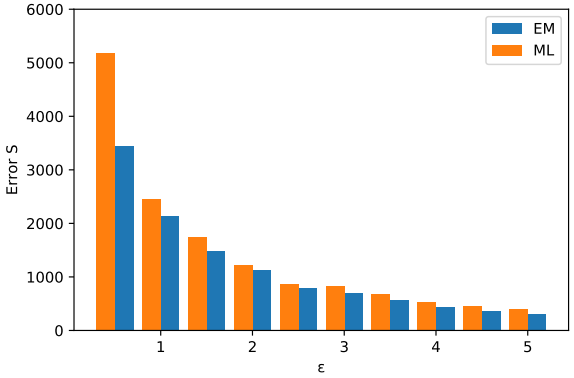

図 17　14:00 の誤差 S

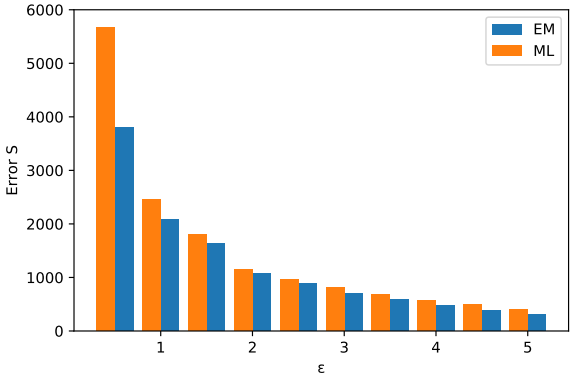

図 18　17:00 の誤差 S

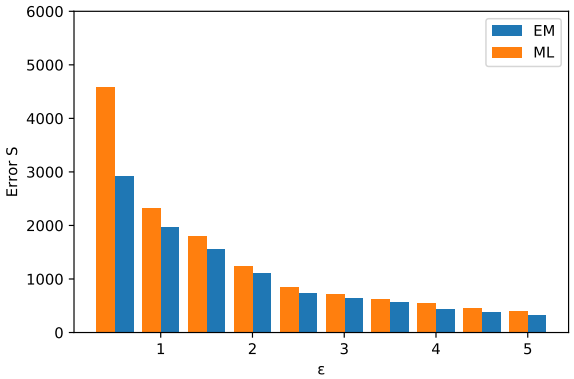

図 19　20:00 の誤差 S

参考文献

[1] NTT Docomo, “モバイル空間統計の「国内人口分布統計（リアルタイム版）」の提供開始” (https://www.nttdocomo.co.jp/info/news_release/2019/12/03_00.html/ ,2020 年 4 月参照).

[2] Cynthia Dwork,Frank McSherry,Kobbi Nissim,Adam Smith,“Calibrating noise to sensitivity in private data analysis”, TCC, Vol. 3876, pp. 265–284, 2006.

[3] John C Duchi,Michael I Jordan,Martin J Wainwright, “Local privacy and statistical minimax rates”, FOCS, pp. 429–438, 2013.

[4] Úlfar Erlingsson, Vasyl Pihur,Aleksandra Korolova,“RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response”,ACM Conference on Computer and Communications Security, pp. 1054-1067, 2014.

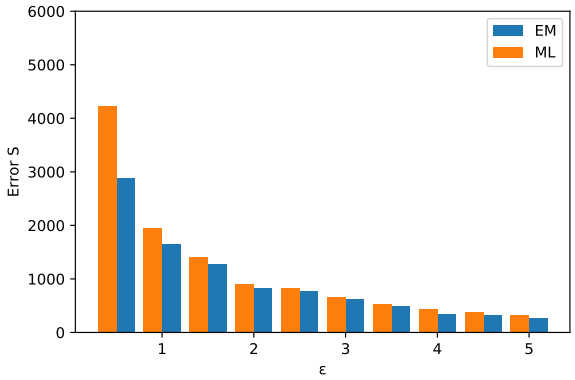

図 20 23:00 の誤差 S

[5] Diferential Privacy Team, Apple, “Learning with privacy at scale”, 2017 (`https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf`, 2012 年 4 月参照).

[6] Stanley L.Warner,“Randomized Response:A Survey Technique for Eliminating Evasive Answer Bias” ,Journal of the American Statistical Association, pp. 63-69, 1965.

[7] 宮川雅巳,“EM アルゴリズムとその周辺”, 応用統計学, Vol 16, No. 1, pp. 1-19, 1987.

[8] Nightley, “疑似人流データ”(`https://nightley.jp/archives/1954/`, 2019 年 10 月参照).

[9] Peter Kairouz, Sewoong Oh, and Pramod Viswanath, “Extremal mechanisms for local differential privacy”, In Advances in neural information processing systems, pp. 2879–2887, 2014.

[10] Anna Mochizuki,Hiroaki Kikuchi, “Priacy-preserving Collaborative Fitering Using Randomized Response”, Journal of information Processing, Vol. 21, No.4, pp. 617-623，2013.

[11] 小野元，福地一斗，佐久間淳，“局所差分プライバシー制約下における逐次 heavy hitters 検知”，DEIM Forum 2018, E1-3, 2018.

[12] Zhan Qin，Yin Yang，Ting Yu, “Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy”, ACM CCS 2016, pp.192-203 , 2016.

[13] Hiroaki Kikuchi，Jin Akiyama，Howard Gobioff，”Stochastic Voting Protocol To Protect Voters Privacy”, WIAPP’99, pp. 103-111, 1999.