DBSJ Japanese Journal, Vol.13-J, No.1
October 2014

クラスタリングと空間分割の併用による効率的なk-匿名化

Efficient k-Anonymization by Combining Clustering and Space Partitioning

新井 淳也♡ 鬼塚 真◇ 塩川 浩昭♠

Junya ARAI Makoto ONIZUKA
Hiroaki SHIOKAWA

個人情報利用時のプライバシー保護技術としてk-匿名化が用いられている．既存手法ではk-匿名化のためのデータ変換時に処理の速さと情報損失の低さのいずれかが犠牲になってしまう．本稿ではこの問題を解決する**2**つの手法を提案する．**1**つ目は，高速な処理と低い情報損失を両立するための，空間分割とクラスタリングの併用手法である．**2**つ目は，レコードを頂点としたグラフの構築によって情報損失をより減少させるクラスタリングアルゴリズムである．**2**つの手法を組み合わせることで，既存手法と同等の情報損失量を保ちつつ最大**10**倍の高速化が可能であることを確認した．

k-anonymity is a requirement for personal data to protect privacy. In order to achieve fast k-anonymization without sacrificing quality of generated data, in this paper we propose two methods: (i) combinatory method of clustering and space partitioning and (ii) a novel clustering algorithm that finds clusters by analyzing a nearest-neighbor graph. Our experimental results show that k-anonymization with our methods is at most 10 times faster than existing methods without incurring additional information loss.

1. はじめに

プライバシーを保護するため，データに含まれる個人情報から実在する個人を特定できない形へデータを変換する匿名化技術が用いられている．ここでは匿名化するデータとして表1 (a)のような個人の属性を含むレコードから成るテーブルを想定する．このテーブルから個人の年収を知られないよう匿名化するためには，名前や個人番号（マイナンバーや米国の社会保障番号）のような明確に個人を識別し得る属性を取り除くだけでは不十分である．なぜなら，性別，年齢，出身地といった単体では個人を特定できないような属性でも，複数の属性の組合せを別のデータベースと突き合わせることで個人を一意に特定できてしまう場合があるからである．このような属性群は準識別子 (quasi-identifier) と呼ばれている [27]．

準識別子から個人を特定できないようにする手法としてはk-匿名化 [27] が代表的である．k-匿名化とは，k人以上が同じ準識別子を持つようなデータへ元のデータを変換することである．表1 (a)を3-匿名化した結果の例を表1 (b), (c)に示す．表1 (b)は年齢を数値の範囲へ置き換え，出身地のようなカテゴリ値をより一般的な区分（千葉と東京なら南関東など）へ置き換える一般化 [27] によってk-匿名化されている．一方表1 (c)は数値を平均値へ，カテゴリ値を最頻値へ置き換えるミクロアグリゲーション [8] によってk-匿名化されている．どちらの表も3人が同じ準識別子を持つため，準識別子をもとに年収を特定することはできない．

表 1: 年収データに対するk-匿名化の例

(a) 元のテーブル

名前	性別	年齢	居住地	年収
越谷	男	24	千葉	350 万
加賀山	女	31	埼玉	300 万
宮内	女	56	埼玉	1 億 200 万
石川	女	36	茨城	800 万
一条	男	31	東京	400 万
富士宮	女	28	千葉	600 万

(b) 一般化により 3-匿名化されたテーブル

名前	性別	年齢	居住地	年収
—	人	[24–31]	南関東	350 万
—	女	[31–56]	関東	300 万
—	女	[31–56]	関東	1 億 200 万
—	女	[31–56]	関東	800 万
—	人	[24–31]	南関東	400 万
—	人	[24–31]	南関東	600 万

(c) ミクロアグリゲーションにより 3-匿名化されたテーブル

名前	性別	年齢	居住地	年収
—	男	28	千葉	350 万
—	女	41	埼玉	300 万
—	女	41	埼玉	1 億 200 万
—	女	41	埼玉	800 万
—	男	28	千葉	400 万
—	男	28	千葉	600 万

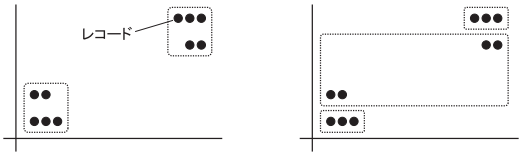
しかしながら，匿名化処理ではデータの変換によって情報が失われることが問題となる．例えば男女の情報が失われ「人」になってしまう場合（表1 (b)）や，逆の性別になってしまう場合（表1 (c)の最後のレコード）がある．このような情報損失が大きいと，匿名化されたデータと元データの間で分析結果が異なってしまう．情報損失を小さくするためには互いにデータの似通ったレコードを集めてグループを作り，各グループ内のレコードに同じ準識別子を与えればよい．従ってk-匿名化のためには，与えられたレコードの集合をkレコード以上から成るグループへ情報損失を最小化するように分割する必要がある．

kレコード以上から成るグループに分割された状態はしばしばk-分割と呼ばれている [9,16,26]．最適なk-分割の作成はNP困難である [2,14,22] ため，より情報損失を小さくするヒューリスティクスの研究が行われている．

k-分割を作成するヒューリスティクスは，kd-tree [12] 構築アルゴリズム等の空間分割に基づく手法 [17,22] とクラスタリングに基づく手法 [6,9,16,20,24,26] に分けられる．空間分割に基づく手法はレコード数nに対する計算量が$O(n \log n)$だが，作成可能な分割に制約があるためクラスタリングに基づく手法より情報損失が大きい [6]．一方でクラスタリングに基づく手法は計算量が$O\left(n^2\right)$と空間分割より大きいが，前述のとおり情報損失は小さい．このように既存の手法では計算量の小ささと低い情報損失を両立できていない．さらに，クラスタリングに基づく手法でもレコードのクラスタ構造を適切に扱えない場合がある．そのような場面の例を図1に示した．レコードは空間上の座標として点で表されている．理想的には図1 (a)のようにまとまりの良い2つのグループに分割すべきだが，既存手法 [6,16,26] は図1 (b)のように3つのグループへ分割し，情報損失を増大させてしまう．

そこで我々は2つの手法 (i) 空間分割とクラスタリングの併用手法，並びに (ii) 新しいk-分割アルゴリズムの友引法を提案する．1つ目の提案である空間分割とクラスタリングの併用手法では2段階の分割を行う．具体的には，まず空間分割アルゴリズムによって大まかな分割を行い，次にクラスタリングに基づくアルゴリズ

♡ 正会員　日本電信電話株式会社 NTT ソフトウェアイノベーションセンタ　arai.junya@lab.ntt.co.jp
◇ 正会員　大阪大学大学院情報科学研究科マルチメディア工学専攻ビッグデータ工学講座　onizuka@ist.osaka-u.ac.jp
♠ 正会員　日本電信電話株式会社 NTT ソフトウェアイノベーションセンタ　shiokawa.hiroaki@lab.ntt.co.jp



(a) 理想的なグループ　(b) 分布の広いグループ

図 1: 既存手法が作るグループの例 ($k = 3$)

表 2: 既存研究と本研究の分類

		準識別子の変換手法	
		ミクロアグリゲーション	一般化
分割手法	空間分割	—	[17, 22]
	クラスタリング	[9, 16, 20, 26], 友引法	[6, 14, 24]

ムでさらに細かく分割する．これにより，計算量の大きいクラスタリングの対象を空間分割で作られたグループ内のレコードに限定することで高速化すると共に，局所的にはクラスタリングによる柔軟な分割を行い情報損失を低下させることが可能となる．2つ目の提案である友引法は新しい k-分割クラスタリングアルゴリズムである．k 近傍グラフのようにレコードを頂点とし近傍点を接続したグラフの構築と分析によってクラスタを発見し，図 1 (a) のような情報損失の少ない分割を行う．

実験によって友引法は既存手法と比べ最大 16%情報損失を減少させることが分かった．空間分割を併用するとクラスタリングのみ用いた場合と比べ情報損失が増大するが，友引法を使用することでこの損失を補うことができる．その結果，空間分割と友引法の併用はクラスタリングに基づく既存手法のみ用いた場合と同等の情報損失量を維持しつつ最大 10 倍高速な処理を実現した．

2. 関連研究

既存研究と本研究を分類したものが表 2 である．k-匿名化処理は (i) k-分割の作成と (ii) k レコードに同じ準識別子を与えるための準識別子の変換という 2 段階に分けられる．さらに k-分割の作成手法は空間分割とクラスタリングに，準識別子の変換手法はミクロアグリゲーションと一般化にそれぞれ分類できる．ここでは 2 種類の k-分割手法について説明した後，本研究の位置づけを述べる．

2.1 空間分割に基づく分割手法

空間分割に基づく手法 [17, 22] ではレコードを多次元空間上の座標と見做し，k レコード以上含む空間を作れなくなるまで空間の分割を繰り返す．ここで言う空間分割とは，例えばレコード全体を年齢が 30 歳以下の集合と 30 歳より大きい集合に分割するような操作を指す．分割には kd-tree [12] や R-tree [13] のような空間インデックスの構築アルゴリズムが使用される．空間分割に基づく手法はクラスタリングに基づく手法より計算量が小さい一方，情報損失が大きいという欠点がある [6].

2.2 クラスタリングに基づく分割手法

クラスタリングに基づく手法では分割の決定にレコード間の距離を用いる．一般的なクラスタリングアルゴリズムでは 1 つのグループに k レコード以上含まれることを保証できないため，k-分割専用のアルゴリズムが研究されてきた [6, 9, 14, 16, 20, 24, 26].

クラスタリングに基づく手法としては Maximum Distance to Average Vector (MDAV) [10, 16] と Variable-size Maximum Distance to Average Vector (V-MDAV) [26] が代表的である．MDAV はグループの中心となるレコードからユークリッド距離が最も近い $k-1$ レコードを収集し 1 つのグループにする操作を繰り返す．グループのレコード数が k に固定されているため，クラスタを無視し図 1 (b) のように分布の広いグループを作ってしまう場合がある．そこでグループのレコード数が可変になるよう改良することで正確にクラスタを捉えようとしたのが V-MDAV である．V-MDAV のグループ形成ではまず MDAV と同様に近傍の k レコードを新しいグループとする．次に形成中のグループの周囲に存在するレコードが孤立しているか否かを頂点間の距離に関するパラメータ γ によって判定し，孤立したレコードが見つかった場合はそれをグループに追加する．この手続きによって孤立した頂点がグループに含まれないまま取り残され，最終的に分布の広いグループを作ってしまうことを防ぐ．しかし図 1 のように完全に孤立したレコードが存在しない場合に V-MDAV の手法は有効に機能せず，MDAV と同様の分割（図 1 (b)）を作成してしまうという問題がある．

アルゴリズム 1 Mondrian

```
1: function MONDRIAN(partition)
2:     if partition は分割不可能 then
3:         return {partition}
4:     else
5:         dim = CHOOSEDIMENSION()
6:         splitVal = FINDMEDIAN(partition, dim)
7:         lhs = {t ∈ partition | t.dim ≤ splitVal}
8:         rhs = {t ∈ partition | t.dim > splitVal}
9:         return MONDRIAN(lhs) ∪ MONDRIAN(rhs)
10:    end if
11: end function
```

2.3 本研究の位置付け

これまで空間分割に基づく手法とクラスタリングに基づく手法は別々に研究されてきた．本研究では空間分割とクラスタリングを併用することで両者の長所の組み合わせが可能であることを示す．さらに，既存手法より情報損失が小さい k-分割を作成する新しいアルゴリズムである友引法を提案する．友引法は分割手法としてクラスタリングを用い，準識別子の変換手法としてミクロアグリゲーションを想定した k-分割手法に分類される（表 2）．

3. 事前準備

3.1 Mondrian

提案手法では空間分割に基づく k-分割手法として Mondrian [22] を使用する．Mondrian の擬似コードをアルゴリズム 1 に示した．MONDRIAN 関数は引数 *partition* として受け取ったレコードの集合を分割できなくなるまで再帰的に分割を繰り返す．分割に使用する次元は *partition* に含まれる値の幅が最も大きい（最小値と最大値の距離が最大であるような）ものを選ぶ（5 行目）．また分割の境界値 *splitVal* として中央値を用いる（6 行目）ことで分割 *lhs*, *rhs* に含まれるレコード数をなるべく均等にしている．

3.2 情報損失指標

ミクロアグリゲーションにおける情報損失の指標としては次のように定義される SSE/SST を用いるのが一般的である [2, 9, 16, 26].

$$\mathrm{SSE/SST} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n_i}\left(x_{ij}-\bar{x}_i\right)^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}\left(x_{ij}-\bar{x}\right)^2}$$

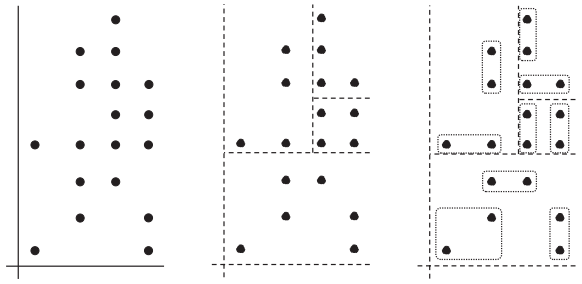
ただし g は構成されたグループの数，n_i は i 番目のグループの要素数，x_{ij} は i 番目のグループに属する j 番目のレコード，$\bar{x}_i$ は i 番目のグループに属するレコードの平均値，$\bar{x}$ は全レコードの平均値を表す．SSE はグループ内分散の和，SST はレコード全体の分散を示すことから，SSE/SST が小さいほど k-分割が最適に近いことを意味する．

4. 提案手法

4.1 空間分割とクラスタリングを併用した k-分割

空間分割手法として Mondrian を併用し k-分割を行う過程の例を図 2 に示した．提案手法は 2 段階から成る．

1. 粗粒度分割　空間分割によってレコード全体を $k_\sharp (k_\sharp > k)$ レコード以上含むグループへ分割する．



(a) 与えるレコードの分布 (b) 粗粒度分割後 ($k_\sharp = 4$) (c) 細粒度分割後 ($k = 2$)

図 2: 空間分割とクラスタリングを併用した匿名化の過程

2. 細粒度分割 粗粒度分割で得た各グループに対してさらにクラスタリングに基づく分割を行い，k-分割を作成する．

細粒度分割においてはレコード数 n に対して計算量 $O\left(n^2\right)$ 以上であるアルゴリズムの利用を想定しているため，粗粒度分割で作られたグループにレコード数の偏りがあると処理時間が増大してしまう．格子状の空間分割は高速だが，グループに含まれるレコード数が均一かつ $k_\sharp$ 以上であるような分割になるとは限らないため粗粒度分割で用いることはできない．本論文では粗粒度分割に適した手法として Mondrian を用いるが，Iwuchukwu らによる R-tree ベースの手法 [17] 等を用いることもできる．

空間分割を併用する利点は 3 つある．1 つ目は，1 章でも述べたとおり，高速な処理と低い情報損失を両立できること．実用上は $k = 5$ 程度の利用が多い [11] ため，万単位のレコード全体の大域的分布がもたらす情報損失への影響は限定的である．2 つ目は，情報損失と処理時間のどちらを優先するか，利用時の状況に応じて連続的に調整できること．大きい $k_\sharp$ では空間分割の特徴が強く現れ高速・高情報損失になるが，小さい $k_\sharp$ ではクラスタリングの特徴が現れ低速・低情報損失となる．3 つ目は，メモリに格納しきれないほど巨大なデータに対する k-分割処理を高速化できること．細粒度分割時は粗粒度分割で生成されたグループを一つずつ処理するため，処理中でないグループはストレージ上へスワップアウトできる．また，buffer tree [3] を用いることにより，空間分割で巨大なデータを扱う際の I/O 効率を高められることが Iwuchukwu らによって指摘されている [17].

4.2 友引法

空間分割に加え，より情報損失の少ない k-分割手法である友引クラスタリングアルゴリズムを提案する．アルゴリズムの擬似コードをアルゴリズム 2 に示す．TOMOBIKI 関数の引数は，m が後述するグラフ処理で使用される定数，k が k-匿名性の k，V がレコードの集合である．アルゴリズムはグラフの構築（MAKEGRAPH 関数）と分割（CUTGRAPH 関数）という 2 段階から成る．

アルゴリズム 2 を用いて最初にグラフの構築について説明する．構築するのは基本的に k 近傍グラフである．ただし，k が k-匿名化の k と重複するため，k 近傍グラフをここでは m 近傍グラフと呼び換える．$m < k$ のとき，通常の m 近傍グラフでは頂点数 k 以下の連結成分ができる場合がある．しかしここでは次のような手続きによってすべての連結成分が k 頂点以上含むグラフを構築する．まず初期状態としてレコードを頂点とし辺がないグラフ $(V, \emptyset)$ を与える（2 行目）．次に頂点数 k 未満の連結成分 G_c に属する頂点と G_c に属さない頂点のペアのうちユークリッド距離が最も近い m ペアを接続する操作（10–11 行目）を繰り返す．そして全ての連結成分が k 頂点以上含むグラフになったら終了する．このようにして作られたグラフをここでは (k, m)-近傍グラフと呼ぶ．なおグラフの構築を開始する前に全ての頂点の組み合わせについて距離を計算し，さらに各頂点について近傍の点を定数時間で取り出せるよう距離の順にソートしておくことで処理の高速化が可能である．簡潔さのためこの手法はアルゴリズム 2 に記述していない．

アルゴリズム 2 友引クラスタリングアルゴリズム

```
1: function TOMOBIKI(m, k, V)
2:     G = MAKEGRAPH(m, k, (V, ∅))
3:     return CUTGRAPH(k, G)
4: end function
5: function MAKEGRAPH(m, k, G)
6:     H = {G_c ∈ G 内の連結成分 | |V(G_c)| < k}
7:     if |H| = 0 then
8:         return G
9:     else
10:        ΔE = ⋃_{G_c∈H} {(u, v) ∈ P | u, v の近さが P 中で m 番目以内 }
11:                ただし P = V(G_c) × (V(G) − V(G_c))
12:        return MAKEGRAPH(m, k, (V(G), E(G) ∪ ΔE))
13:    end if
14: end function
15: function CUTGRAPH(k, G)
16:    return ⋃_{G_c∈G 内の連結成分} CUTCONNECTED(k, G_c)
17: end function
18: function CUTCONNECTED(k, G)
19:    if |V(G)| < 2k then return {V(G)}
20:    s = 任意に選んだ頂点 (s ∈ V(G))
21:    n ← s から最も遠い頂点 (n ∈ V(G))
22:    N ← ∅                          ▷ 切り出されるグラフと隣接する頂点集合
23:    G_1 ← G                        ▷ 切り出したあと残るグラフ
24:    repeat
25:        G' = G から n を除いたもの
26:        G_1 ← G' から頂点数 k 未満の連結成分を除いたもの
27:        N ← (N ∪ {G において n と隣接する頂点 }) ∩ V(G_1)
28:        n ← N の中で (V(G) − V(G_1)) の重心から最も近い頂点
29:    until |V(G)| − |V(G_1)| ≥ k
30:    if |V(G_1)| = 0 then
31:        return {V(G)}
32:    else
33:        G_2 = G から V(G_1) を除いたもの   ▷ 切り出されるグラフ
34:        return CUTGRAPH(k, G_1) ∪ CUTGRAPH(k, G_2)
35:    end if
36: end function
```

次にグラフの分割について説明する．グラフの分割は MAKEGRAPH 関数で作られたグラフの各連結成分に対して CUTCONNECTED 関数を適用することで行われる（16 行目）．CUTCONNECTED 関数では連結成分から k 頂点以上から成るグループを分離していく．連結成分の頂点数が $2k$ 以上だった場合はグループ形成の始点とする頂点を一つ選び，始点の近傍にある頂点を貪欲に収集しグループを形成する．頂点収集の手法は 3 つの特徴を持つ．1 つ目は，頂点が疎らに散らばって残ることを防ぐため，グループ形成の始点としてグラフの端にある頂点を選択する（20, 21 行目）ことである．始点周辺の頂点は取り除かれるため，端にある頂点を選択しなかった場合，空間中の頂点の密度が低下していく．その結果終盤に形成されるグループは遠い頂点を集めて作られることになり，分散が増大してしまう．2 つ目は，高速化のため近傍点の探索範囲をグループから隣接する頂点に限定している（28 行目）ことである．(k, m)-近傍グラフには k 頂点未満の連結成分が存在しないため，探索範囲を限定しても頂点数 k 以上のグループを必ず作ることができる．3 つ目は，切り出されるグループにクラスタ構造が反映されており，含まれる頂点の数が可変であることである．この性質は頂点の切り出しによって k 頂点以下の連結成分になってしまう頂点を形成中のグループに含めることで実現されている．

アルゴリズムの流れを図 3 を用いて説明する．ただし $m = 2$, $k = 3$ とする．まず MAKEGRAPH 関数により構築される (k, m)-近傍グラフが図 3 (a) である．図 3 (a) の左下の連結成分は $2k$ 頂点未満なのでこれ以上分割されず，そのまま一つのグループとなる．グループとなった頂点は取り除き，図 3 (a) の大きい連結成分の分割だけを考える．グループ形成の始点が右下の頂点になったとすると，最初に最近傍の 2 頂点が収集されグラフから取り除かれる（図 3 (b)．次にその 2 頂点の重心から最も近い頂点を収集する（図 3 (c)）．これにより頂点数 2 の連結成分ができるため，それも同じグループに含める．結果として図 3 (c) の右下点線で囲わ

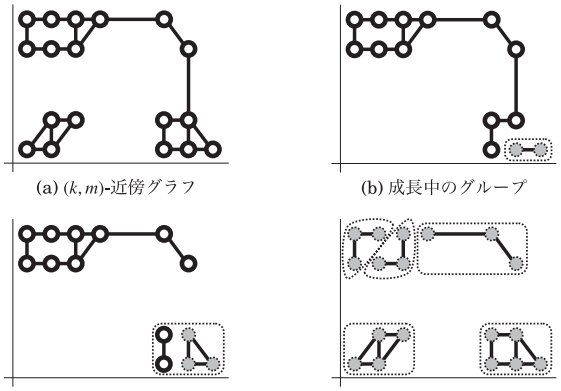


図 3: 友引法の過程

表 3: 匿名化するデータセット

名前	レコード数	属性数
Census [1]	1080	13
EIA [1]	4092	12
Tarragona [1]	834	13
Adult [5]	30162	8
Random（一様分布）	3000	2

れた頂点が一つのグループとなる．同様にしてグループに含まれない頂点がなくなるまでグループの作成を繰り返す．最終的に作成される k-分割の一例は図 3 (d) である．グループ形成の始点がどのように選ばれるかによって分割結果は変化するため，結果は一意に定まらない．

図 3 (c) に見られるように切り出した頂点と接続されている頂点も一緒に切り出されることから，我々はこのアルゴリズムを友引法と呼んでいる．

5. 評価

実際にデータセットを匿名化し，空間分割の併用と友引法の効果を情報損失及び処理速度の観点で既存手法と比較した．実験に使用したマシンの CPU は Intel Core i7-4770K 3.50GHz，メモリは 32.0GB である．実験用プログラムの実装には Haskell を用いた．

実験に使用した統計データを表 3 に示す．Census, EIA, Tarragona はミクロアグリゲーションの，Adult は k-匿名化の既存研究においてそれぞれデファクトスタンダードのデータセットである [4,6,10,18,20–23,26]．Census と Tarragona は配布データに含まれる全ての属性を用いた．EIA からは UTILITYID, UTILNAME, YEAR の 3 属性を除いた．UTILITYID と UTILNAME は個人（団体）を特定する一意な識別子であり，YEAR は全レコードで同一の値を持つためである．Adult は既存研究 [4,18,22] に倣い age, work class, education, marital status, occupation, race, gender, native country の 8 属性を用いた．カテゴリ値には適当な数値表現を与えた上で，各データセットのそれぞれの属性について，データセット中に登場する値の最大値が 1.0，最小値が 0.0 となるように正規化した．値の範囲が大きいレコードがあるとその属性がレコード間距離の算出において支配的になってしまうためである．

比較対象のアルゴリズムには V-MDAV を使用した．V-MDAV はミクロアグリゲーションの研究で手法のベースや比較対象としてしばしば使用されている [7,15,19,25]．さらに友引法と同様生成されるグループの数が可変であるため，比較に適している．孤立したレコードの判定に使用されるパラメータ γ は文献 [26] に準じ 0.2 と 1.1 を用いた．

またグループの大きさ k は実用上 $k = 5$ の利用が多い [11] ことから，実験でも特に指定がない限り $k = 5$ を使用する．

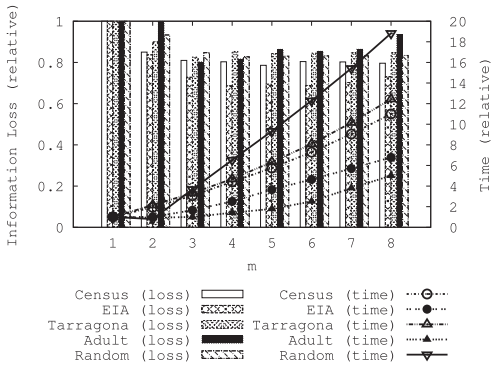


図 4: m に対する情報損失と処理時間

5.1 友引法の評価

まず空間分割を併用せず友引法単体の性能を評価する．

m の最適値

友引法ではパラメータ m を受け取り (k,m)-近傍グラフを構築する．最適な m を探すため m の値を変動させつつ情報損失 (information loss) と実行時間 (time) を調査した（図 4）．情報損失，実行時間共に各データセットについて $m = 1$ での値を 1 とする相対値で表した．相対実行時間は $m \geq 2$ においていずれのデータセットでも単調に増加している．m を大きくするにつれてノードの平均次数が増大し，グループに追加する頂点（アルゴリズム 2, CUTCONNECT 関数内の変数 n）の探索にかかる時間が増すためである．一方，相対情報損失は $m = 3$ までに大きく低下し，そこからは小幅の増減が続く．全データセットで最小の情報損失を示す m は見つからないものの，m を大きくするにつれ実行時間が増大することも踏まえると $m = 3$ が汎用的なパラメータとして利用できる．

情報損失

次に，データセット毎にグループの大きさ k を変化させ情報損失の大きさを比較した．最高の性能を調査するため (k,m)-近傍グラフの m はデータセット毎に最適値を用いた．即ち Census は 5, EIA と Random は 4，他は 3 である．結果を図 5 に示す．凡例で V-MDAV に付いた括弧内はパラメータ γ の値である．Tarragona の $k \geq 5$ を除いたすべてのケースにおいて友引法は V-MDAV より小さい情報損失量となった．特に EIA の $k = 3$ では V-MDAV ($\gamma = 0.2$) より約 16%情報損失を低下させている．情報損失量の低下は友引法がレコードの分布に存在したクラスタ構造を分割に反映した結果だと考えられる．

レコード数に対する実行時間の変化

Adult データセットの一部を用い，処理対象のレコード数に対する実行時間の変化を調査した（図 6）．レコード数 n に対し $O\left(n^2\right)$ である V-MDAV と沿うように友引法も実行時間が増加している．このことから友引法の計算量も $O\left(n^2\right)$ であると推測される．

5.2 空間分割併用の評価

次に空間分割と友引法を併用した場合の情報損失と実行時間について調査した．便宜上空間分割を併用する友引法を友引 $\sharp$ と呼ぶことにする．データセットはレコード数が多い EIA と Adult を用いた．空間分割によって作られる空間のレコード数 $k_\sharp$ を 5 から各データセットのレコード数まで変化させた場合と Mondrian のみ用いた場合の結果が図 7 である．$k_\sharp$ を大きくすると徐々にクラスタリングの影響が支配的となり，処理にかかる時間が延びる代わりに情報損失は低下していく様子が図 7 から読み取れる．$k_\sharp$ がデータセットのレコード数と等しい場合は一切空間分割が行われず，全てクラスタリングで処理されていることを意味する．

V-MDAV と友引 $\sharp$ の情報損失量を比較すると，EIA において V-MDAV ($\gamma = 0.2$) が 25.00 秒かかった情報損失量 0.02399 以下

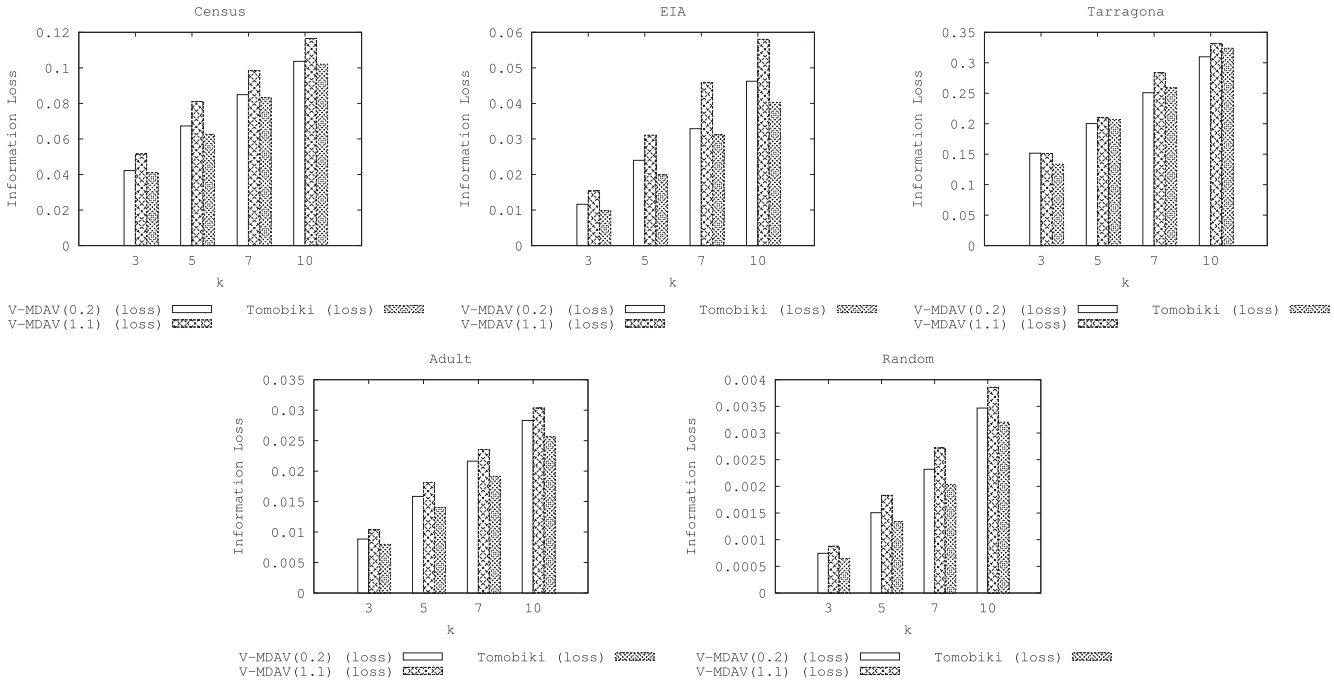


図 5: k に対する情報損失量の変化

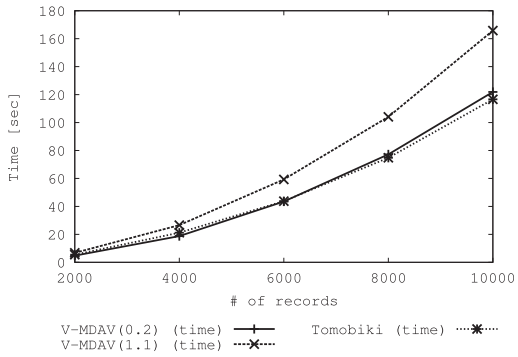


図 6: レコード数に対する処理時間の変化

を反引 ♯ ($k_\sharp = 320$) は 2.465 秒で実現しており，約 10 倍高速である．Adult においても V-MDAV ($\gamma = 0.2$) で 1235.7 秒かかった情報損失量 0.0159 を反引 ♯ ($k_\sharp = 3840$) は 160.2 秒で得ており，およそ 7.7 倍高速化している．

6. 結論

プライバシー保護のため k-匿名化が利用されている．これまで空間分割に基づく手法とクラスタリングに基づく手法が k-匿名化のために提案されてきたが，前者は情報損失が大きく，後者は処理が遅い．そこで我々は 2 つの手法を提案する．一つは空間分割を併用することにより，クラスタリングに基づく手法が持つ情報損失の少なさを保ちつつ匿名化処理を高速化する手法である．もう一つは，最初にレコードを頂点とするグラフを構築することでクラスタ構造を捉える新しい k-分割アルゴリズムである．実際の統計データを用いた実験により，この分割アルゴリズムは既存手法と比べ最大 16%情報損失を低下させることが分かった．さらに空間分割を併用することで既存手法と同等の情報損失のデータを約 10 倍高速に得ることができた．

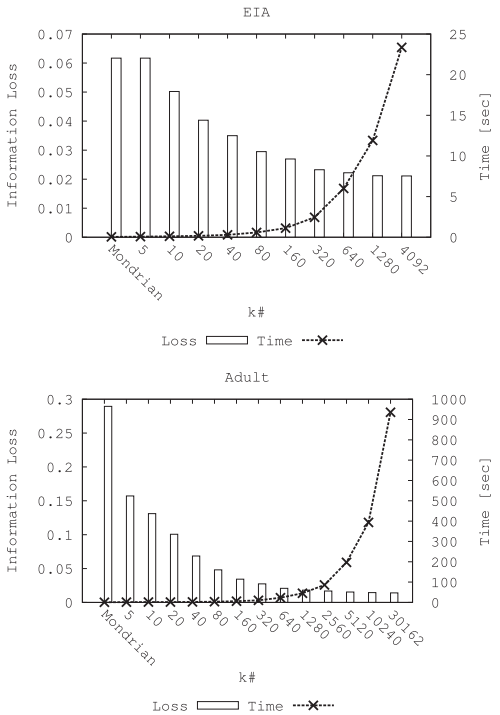


図 7: $k_\sharp$ 対する情報損失と処理時間の変化

[文献]

[1] Statistical Disclosure Control - Testsets. http://neon.vb.cbs.nl/casc/CASCtestsets.htm.

[2] J. D.-f. Anna Oganian. On the Complexity of Optimal Microaggregation for Statistical Disclosure Control.

[3] L. Arge. The buffer tree: A new technique for optimal

I/O-algorithms. In S. G. Akl, F. Dehne, J.-R. Sack, and N. Santoro eds., *Algorithms and Data Structures*, Vol. 955 of *Lecture Notes in Computer Science*, pp. 334–345. Springer Berlin Heidelberg, Berlin, Heidelberg, Jan. 1995.

[4] R. Bayardo and R. Agrawal. Data Privacy through Optimal k-Anonymization. In *21st International Conference on Data Engineering (ICDE'05)*, pp. 217–228. IEEE, 2005.

[5] C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases, 1998.

[6] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-Anonymization Using Clustering Techniques. In R. Kotagiri, P. Krishna, M. Mohania, and E. Nantajeewarawat eds., *Advances in Databases: Concepts, Systems and Applications SE - 18*, Vol. 4443 of *Lecture Notes in Computer Science*, pp. 188–200. Springer Berlin Heidelberg, 2007.

[7] S. K. Chettri and B. Borah. MDAV2K: a variable-size microaggregation technique for privacy preservation. In *International conference on information technology convergence and services, In*, pp. 105–118, 2012.

[8] L. Cox, B. Johnson, S.-K. McDonald, D. Nelson, and V. Vazquez. Confidentiality issues at the Census Bureau. In *Proceedings of the First Annaul Census Bureau Research Conference, Washington, DC: US Government Printing Office*, pp. 199–218, 1985.

[9] J. Domingo-Ferrer and J. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189–201, 2002.

[10] J. Domingo-Ferrer and V. Torra. Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, Aug. 2005.

[11] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association : JAMIA*, 16(5):670–82, Jan. 2009.

[12] J. H. Freidman, J. L. Bentley, and R. A. Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3):209–226, Sept. 1977.

[13] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data - SIGMOD '84*, Vol. 14, p. 47, New York, New York, USA, June 1984. ACM Press.

[14] X. He, H. Chen, Y. Chen, Y. Dong, P. Wang, and Z. Huang. Clustering-Based k-Anonymity. In P.-N. Tan, S. Chawla, C. Ho, and J. Bailey eds., *Advances in Knowledge Discovery and Data Mining SE - 34*, Vol. 7301 of *Lecture Notes in Computer Science*, pp. 405–417. Springer Berlin Heidelberg, 2012.

[15] K. L. Huang, S. S. Kanhere, and W. Hu. Towards privacy-sensitive participatory sensing. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pp. 1–6. IEEE, Mar. 2009.

[16] A. Hundepool, A. van de Wetering, R. Ramaswamy, L. Franconi, S. Polettini, A. Capobianchi, P.-P. de Wolf, J. Domingo, V. Torra, R. Brand, and S. Giessing. - ARGUS version 4.2 User ' s Manual, 2008.

[17] T. Iwuchukwu and J. F. Naughton. K-anonymization As Spatial Indexing: Toward Scalable and Incremental Anonymization. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, pp. 746–757. VLDB Endowment, 2007.

[18] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, p. 279, New York, New York, USA, July 2002. ACM Press.

[19] H. Jian-min, C. Ting-ting, and Y. Hui-qun. An Improved V-MDAV Algorithm for l-Diversity. In *2008 International Symposiums on Information Processing*, pp. 733–739. IEEE, May 2008.

[20] M. Laszlo and S. Mukherjee. Minimum spanning tree partitioning algorithm for microaggregation. *IEEE Transactions on Knowledge and Data Engineering*, 17(7):902–911, July 2005.

[21] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient Full-domain K-anonymity. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pp. 49–60, New York, NY, USA, 2005. ACM.

[22] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian Multidimensional K-Anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pp. 25–25. IEEE, Apr. 2006.

[23] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, 2007.

[24] J.-L. Lin and M.-C. Wei. An Efficient Clustering Method for K-anonymization. In *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society*, PAIS '08, pp. 46–50, New York, NY, USA, 2008. ACM.

[25] J.-L. Lin, T.-H. Wen, J.-C. Hsieh, and P.-C. Chang. Density-based microaggregation for statistical disclosure control. *Expert Systems with Applications*, 37(4):3256–3263, Apr. 2010.

[26] A. Solanas and A. Martinez-Balleste. V-MDAV: A Multivariate Microaggregation With Variable Group Size. In *17th COMPSTAT Symposium of the IASC*, Rome, 2006.

[27] L. Sweeney. K-anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

新井 淳也 **Junya ARAI**

2011 年東京大学理学部情報科学科卒業．2013 年同大学院情報理工学系研究科コンピュータ科学専攻修士課程修了．同年，日本電信電話株式会社入社．現在，日本電信電話株式会社 NTT ソフトウェアイノベーションセンタ社員．匿名化及び大規模グラフデータの並列分散処理に関する研究開発に従事．日本データベース学会，ACM 各会員．

鬼塚 真 **Makoto ONIZUKA**

1991 年東京工業大学工学部情報工学科卒業．同年，日本電信電話株式会社入社．2000〜01 年ワシントン大学客員研究員，2010〜14 年日本電信電話株式会社特別研究員，2012〜14 年電気通信大学客員教授，現在，大阪大学大学院情報科学研究科教授．博士（工学）．大規模グラフデータの分散データ処理に関する研究開発に取り組んでいる．2004 年情報処理学会山下記念賞，2008 年データベース学会上林奨励賞など受賞．情報処理学会，電子情報通信学会，日本データベース学会，ACM 各会員．

塩川 浩昭 **Hiroaki SHIOKAWA**

2009 年筑波大学第三学群情報学類卒業．2011 年同大学院システム情報工学研究科博士前期課程修了．同年，日本電信電話株式会社入社．現在，日本電信電話株式会社 NTT ソフトウェアイノベーションセンタ研究員，および筑波大学大学院システム情報工学研究科博士後期課程在籍．大規模データ分析，分散並列処理の研究開発に従事．2013 年 DEIM Forum 2013 最優秀論文賞および優秀論文賞，2014 年 DEIM Forum 2014 優秀論文賞，日本データベース学会平成 25 年度論文賞受賞．日本データベース学会会員．