# 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

新井　淳也† 　鬼塚　真† 　塩川　浩昭†

† 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
〒 180–8585 東京都武蔵野市緑町 3-9-11
E-mail: †{arai.junya,onizuka.makoto,shiokawa.hiroaki}@lab.ntt.co.jp

**あらまし**　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．$k$-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の集合へと分割する必要がある．その際，データの変換で生じる情報損失を抑えられるようなグループを高速に作成できることが望ましい．これまで空間分割に基づく分割手法とクラスタリングに基づく分割手法が提案されてきたが，これらは高速な処理と低い情報損失を両立できていない．本稿では 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である．2 つ目は，レコードを頂点とするグラフの構築によりクラスタを捉え，情報損失の小さい分割を作成するアルゴリズムである．2 つの手法を組み合わせることで，既存手法より最大 10 倍高速に情報損失の小さい分割を行えることが確認された．

**キーワード**　匿名化, クラスタリング, プライバシー保護

## 1. はじめに

個人情報を含むデータは統計や医療のために多様な組織で収集されている．収集されたデータは分析して知見を得るための利用価値が高い反面，個人のプライバシーを侵害する危険を孕んでいる．そこで，データ内の情報と個人を結び付けられない形へのデータの匿名化が行われる．

ここでは匿名化するデータとして表 1 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，複数の属性を組み合わせることで個人を特定できてしまう場合があるからである． このような他者でも比較的情報を入手しやすく且つ複数組み合わせることで個人を特定できてしまうような属性群を**準識別子** (quasi-identifier) [9] と呼ぶ．

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** [31] 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 を 3-匿名化した結果の例を表 2,3 に示す．表 2 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大きな区分（千葉と東京なら南関東）へ置き換える**一般化** [30] によって $k$-匿名化されている．一方表 3 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, 12, 33] によって $k$-匿名化されている．表 2,3 のどちらも 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．

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．例えば男女の情報が失われ「人」になってしまったり（表 2），逆の性別になってしまったり（表 3 の最後のレコード）する．情報に損失の大きいデータを分析すると，元のデータと異なる結果になってしまう．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，同じ準識別子を与える必要がある．すなわち，$k$-匿名化のためには次のような問題を解かなければならない: 与えられた

表 1　年収データ

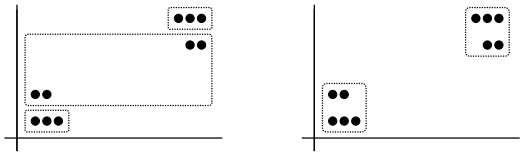
| 名前 | 性別 | 年齢 | 居住地 | 年収 |
|---|---|---|---|---|
| 越谷 | 男 | 24 | 千葉 | 350 万 |
| 加賀山 | 女 | 31 | 埼玉 | 300 万 |
| 宮内 | 女 | 56 | 埼玉 | 1 億 200 万 |
| 石川 | 女 | 36 | 茨城 | 800 万 |
| 一条 | 男 | 31 | 東京 | 400 万 |
| 富士宮 | 女 | 28 | 千葉 | 600 万 |

表 2　一般化により 3-匿名化された年収データ

| 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 |

表 3　ミクロアグリゲーションにより 3-匿名化された年収データ

| 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 |



(a) 分散の大きいグループ　　(b) 理想的なグループ

図 1　既存手法が分散の大きいグループを作る例 ($k = 3$)

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．

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 [11, 19, 29]．最適な $k$-分割の作成は NP 困難である [2, 17, 25] 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 [14] 構築アルゴリズム等の空間分割に基づく手法 [20, 25] とクラスタリングに基づく手法 [6, 11, 19, 23, 27, 29] に分けられる．レコード数 $n$ に対する計算量は前者が $\mathcal{O}(n \log n)$，後者が $\mathcal{O}(n^2)$ である．つまり空間分割に基づく手法のほうが計算量は小さい．診療記録や購買履歴のように複数のレコードが 1 人に対応するデータは人口を超えて際限なく増大し得るため，レコード数に対する計算量の小ささは重要である．一方，クラスタリングのほうが空間分割よりも柔軟に分割を行うことが可能であるため情報損失は小さい [6]．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既存手法はレコードのクラスタ構造を適切に扱えない場合がある．作成されるグループの数が $\lfloor n/k \rfloor$（$n$ はレコード数）に固定されるアルゴリズム [6, 19] はクラスタを無視して分散の大きいグループを作ってしまう (図 1(a))．グループ数が可変である手法 [29] も同様にクラスタを正確に捉えられず図 1(a) のようなグループを作ってしまう．

そこで我々は 2 つの手法を提案する．それらは (i) 空間分割とクラスタリングの併用，並びに (ii) 新しい $k$-分割アルゴリズムの**友引法**である．

1 つ目の提案である空間分割とクラスタリングの併用では 2 段階の分割を行う．即ち，まずレコード数 $n$ に対し $\mathcal{O}(n \log n)$ の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 $\mathcal{O}(n^2)$ のアルゴリズムでさらに細かく分割する．レコード全体の分布を分析する一般的なクラスタリングと異なり，$k$-分割では近傍のレコードでグループを作成できさえすればよい．そのためには局所的なレコードの分布が分かっていれば十分である．従って，クラスタリングの対象が空間分割によって作られた狭い空間内のレコードに限定されても情報損失は減少する．計算量の小さい空間分割を途中まで使用することで $k$-分割処理全体の計算量は $\mathcal{O}(n^2)$ より小さくなるため，空間分割の併用によって高速な処理と低い情報損失を両立できる．

2 つ目の提案である友引法は，ミクロアグリゲーションによる $k$-匿名化のための $k$-分割クラスタリングアルゴリズムである．友引法の特徴はグラフを用いて既存手法より低い情報損失を実現する点にある．グラフは $k$ 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て構築する．友引法は近傍のレコードを貪欲に収集し分散の小さいグループを形成するだけでなく，グラフから発見されるクラスタを考慮して総グループ数を変化させる．これによってより全体最適に近い分割が可能になる．例えば図 1(b) のような分割を作成する．

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実験によって分かった．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．実験では，空間分割と友引法の併用はクラスタリングベースの既存手法のみを用いた場合と同等の情報損失量を約 10 倍高速に実現した．

表 4　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 | クラスタリング | [11, 19, 23, 29], 友引法 | [6, 17, 27] |
| | 空間分割 | — | [20, 25] |

## 2. 関 連 研 究

$k$-匿名化は (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．$k$-分割手法をクラスタリングと空間分割に，準識別子の変換手法をミクロアグリゲーションと一般化に分類し，これまでの研究と本研究を分類したものが表 4 である．それぞれの分類について詳しく説明した後，本研究の位置づけを述べる．

### 2.1 分 割 手 法

#### 2.1.1 空間分割に基づく分割手法

空間分割に基づく手法 [20, 25] ではレコードを多次元空間上の点と見做し，$k$ 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 [14] や R-tree [16] 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．

特徴としてレコード数に対する計算量はクラスタリングに基づく手法より小さく高速だが，情報損失が大きい [6]．図 2 がその理由の一端を示している．$k = 2$ とすると，5 レコードある図 2(a) は 2 グループへ分割できるはずである．しかし実際には図 2(b) で示したいずれの点線で分割してもレコード数 1 のグループが生じるため，分割できない．結果として 5 レコード全体で 1 つのグループとなってしまい，情報損失が増加する．

#### 2.1.2 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レコード間の距離を用いて分割を行う．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, 11, 19, 23, 27, 29]．

クラスタリングに基づく手法はグループ数が固定であるものと可変であるものに分けられる．前者においては Maximum Distance to Average Vector (MDAV) [12, 19] が，後者においては MDAV を基にした Variable-size Maximum Distance to

**Algorithm 1** V-MDAV のレコード追加処理

```
1: function extendGroup(G, R, γ)
2:     while |G| < 2k do
3:         (g, r) = arg min_{(g,r)∈G×R} distance(g, r)
4:         d_in = distance(g, r)
5:         d_out = min_{s∈R−{r}} distance(r, s)
6:         if d_in < γ d_out then
7:             G ← G ∪ {r}
8:             R ← R − {r}
9:         else
10:            break
11:        end if
12:    end while
13:    return (G, R)
14: end function
```

Average Vector (V-MDAV) [29] が代表的である．MDAV はグループの中心とするレコードからユークリッド距離が最も近い $k-1$ レコードを収集して 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，クラスタを無視して分散の大きいグループを作ってしまう場合がある（図 1(a)）．そこで V-MDAV ではアルゴリズム 1 に示される extendGroup 関数でグループにレコードを追加する．extendGroup 関数が受け取る引数 $G$ は MDAV と同様に作られた $k$ レコードから成るグループ，$R$ はまだグループに分割されていないレコードの集合，$\gamma$ は判定処理に利用するために利用者が与えるパラメータである．しかしながら，extendGroup 関数が追加するのは他のレコードが近傍にない孤立したレコードのみである．そのため，図 1(b) のような理想的な分割ではなく MDAV と同様に図 1(a) のような分散の大きいグループを作ってしまう．

### 2.2 準識別子の変換手法

#### 2.2.1 一般化による $k$-匿名化

表 2 のように，数値を値の範囲へ，カテゴリをより大きな概念へ置き換える変換手法は一般化 [30] と呼ばれる．

一般化による $k$-匿名化はさらに**大域的再符号化** (global recoding) と**局所的再符号化** (local recoding) に分けられる [32]．大域的再符号化ではレコードが持つ属性値を一般化する全レコード共通の関数が存在する．他方，局所的再符号化ではそのような関数を定義できない．例えば表 1 から表 2 への変換において，年齢の 31 という値は [24–31] と [31–56] の両方に一般化されている．このことから使用された一般化関数が全レコード共通でないことが分かる．

局所的再符号化は大域的再符号化より柔軟な一般化が可能であるため情報損失は低下する．一方で局所的再符号化による**データ探査問題** (data exploration problem) の発生が Fung らによって指摘されている [15]．例えば一般的なデータマイニング手法では 千葉 ⊂ 南関東 ⊂ 関東 のような包含関係や，表 2 における年齢のように重なりのある関係を解釈することができない．このような分析の困難さをデータ探査問題と称している．

#### 2.2.2 ミクロアグリゲーションによる $k$-匿名化

ミクロアグリゲーションとは同じグループに属する $k$ 人以上のレコードの値を平均値に書き換えることにより個人の特定を防ぐ手法で，統計的開示抑制技術として 1980 年代から存在している [8,33]．元々ミクロアグリゲーションによる値の書き換えはレコードの全属性に対し行われていたが，書き換えを準識別子に限定することによって $k$-匿名化にも利用される [12]．表 1 をミクロアグリゲーションによって 3-匿名化したものが表 3 である．ミクロアグリゲーションには一般化に対して 3 つの利点がある．1 番目に，大域的再符号化による一般化と比べ情報損失が少ない [12] こと．2 番目に，局所的再符号化による一般化で発生するデータ探査問題が発生しないこと．ミクロアグリゲーションは匿名化前のデータにおいても有効な値のみ使用するため，元データと同じアルゴリズムで分析が可能である．3 番目に，数値データの情報がより多く残ること．例として一般化では $\{0,0,0,0,10\}$ も $\{0,10,10,10,10\}$ も [0–10] となってしまい区別できないのに対し，ミクロアグリゲーションではそれぞれ平均値の 2 と 8 になり大小関係が残る．

**Algorithm 2** Mondrian のアルゴリズム

```
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
```

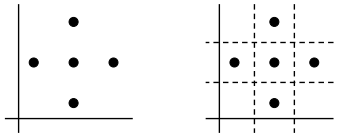
### 2.3 本研究の位置付け

1 つ目の提案手法である空間分割とクラスタリングの併用は $k$-分割そのものを行う手法ではなく，空間分割による分割手法とクラスタリングによる分割手法を組み合わせることで高速かつ情報損失の小さい $k$-分割が可能になるという着想を述べたものである．2 つ目の提案手法である友引法は分割手法としてクラスタリングを，準識別子の変換手法としてミクロアグリゲーションを用いた $k$-分割手法の一つである（表 4）．

## 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 [25] を使用するため，その概要を述べる．Mondrian の疑似コードをアルゴリズム 2 に示した．mondrian 関数はまず引数 *partition* として受け取ったレコードの集合がさらに分割可能であるかどうかを判定する．もし *partition* が分割不可能であれば，これ以上分けられないレコードの集合として $\{partition\}$ を返却する．そうでなければ *partition* をさらに分割するため，分割する次元を決定する．k 個以上のレコードを含むグループを作成できる次元が複数ある場合は，*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）次元を選び（chooseDimension 関数），*dim* と置く．次に次元 *dim* に関して *partition* の中での中央値を探し（findMedian 関数），$splitVal$ と置く．中央値で分割することにより 2 つの分割に含まれるレコード数が均等に近くなる．最後に，次元 *dim* について $splitVal$ より大きいレコード群と小さいレコード群の 2 つのグループを作成し，それぞれについて再帰的に mondrian 関数を適用する．



(a) (b)

図 2　これ以上分割できない状態の例 ($k = 2$)

### 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 SSE/SST が用いられており [2, 11, 19, 29]，本研究もそれに倣う．SSE (sum of squared errors) はグループ内のデータがどの程度似通っている（同質である）かを表し，次のように定義される．

$$SSE = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \overline{x}_i)^2$$

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，$\overline{x}_i$ は $i$ 番目のグループに属するレコード値の平均を表す．SSE が小さいほどグループ内のデータの同質性は高い．さらにグループ間のデータの差異の大きさを SSA (treatment sum of squares) として次のように定義する．

$$SSA = \sum_{i=1}^{g} n_i (\overline{x}_i - \overline{x})^2$$

ただし $\overline{x}$ は匿名化対象レコードすべてのデータの平均である．SSE と SSA の和として SST (total sum of squares) を次のように定義する．

$$SST = SSE + SSA = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - \overline{x})^2$$

最適なクラスタリング結果とは SSE を最小化する（同時に SSA を最大化する）ことなので，$[0, 1]$ に正規化された情報損失は SSE/SST で表される．$k$-分割が最適に近いほど SSE/SST は小さくなる．

## 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

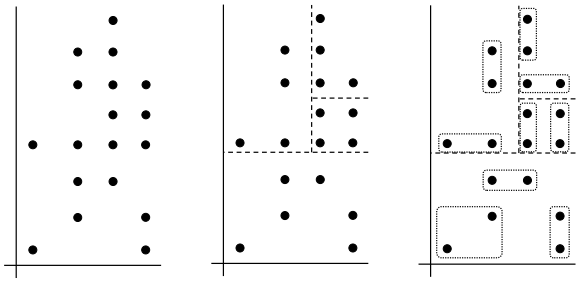
提案手法は 2 段階から成る．

**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．

**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．

本論文では粗粒度分割時の分割手法として Mondrian を用いるが，Iwuchukwu らによる R-tree ベースの手法 [20] 等を用いることもできる．Mondrian を併用して $k$-分割を行う過程の例を図 3 に示した．

細粒度分割ではレコード数 $n$ に対して $\mathcal{O}(n^2)$ 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るグループはなるべく均等な数のレコードを含むことが望ましい．このことは $n$ レコードを $x$ レコードと $y$ レコードに 2 分割するとき $x^2 + y^2$ が $x = y = n/2$ で最小値を取ることからも直感的に分かる．格子状に空間を分割すると分布によってはグループ内のレコード数に偏りが生じてしまうため適さない．



(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 3　空間分割とクラスタリングを併用した匿名化の過程

空間分割を併用する利点は 3 つある．

（1） 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 [13] 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） 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．高速・低情報損失を両立する中でも $k_\sharp$ の大きさによって結果は変化する．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．逆に小さい $k_\sharp$ 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） 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, 10] 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 [20]．

### 4.2　友 引 法

空間分割に加え，より情報損失の少ないミクロアグリゲーションアルゴリズムである友引クラスタリングアルゴリズムを提案する．アルゴリズムの疑似コードをアルゴリズム 3 に示す．tomobiki 関数の引数は，$m$ が後述するグラフで使用される定数，$k$ 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．

まず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もあるが，ここではすべての連結成分が $k$ 頂点以上含むグラフを構築する．そのようなグラフを構築するために，頂点数 $k$ 未満の連結成分 $G_c$ に属するレコードと $G_c$ に属さないレコード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．初期状態としては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）．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た上，各頂点について近傍の点をすぐに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 3 に記述していない．

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（cutGraph 関数）．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,「虫食い状」に頂点が残ることを防ぐため，グループ形成の始点になるべくグラフの端にある頂点を選ぶこと（20, 21 行目）．端にあるレコードから収集しなかった場合，分割が進行するにつれレコードの分布が虫に食われた葉のように隙間だらけになる．すると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こと（28 行目）．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．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 4 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 4(a) である．図 4(a) の左上の連結成分は $2k$ 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 4(a) の大きい連結成分の分割だけを考える．

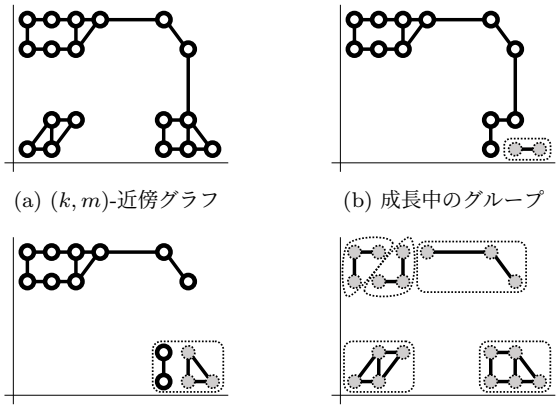
グループ形成の始点が右下の頂点になったとすると，最初に最近傍の 2 頂点が収集されグラフから取り除かれる（図 4(b)．次にその 2 頂点の重心から最も近い頂点を収集する（図 4(c)）．これにより頂点数 2 の連結成分ができるため，それも同じグループに含める．結果として図 4(c) の右下点線で囲われた頂点が一つのグループとなる．同様にしてグループに含まれない頂点がなくなるまでグループの作成を繰り返す．最終的に作成される $k$-分割の一例は図 4(d) である．グループ形成の始点がどのように選ばれるかによって分割結果は変化するため，結果は一意に定まらない．図 4(b) に見られるように切り出した頂点と接続されている頂点（友人）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 PC 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

**Algorithm 3** 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
```



(a) $(k,m)$-近傍グラフ　(b) 成長中のグループ

(c) グループ完成　(d) 最終的にできる $k$-分割

図 4　友引法の過程

を用いた．実験に使用した統計データを表 5 に示す．Census, EIA, Tarragona はマイ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 [4,6,12,21,23–26,29]． 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, 21, 25] 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

表 5　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 |

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 [7,18,22,28]．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．グループの大きさ $k$ は実用上 $k=5$ の利用が多い [13] ことから，実験でも特に指定がない限り $k=5$ を使用する．

### 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．

#### 5.1.1　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と実行時間を調査した（図 5）．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 $m \geqq 2$ 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 3，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．

#### 5.1.2　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．文献 [29] に準じ，比較対象とする V-MDAV に与えるパラメータ $\gamma$ は 0.2 と 1.1 を用いた．ま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 6 に示す．Tarragona の $k \geqq 5$ 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．

#### 5.1.3　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情報損失と実行時間の変化を調査した（図 7）．レコード数 $n$ に対し $\mathcal{O}\left(n^2\right)$ 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 $\mathcal{O}\left(n^2\right)$ であると推測される．

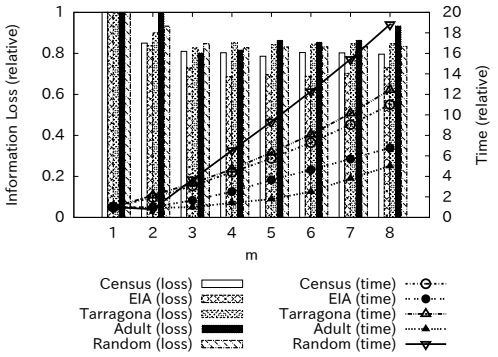



図 5　m に対する情報損失と処理時間

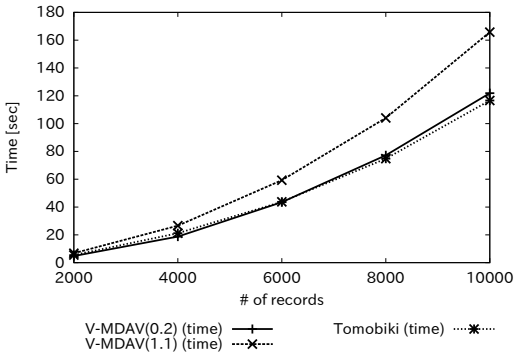



図 7　レコード数に対する情報損失と処理時間の変化

### 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 8 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 8 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る．Mondrian のみ用いた場合，図 2 のように $2k$ レコード以上含むグループが作られ得る．Adult の $k_\sharp=5$ のとき Mondrian より情報損失が少ないのはそのようなグループをさらに分割できるためである．

さらに細かい数値での比較のため特徴的な点の値を表 6 にまとめた．EIA（表 6(a)）では V-MDAV ($\gamma=0.2$) で 25.00 秒かかった情報損失量 0.02399 以下を友引 $\sharp(k_\sharp=320)$ は 2.465 秒で実現しており，約 10 倍高速である．Adult（表 6(b)）においても V-MDAV ($\gamma=0.2$) で 1235.7 秒かかった情報損失量を友引 $\sharp(k_\sharp=3840)$ は 160.2 秒で得ており，およそ 7.7 倍高速化している．また Adult で Mondrian と友引 $\sharp(k_\sharp=5)$ を比較すると，実行時間が 2.5 倍になった一方で情報損失は 46%減少している．

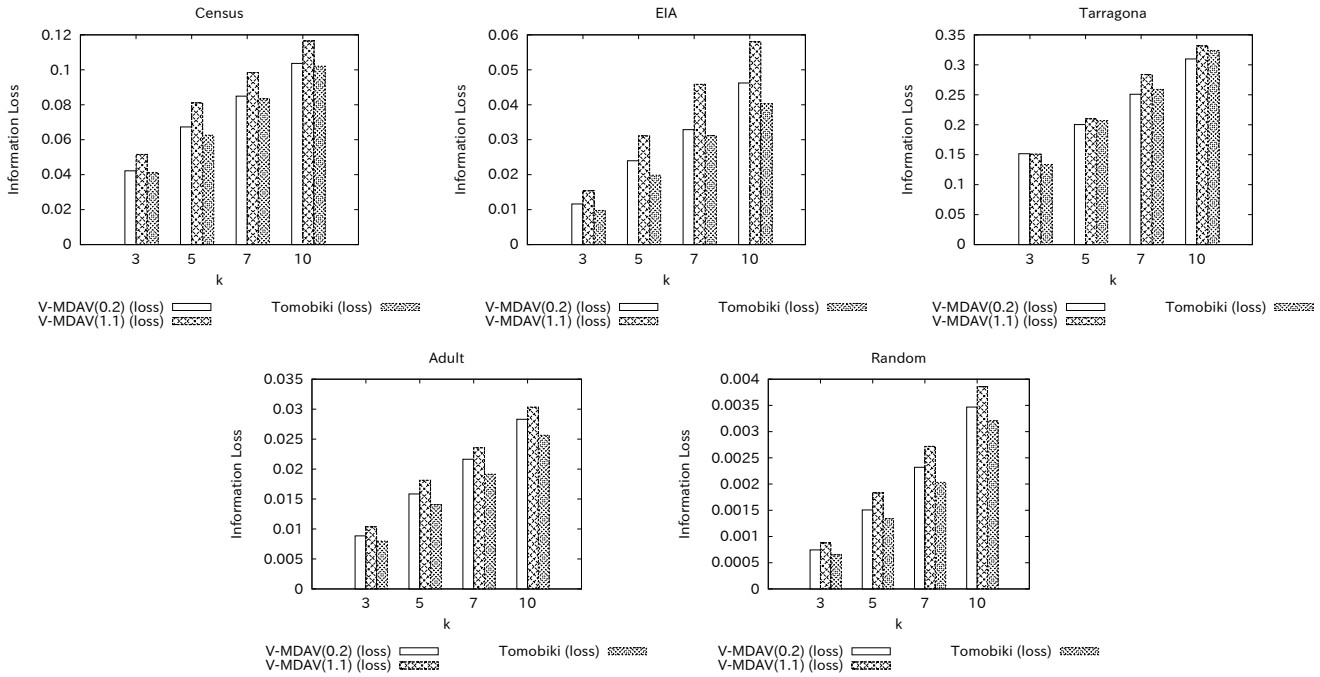



図 6　$k$ に対する情報損失量の変化

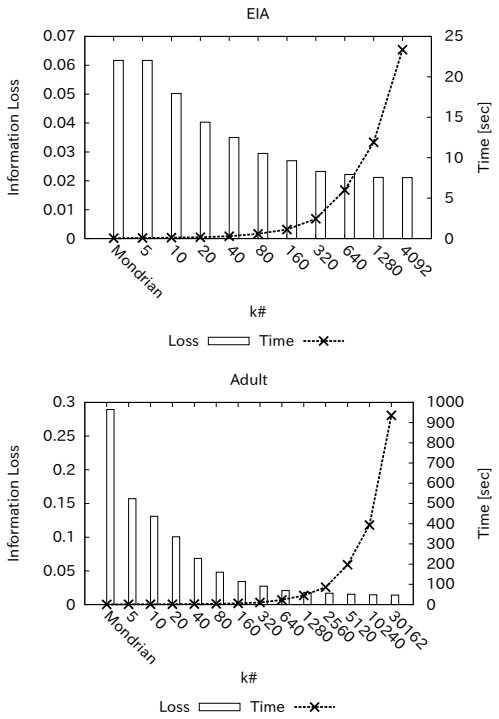



図 8　$k_\sharp$ 対する情報損失と処理時間の変化

表 6　友引 ♯ と V-MDAV の比較

(a) EIA データセット

| アルゴリズム | 情報損失 | 実行時間 [秒] |
|---|---|---|
| 友引 ♯ ($k_\sharp = 5$) | 0.06169 | 0.07755 |
| 友引 ♯ ($k_\sharp = 320$) | 0.02325 | 2.465 |
| 友引 ♯ ($k_\sharp = 4092$) | 0.02111 | 23.49 |
| Mondrian | 0.06169 | 0.03996 |
| V-MDAV ($\gamma = 0.2$) | 0.02399 | 25.00 |
| V-MDAV ($\gamma = 1.1$) | 0.03108 | 39.98 |

(b) Adult データセット

| アルゴリズム | 情報損失 | 実行時間 [秒] |
|---|---|---|
| 友引 ♯ ($k_\sharp = 5$) | 0.1572 | 1.669 |
| 友引 ♯ ($k_\sharp = 3840$) | 0.01586 | 160.2 |
| 友引 ♯ ($k_\sharp = 30162$) | 0.01405 | 918.0 |
| Mondrian | 0.2895 | 0.6574 |
| V-MDAV ($\gamma = 0.2$) | 0.01585 | 1236 |
| V-MDAV ($\gamma = 1.1$) | 0.01813 | 1538 |

## 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．これまで空間分割ベースの手法とクラスタリングベースの手法が $k$-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，クラスタリングベースの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．

将来の課題としてはパラメータフリー化が挙げられる．空間分割の併用では $k_\sharp$ をパラメータとして指定する必要がある．また，友引法で使用する $(k, m)$-近傍グラフの $m$ や V-MDAV の $\gamma$ のように，グループ内のレコード数が可変であるような $k$-分割アルゴリズムはパラメータを持っている．最適なパラメータを推定する技術や，パラメータを受け取らずに多様なレコード分布のデータに対応できるアルゴリズムが必要である．

## 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] T. Dalenius. Finding a Needle In a Haystack or Identifying Anonymous Census Records. *Journal of Official Statistics*, 2(3):329 – 336, 1986.

[10] J. V. den Bercken, B. Seeger, and P. Widmayer. A Generic Approach to Bulk Loading Multidimensional Index Structures. In *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, pp. 406–415, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving Data Publishing: A Survey of Recent Developments. *ACM Comput. Surv.*, 42(4):14:1—–14:53, 2010.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

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

[30] L. Sweeney. Achieving K-anonymity Privacy Protection Using Generalization and Suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

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

[32] L. Willenborg and T. de Waal. *Elements of Statistical Disclosure Control*. Springer-Verlag, 2000.

[33] M. K. Wolf. Microaggregation and disclosure avoidance for economic establishment data. In *annual meeting of the American Statistical Association, New Orleans, Louisiana*, 1988.