DEWS2007 A7-10

k-匿名性を利用したデータ一般化によるプライバシー保護

村本　俊祐†　　上土井陽子††　　若林　真一††

広島市立大学 情報科学部 〒 731–3194 広島市安佐南区大塚東三丁目 4–1
E-mail: †shun@lcl.ce.hiroshima-cu.ac.jp, ††{yoko,wakaba}@ce.hiroshima-cu.ac.jp

あらまし　本稿ではデータベース上の入力データテーブルにおいてデータの一般化を行うことにより，このテーブルに k-匿名性という性質を保持させるプライバシー保護技法について考察する．従来よりデータテーブルに k-匿名性を保持させるようにデータテーブルを変換するアルゴリズムは開発されていたが，元のデータテーブルに対するデータ歪曲度（元のデータテーブルとのデータ値の変化の度合い）が高い場合があったり，発見的手法を取り入れていた結果，満足なプライバシー保護が行われていなかった．本稿では，それらの従来手法の問題点を解消するため k-匿名性を保持し，尚且つデータ歪曲度の小さい結果テーブルを出力することを目的としたアルゴリズムを提案し，シミュレーション実験によりその有効性を検証する．また，アルゴリズム開発において重要となるデータテーブルにおけるデータ値と一般化についての関係や一般化を行う関数等を *Swenney* による先行研究を参考にし形式的に定義する．

キーワード　データテーブル，プライバシー保護，一般化，k-匿名性

Privacy Protection by Data Generalization to Achieve k-Anonymity

Shunsuke MURAMOTO†, Yoko KAMIDOI††, and Shin'ichi WAKABAYASHI††

Faculty of Information Sciences, Hiroshima City University
3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan
E-mail: †shun@lcl.ce.hiroshima-cu.ac.jp, ††{yoko,wakaba}@ce.hiroshima-cu.ac.jp

Abstract　In this paper, we consider a privacy protection technique to convert an input data table in a database into one maintaining k-anonymity by generalizing data. There have been previous methods to convert a data table so as to maintain k-anonymity. However, when compared a resultant data table with an original data table, there were cases, in which a degree of data distortion (a degree of difference between original data and result data) of a resultant table was high. Additionally, introduing heuristic techniques into these methods results in unsatisfactory privacy protection. In order to resolve those weak points, we develop an algorithm that can output a resultant table that maintains k-anonymity and has a small data distortion degree. Moreover, we formaly define several concepts and terminologies concerning with generalizations of data based on the earlier work by *Swenney*.

Key words　data table, privacy protection, generalization, k-anonymity

1. はじめに

統計調査や医療によって得られたデータで，かつ集計されるまえの個票データ (マイクロデータ) は分析者がそれぞれ独自の視点で再分析可能であることから一般に高い価値を持つ．マイクロデータに対するプライバシー保護の簡単な方法に重要な識別情報 (名前など) を非公開にする方法がある．しかし，ただ単に識別情報を非公開にしただけではデータテーブル数個を組み合わせることによって非公開のデータ項目が推測できる可能性がある．データ項目の推測を防ぐために，データテーブルに k-匿名性を持たせることが考えられている [3]．従来手法 [1] [2] では k-匿名性保持のためのデータ操作で結果データを過度に歪曲したり，確実な推測防止が保証でき

ないという欠点があった．本研究ではそれらの欠点の克服を目的として新しいプライバシー保護アルゴリズムを提案し，評価する．

2. 定　　義

本稿で使用する各種用語を以下に定義する．また，本章の定義は文献 [3] を参考にして定めた．文献 [3] における一般化関数や k-匿名性の定義は文章によるものであったので，本稿では形式的に数式を用いて再定義した．それに伴い，本稿で対象とするデータテーブルや属性，属性の値と一般化関数の関係を見直し，同じく再定義した．

2.1 データテーブル

本稿で考慮するデータテーブルは，縦にタプル，横にフィールドを取るテーブルとする．各々のフィールドを属性と呼び，実際に保持するデータ値の意味カテゴリーとする．本論文ではタプル，属性がそれぞれ有限個のテーブルについて考慮する．

$T(A1, A2, ..., AM)$ を属性 $A1, A2, ..., AM$ を持つテーブルを表す．また，属性集合 $\{Ai, ..., Aj\} \subseteq \{A1, ..., AM\}$ における T の部分テーブルを $T[Ai, ..., Aj]$ で表す．図 1 のようなテーブルを考える．このテーブルを T とおくと，T は N 個のタプルの集合として以下のように表現できる．

$$T(A1, A2, ..., AM) = \{t1, t2, t3, ..., tN\}$$

2.2 属　　性

一般にデータテーブルを構成する属性の値には個々に意味がある．その属性が持つ値が直接個人を特定できる属性，例えば名前，電話番号，指紋情報などを識別子と呼ぶ．識別子でない属性であっても他の識別子でない属性と情報を組み合わせることによって識別子と同じ働きをする可能性がある．組み合わせることで直接，個人情報につながる情報が得られる可能性があるという意味から，このような属性を準識別子 (Quasi-identifier:QI) と呼ぶ．

2.2.1 属性一般化階層 DGH (Domain Generalization Hierachies)

本論文では，各属性の値を一般化することでテーブルの k-匿名性保持の達成を目標にしている．このため，各属性には一般化されてない値，一般化された値，さらに一般化された値などの様々な一般化のレベルにある値が含まれる．

属性一般化階層 DGH とは，属性の値を一般化の度合いにより集合に分け，またその集合同士の一般化による関係を階層的に示したものであり，階層的に分けられた属性に属する値がどのような値に一般化されるかを表す一般化関数の集合である．例えば図 3 の DGH を例に挙げると属性 $A4$ に関する一般化階層 DGH は，図中の一般化関数 f を用いて次のように記述できる．

$$DGH_{A4} = \{f_{A4,1}, f_{A4,2}, f_{A4,3}\}$$

一般的に n_i 階層まで一般化できる属性 Ai の DGH_{Ai} は

$$DGH_{Ai} = \{f_{Ai,1}, f_{Ai,2}, ..., f_{Ai,n_i}\}$$

と表す．

また各属性 A は一般化されてない値と一般化された値の集合の集合として定義する．$A1(Race)$ を例に挙げると $A1$ は図 2 のような領域一般化階層 DGH を持っているとき，以下により表される．

$$A1 = \{A1_1, A1_2, A1_3\}$$

また，ここでの属性一般化階層の例において数値データは下一桁から一般化を行っている．しかし，属性一般化階層や次にあげる値一般化階層は基本的にデータ管理者の視点から任意に作成でき，例えば数値データの場合では任意の 1 箇所の桁が隠された状態を 1 回一般化された階層，任意の 2 箇所の桁が隠された状態を 2 回一般化された階層など，と定めることも可能である．同様に数値でなく，値そのものが意味を持つ単語の場合も同じように，データ管理者が一般化された値及び一般化階層を作成する必要がある．よって最大限にデータテーブルを効率的に一般化させるためには，データテーブルを提供するとともに，各属性独自の一般化階層が別途必要となる．

2.2.2 値一般化階層 VGH (Value Generalization Hierachies)

各領域内の値の一般化の関係を値一般化階層 VGH と呼ぶ．各属性 Ai の値一般化階層 VGH_{A_i} は木として表現される．属性 A_i において最大一般化されたもの（それ以上一般化できない値の階層）をこの木の根（root）とする．また，この木は各値を節点とし，すべての節点ペア v_i, v_j において DGH_{A_i} に属する一般化関数 f によって $f(v_i) = v_j$ のとき，そのときに限り v_i の親が v_j であるという条件を満たす木である．

	A1 Race	A2 BirthDate	A3 Gender	A4 ZIP	A5 ID
t1	black	1964	male	02138	00101
t2	black	1964	female	02139	00204
t3	black	1967	female	02138	00225
.					
.					
.					
tn	white	1965	male	02141	00105

図 1　入力テーブルの例

2.3 一般化関数

本節では，属性一般化階層 DGH と値一般化階層 VGH の定義に基づいて値，タプル，テーブルの一般化を階層的に定義する．

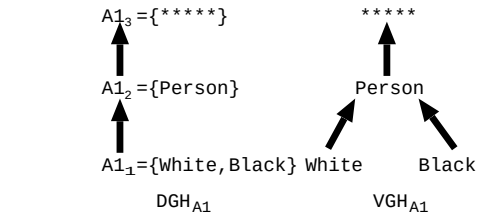

図 2　属性 $A1(Race)$ の属性一般化階層 DGH と値一般化階層 VGH

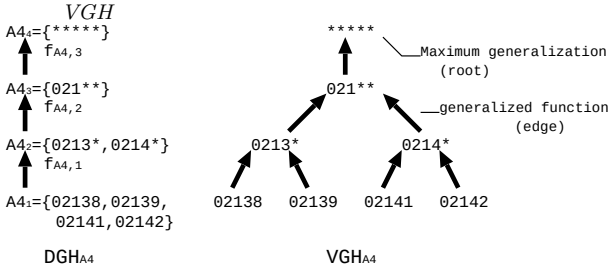

図 3　属性 $A4(ZIP)$ の属性一般化階層 DGH と値一般化階層 VGH

2.3.1　値一般化関数

属性 A は n_i 階層からなり，その属性一般化階層は $DGH_A = \{f_{A,1}, ..., f_{A,n_i}\}$ で表されると仮定する．ある $i, j (0 \leqq i \leqq j \leqq n_i)$ の組において以下を満たす関数 f_A を属性 A に対する値一般化関数とする．

$$f_A = f_{A,i} \circ ... \circ f_{A,j}$$

このとき，f_A は以下を満たす．

$$f_A : A_i \rightarrow A_{j+1}$$
$$A_i, A_{j+1} \in A,\ 1 \leqq i \leqq j \leqq n_i$$
$$\forall v \in A_i,\ \exists v' \in A_{j+1}[f_A(v) = v'] = 1$$

2.3.2　タプル一般化関数

属性 $A1, ..., AM$ においてタプル $t(A1, ..., AM) = (v1, v2, ..., vM)$，ここで各 $vz (1 \leqq z \leqq M)$ は $vz \in Az_j \in Az$ を満たす値とするとき，各属性 $A1, ..., AM$ において以下を満たす関数 F_t をタプル t に関する一般化関数とする．

$$F_t : \{(v1, v2, ..., vM)\} \rightarrow A1_{y_1} \times A2_{y_2} \times ... \times AM_{y_M}$$

ここで，各 $vi (1 \leqq i \leqq M)$ において $vi \in Ai_{x_i}$ を満たす x_i と y_i は $1 \leqq x_i \leqq y_i \leqq n_i$ を満たす．

このとき，F_t は以下を満たすとする．

$$F_t(v1, ..., vM) = (f_{A1}(v1), ..., f_{AM}(vM))$$

ここで，各 $i (1 \leqq i \leqq M)$ に関する $f_{Ai} : Ai_{xi} \rightarrow Ai_{yi}$ は値一般化関数である．

2.3.3　テーブル一般化関数

属性 $A1, A2, ..., AM$ において，テーブル $T(A1, ..., AM) = \{t1, t2, ..., tN\}$，ここで $ti (1 \leqq i \leqq N)$ はタプルとしたときに以下を満たすタプル一般化関数の集合である関数 g_T をテーブル T に対するテーブル一般化関数とする．

$$g_T : \{T(A1, A2, ..., AM)\} \rightarrow (A1' \times A2' \times ... \times AM')^N$$

ここで $Ai' = \bigcup_{A \in Ai} A\ (1 \leqq i \leqq M)$ とする．

このとき，g_T は以下を満たすとする．

$$g_T(t1, t2, ..., tN) = \{F_{t1}(t1), ..., F_{tN}(tN)\}$$

各 $j (1 \leqq j \leqq N)$ に関する F_{tj} はタプル一般化関数である．

2.4　*k*-匿 名 性

テーブル一般化関数を用いて

$$g_T(T) = g_T(t1, t2, ..., tN) = \{F_{t1}(t1), ..., F_{tN}(tN)\} = T'$$

のようにテーブル T との関係を表現できるテーブル T' が与えられたとき，以下の条件を満たすテーブル T' はテーブル T に対し k-匿名性を保持しているという．

$$\forall ti \in T[|\{tj : F_{ti}(ti) = F_{tj}(tj) \wedge tj \in T\}| \geqq k]$$

k-匿名性を保持しているテーブルの例をあげる．図 4(a) のテーブル T と，その一般化テーブルの 1 つである図 4(b) のテーブル T' を考える．図 4(a) の初期テーブル T は $t1$ と $t2$ は全属性の値が一致しているので，この 2 つのタプルの非公開データはデータ組合せによる特定は防止されていると言える．しかしそれ以外のタプルはどれも他のタプルと一致していない独立状態である．次に図 4(b) の一般化テーブル T' に注目すると一般化により $t3'$, $t5'$, $t7'$ の 3 つのタプルと $t4'$, $t6'$ の 2 つのタプルはそれぞれ全属性の値が一致したタプルに変換されている．全てのタプルが自分を含め 2 個以上のタプルと一致している．このとき，テーブル T' は 2-匿名性を満たしているという．

	Race	BirthDate	Gender	ZIP
t1	Black	1964	f	02138
t2	Black	1964	f	02138
t3	Black	1967	m	02141
t4	White	1971	f	02139
t5	White	1967	m	02141
t6	White	1971	m	02139
t7	White	1965	m	02141

(a) 初期テーブル PT

	Race	BirthDate	Gender	ZIP
t1′	Black	1964	f	02138
t2′	Black	1964	f	02138
t3′	Person	196*	m	02141
t4′	White	1971	*	02139
t5′	Person	196*	m	02141
t6′	White	1971	*	02139
t7′	Person	196*	m	02141

(b) 一般化テーブル RT

図 4　一般化による 2-匿名性を満たすテーブルへの変換

3 章で詳しく説明するが，従来手法の *Datafly*，本稿で提案するアルゴリズムともう一つの従来手法 *μ-Argus* では同一値を持つタプルの判定の定義が異なる．従来手法の *Datafly*，本研究で提案するアルゴリズムにおける同一タプル判定の定義においては，上記のように全属性の値の組合せが一致しているタプルを同一としている．

2.5 k-極小一般化

あるテーブルに対し，k-匿名性を保持している一般化テーブルは一般的には複数ある．その複数の一般化されたテーブルを比較するため，極小一般化という概念が取り入れられている [3].

2.5.1 k-極小一般化の定義

$T_l(A0,...,AM)$，$T_m(A0,...,AM)$ をテーブルとする．このテーブルにおける準識別子を $QI_T = \{Ai,...,Aj\} \subseteqq \{A0,...,AM\}$ とする．このとき，テーブル一般化関数 $g_{T_{l_1}}$ によって $T_m[QI_T] = g_{T_{l_1}}(T_l[QI_T])$ のようにテーブル T_l と T_m の一般化の関係が示されるとする．このとき，

（1） T_m は QI_T において k-匿名性を保持している．

（2） $T_z(A0,...,AM)$ をテーブルとする．テーブル一般化関数 $g_{T_{l_2}}$，g_{T_z} によって $g_{T_{l_2}}(T_l[QI_T])$=$T_z[QI_T]$, $g_{T_z}(T_z[QI_T])$=$T_m[QI_T]$ の関係がそれぞれ成り立つような k-匿名性を保持しているテーブル T_z が存在しない．

以上の 2 つを満たしているときに限りテーブル T_m は準識別子 QI_T 上において k-匿名性を保持しているテーブル T_l の極小一般化であると定義する．

2.6 k-極小歪曲

k-極小一般化では k-匿名性を保持している複数のテーブルにおいて，その複数のテーブルがテーブル一般化関数によって直接に一般化の関係が成り立っているときには比較ができるが，直接には一般化の関係が成り立ってないときには比較ができないという欠点がある．したがって本稿では次にあげるデータ歪曲度（データ操作が行われたデータが元のデータに比べてどの程度変化したかという度合いを数値化したもの）を算出する式 DIS を使ってテーブル同士を比較することを提案する．データ歪曲度算出式 DIS は文献 [3] において定義されていたデータ精度算出式 $Prec$ を参考にし，本稿で定義した用語を用いて再定義した関数である．またデータ歪曲度算出式 DIS と極小一般化の概念を組み合わせて，極小歪曲を定義する．

2.6.1 データ歪曲度算出式 DIS

データ歪曲度を表現するのに以下の式を使う．

$PT(A1,...,AM), RT(A1,...,AM)$ をテーブルとし，それぞれのタプルを $t \in PT = \{t1,...,tN\}$, $t' \in RT = \{t'1,...,t'N\}$ とする．ここで $t(Ai)$ によりタプル t の属性 Ai の値を示すものとする．

このテーブルにおける準識別子を $QI_T = \{Ai,...,Aj\} \subseteqq \{A0,...,AM\}$ とする．このときテーブル一般化関数 g_{PT} によって $RT[QI_T] = g_{PT}(PT[QI_T])$ が成り立つとする．一般化後のテーブル RT の一般化前のテーブル PT に対するデータ歪曲度算出式 DIS を以下に定義する．ここで，式中の関数 $h(Tree, value)$ は第一引数 $Tree$ に対象となる木を，第二引数 $value$ に対象となるデータ値をそれぞれ入力することで，その木 $Tree$ における根からの対象データ値 $value$ の高さを返す関数とする．

$$DIS(RT) = \frac{\sum_{Ai \in QI} \sum_{tj \in PT} \frac{h(VGH_{Ai}, tj(Ai)) - h(VGH_{Ai}, tj'(Ai))}{|DGH_{Ai}|}}{|PT| \cdot |QI|}$$

一般化テーブル RT が一般化される前のテーブル PT とデータ値がまったく同じであれば $DIS(RT)$ は 0 となる．また，一般化が行われるにつれて数値は大きくなり，全てのデータ値が完全に抑制された状態（すべてが $*$ で示されている状態）だと $DIS(RT)$ は 1 となる．したがって，関数 DIS は 0 から 1 の値を取る．

2.6.2 k-極小歪曲の定義

$T_l(A0,...,AM)$，$T_m(A0,...,AM)$ をテーブルとする．これらのテーブルにおける準識別子を $QI_T = \{Ai,...,Aj\} \subseteqq \{A0,...,AM\}$ とする．このとき，テーブル一般化関数 $g_{T_{l_1}}$ によって $T_m[QI_T] = g_{T_{l_1}}(T_l[QI_T])$ のようにテーブル T_l とテーブル T_m の一般化の関係が示されるとする．このとき，

（1） T_m は QI_T において k-匿名条件を満たしている．

（2） 以下の 2 つの条件を満たすテーブル $T_z(A0,...,AM)$ が存在しない．

- テーブル一般化関数 $g_{T_{l_2}}$ によって，$T_z[QI_T] = g_{T_{l_2}}(T_l[QI_T])$ のようにテーブル T_l とテーブル T_z の一般化の関係が示され，かつ，テーブル一般化関数 g_{T_z} によって，$T_m[QI_T] = g_{T_z}(T_z[QI_T])$ のようにテーブル T_z とテーブル T_m の一般化の関係が示される．
- テーブル T_z は k-匿名性を保持していて

$DIS(T_l) \leqq DIS(T_z)$ かつ $DIS(T_z) \leqq DIS(T_m)$

を満たしている．

以上の 2 つ条件を満たしているときに限りテーブル T_m は準識別子 QI_{T_l} 上においてテーブル T_l に対する k-匿名性を保持している極小歪曲な一般化テーブルであると定義する．

また，テーブル T_lとテーブル T_m が与えられ，T_mは T_l の一般化テーブルであり，k-匿名性を保持しているとき以下が言える．

T_m は T_l の極小歪曲テーブル

$\Rightarrow T_m$ は T_l の極小一般化テーブル

3. 従来手法

3.1 *Datafly* [2], [3]

このアルゴリズムの長所は結果テーブルが k-匿名条件

を満たしていることである．このアルゴリズムにおける k-匿名性の定義は本稿で形式的に再定義したものと同じである．短所は，このアルゴリズムのテーブル作成方法により，一般化の際に関係の無いタプルの属性も一緒に一般化してしまうことにより出力される結果テーブルではデータが必要以上に歪曲される場合があるという点である．図 5 が入力テーブル T とされたときの $Datafly$ の出力テーブル T' の例を図 6 に示す．図 6 のテーブル中のタプル $t4$ と $t6$ が完全に抑制されているのは $Datafly$ の仕様である．この点も，テーブル全体のデータが必要以上に歪曲されている原因であると考えられる．

Race	BirthDate	Gender	ZIP	#Occurs	
black	9/20/65	male	02141	1	t1
black	2/14/65	male	02141	1	t2
black	10/23/65	female	02138	1	t3
black	8/24/65	female	02138	1	t4
black	11/7/64	female	02138	1	t5
black	12/1/64	female	02138	1	t6
white	10/23/64	male	02138	1	t7
white	3/15/65	female	02139	1	t8
white	8/13/64	male	02139	1	t9
white	5/5/64	male	02139	1	t10
white	2/13/67	male	02138	1	t11
white	3/21/67	male	02138	1	t12
2	12	2	3		

図 5　$Datafly$ に入力されるテーブル T

Race	BirthDate	Gender	ZIP
black	1965	male	02141
black	1965	male	02141
black	1965	female	02138
black	1965	female	02138
black	1964	female	02138
black	1964	female	02138
white	1964	male	02139
white	1964	male	02139
white	1967	male	02138
white	1967	male	02138

図 6　$Datafly$ により出力される一般化テーブル T' の例

3.2　μ-$Argus$ [1], [3]

このアルゴリズムの長所は $Datafly$ がテーブル全体に対し一般化を行っていた点とは違い，あるタプルを対象としたときに，そのタプルの属性の中で k-匿名条件を満たすのに必要な属性だけを抑制しているので結果テーブルの歪曲度が小さいということである．また短所は，結果テーブルのタプルの属性は単独で抑制されているだけなので（一般化ではなく操作されたデータは抑制状態になっている）表面上では k-匿名条件を満たしているように見えるが本稿で定義した k-匿名条件を満たしている状態に比べデータが予測される可能性がある分，データ保護が十分になされていない．例をあげると，このアルゴリズムでは図 7 のような二つのタプルは同じものとして認識した状態で k-匿名条件を満たしているとしている．本稿での k-匿名性の定義と，このアルゴリズムにおける k-匿名性の定義は同じであるが，提案アルゴリズムでは結果テーブルが図 7 のような 2 つのタプルを同じものとしておらず k-匿名条件を満たしたテーブルとはしていない．両タプルの一般化されている属性，一般化されていない属性に関わらず，全属性が一致している場合のみ，本稿では，2 つのタプルは同一としている．

Race	BirthDate	Gender	ZIP
black	1964	*	*****
black	1964	f	02138

図 7　μ-$Algus$ タプル識別の例

4.　提案アルゴリズム $MinDIS$

$Datafly$，μ-$Argus$ より，すべてのタプルにおいて値それぞれに注目したとき，一般化されているかいないかに関係なく，完全に全属性に対し一致したときのみ 2 つのタプルが同じとして k-匿名性の保持を目標としなければデータ保護は行えないという点がわかった．また，一般化テーブルのデータ歪曲を低くするためには属性値に注目して，必要なタプルにおいてだけ一般化を行うことが有効であるとわかった．

以上の点をふまえて，データ推測を確実に防ぎ，かつ，初期テーブルに比べてデータ歪曲度の低い一般化テーブルを出力できるアルゴリズムを提案する．提案アルゴリズムの概要を図 9 に示す．

図 9 での頻度リストとは，$Datafly$ における $frequency\ list$ に似たもので，リスト上のタプルが出現した回数を保持したリストである．タプルの出現回数は図 8 での $\#occurs$ にあたる．例えば，リスト上の一番上のような属性データの組合せを持つタプルと同じ内容をもつタプル $t1$ とタプル $t2$ の二つがテーブルに存在しているという意味で $\#occurs$ が 2 となっている．このように現状態でのテーブルのタプル情報を要約した表を頻度リストと呼んでいる．

Race	BirthDate	Gender	ZIP	#Occurs	
black	1965	male	02141	2	t1,t2
black	1965	female	02138	2	t3,t4
black	1964	female	02138	2	t5,t6
white	1964	male	02138	1	t7
white	1965	female	02139	1	t8
white	1964	male	02139	2	t9,t10
white	1967	male	02138	2	t11,t12

図 8　頻度リストの例

4.1　実　行　例

提案アルゴリズム $MinDIS$ においても例を挙げて動作を説明する．与えられる初期テーブルは図 5 と同じテーブルとする．

また属性の DGH，VGH は，Race は図 2，BirthDate は図 11，Gender は図 10，ZIP は図 3 をそれぞれ使用するとする．従来手法の例と同じく k は 2 として実行する．

まず，$step1$ で初期テーブルが k-匿名性を満たしてい

Input: テーブル PT; 準識別子 $QI = (A1, ..., An)$, 整数 $k(k \geqq 2 \wedge |PT| \geqq k)$, DGH_{Ai}, VGH_{Ai}, where $i = 1, ..., n$
Output: k-匿名性を保持したテーブル MGT
Assumes: $|PT| \geqq k$
Method:
step1. **If**(PT が k-匿名性を満たしている)**then do**
  step1.1. $MGT \leftarrow PT$, *step*4 へ.
step2. **else do**
  step2.1. PT から頻度リスト *freq* を作る.
  step2.2. 頻度 k 以下のタプルをランダムで選ぶ.
  step2.3. 選んだタプルと仮に一般化したとき最も *DIS* の低いタプルを探す.
  step2.4. 選ばれた 2 つのタプルを実際に一般化し *freq* を更新.
  step2.5. 頻度が k 未満のタプルが存在するならば *step*2.2 へ.

step3. $MGT \leftarrow$ *freq* からテーブル RT を作成.
step4. **Return** MGT

図 9　提案アルゴリズム *MinDIS*

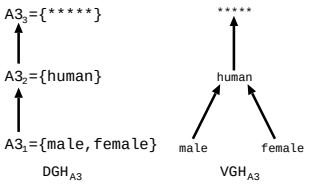

図 10　属性 $A3(Gender)$ の属性一般化階層 DGH と値一般化階層 VGH

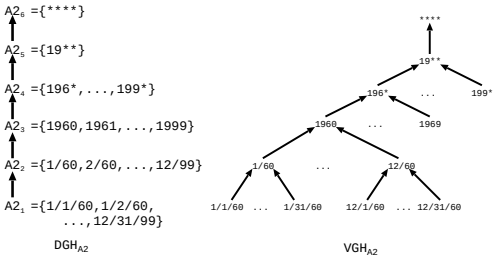

図 11　属性 $a2(BirthDate)$ の属性一般化階層 DGH と値一般化階層 VGH

るか確認する．しかし，初期テーブルは全てのタプルの属性値組合せが独立しており，k-匿名性を満たしてはいない．したがって，*step*2 へ移行する．

次に，*step*2 ではテーブルから図 8 のような頻度リストを作成する．先ほど述べたように，全てのタプルは独立しているので *occurs* は全て 1 となる．*step*2.2 において *occurs* が k 未満のタプルをランダムで選ぶ．ここでは仮に上から 2 番目のタプル (t2) が選ばれたとする．*step*2.3 では先ほど *step*2.2 で選ばれたタプルとそれ以外のタプルを仮に一般化させ属性組合せ値が同一になるように一般化した時のデータ歪曲度を算出する．t2 と他のタプルとを一般化させたときのデータ歪曲度は，

$$t2, t1 : DIS = 0.100,\ \ t2, t3 : DIS = 0.392$$
$$t2, t4 : DIS = 0.392,\ \ t2, t5 : DIS = 0.442$$
$$t2, t6 : DIS = 0.442,\ \ t2, t7 : DIS = 0.442$$
$$t2, t8 : DIS = 0.516,\ \ t2, t9 : DIS = 0.442$$
$$t2, t10 : DIS = 0.442,\ \ t2, t11 : DIS = 0.442$$
$$t2, t12 : DIS = 0.442$$

となり，データ歪曲度の最も小さかった t1 が候補に選ばれる．*step*2.4 で実際に選ばれた 2 つのタプルを一般化して $freq$ を更新する．*step*2.5 で再度，$freq$ に格納されている各タプルの *occurs* を調べ，もし k 未満のタプルが存在していれば *step*2.2 へ戻る．

ここで *step*2.2 に戻り，更新された $freq$ を元に同じ手順で一般化する候補のタプルを探していく．この手順を繰り返し，$freq$ においてすべてのタプルの *occurs* が k 以上 (この例では 2 以上) になるテーブルを作成する．仮に次は t5,t6 のペアが，続いて t9, t10 のペア，t11, t12 のペア，t3, t4 のペアが選ばれ，一般化が行われたとする．残った *occurs* が 2 未満のタプル t7, t8 のうち，続くループ中で t7 は t9, t10 のグループと，また t8 は t3, t4 のグループと一般化されて，最終的に k-匿名性を満たしたテーブルへと変換される．

実行例の結果として出力される一般化テーブルを図 12 に示す．

Race	BirthDate	Gender	ZIP
black	1965	male	02141
black	1965	male	02141
Person	1965	female	0213*
Person	1965	female	0213*
black	1964	female	02138
black	1964	female	02138
white	1964	male	0213*
Person	1965	female	0213*
white	1964	male	0213*
white	1964	male	0213*
white	1967	male	02138
white	1967	male	02138

図 12　提案アルゴリズム *MinDIS* の実行例における結果テーブル

4.2 特　　徴

提案アルゴリズム *MinDIS* は従来手法 $Datafly$ のように属性一般化階層 DGH と値一般化階層 VGH を使用し，その一般化も完全に属性値組合せが同じタプルを同一値を持つと定義する．また，全てのタプルにおいて自分を含み同一値を持つタプルが k 以上の状態を k-匿名性を保持していると定義したので，提案手法 *MinDIS* の結果テーブルが k-匿名性を保持していることは確実に保証される．

また従来手法 μ-$Argus$ と同様にタプルごとの属性値に注目し，二つのタプルが同一値を持つように一般化するのに必要なタプルの属性値だけを一般化している．したがって，従来手法 μ-$Argus$ のデータ歪曲度ほど小さくはないが，従来手法 $Datafly$ に比べデータ歪曲度のより低い結果テーブルを出力できると予想される．

表 1　提案手法 *MinDIS* と従来手法 *Datafly* のデータ歪曲度に関する比較

		MinDIS						Datafly		
		k=2		k=5		k=10				
TUPLE	ATT	min	max	min	max	min	max	k=2	k=5	k=10
10	5	0.150	0.250	0.350	0.450			0.450	0.550	
	10	0.450	0.575	0.570	0.678			0.700	0.775	
	100	0.714	0.725	0.752	0.764			0.893	0.903	
	1000	0.742	0.748	0.784	0.786			0.999	0.922	
100	5	0.150	0.250	0.254	0.285	0.283	0.317	0.250	0.450	0.460
	10	0.405	0.419	0.520	0.542	0.512	0.548	0.675	0.675	0.700
	100	0.663	0.668	0.795	0.802	0.794	0.803	0.901	0.901	0.903
	1000	0.727	0.729	0.851	0.852	0.851	0.853	0.918	0.922	0.922
1000	5	0.125	0.129	0.174	0.175	0.189	0.192	0.200	0.250	0.250
	10	0.298	0.320	0.419	0.428	0.415	0.425	0.575	0.575	0.675
	100	0.632	0.640	0.774	0.780	0.779	0.782	0.881	0.891	0.901
10000	5	0.100						0.200		
	10	0.225						0.475		
	100	0.608						0.870		

従来手法 *Datafly* や *μ-Argus* が一つの入力テーブルに対し 1 通りの結果テーブルを出力するのに対し提案アルゴリズム *MinDIS* で出力される結果一般化テーブルは一般化するタプルを選択する際にランダム選択を行っていることから，実行するたびに異なる出力結果が得られる．

従来手法と異なり，実行されるたびに異なる結果テーブルが出力されるという問題点はあるが，提案アルゴリズム *MinDIS* は k-匿名性を保持するのに必要な個所だけを一般化し k-匿名性を保持した段階でテーブルを出力している．故に結果テーブルはどれも極小歪曲テーブルであり極小一般化テーブルでもある．極小一般化の定義中にも記述したがあくまで結果テーブルから直接に一般化して得られるテーブルの集合の中では極小の一般化であるので，結果テーブルが最小歪曲であることは保証されていない．極小歪曲においても同様のことが言える．しかし，一般化の際にタプルに注目したときデータ歪曲の一番少ない一般化対象タプルを選択しているので，極小歪曲テーブルの中でも比較的テーブルのデータ歪曲度が低い結果が得られると予測される．結果一般化テーブルはどれも極小歪曲であることを利用し，ある属性についてのデータ歪曲度に注目して，複数の結果一般化テーブルの中でその属性が一番歪曲されていないテーブルを出力することも提案アルゴリズムを使用すれば可能である．

5. 実　　験

従来手法 *μ-Argus* [1] はデータが推測される可能性がある欠点が指摘されていた [3]．よって確実にデータ推測を防いでいる *Datafly* [2] と提案アルゴリズム *MinDIS* を計算機上 (UltraSPARC-IIi 440MHz, メモリーサイズ: 512 M byte) に *C++*言語でプログラムを記述して実装し，シミュレーション実験により性能を比較した．提案アルゴリズム *MinDIS* と従来手法 $Datafly$ は同じ属性一般化階層 *DGH* と値一般化階層 *VGH*，k-匿名性判定を取り入れているので，本節では $Datafly$ との初期テーブルに対する出力された結果一般化テーブルのデータ歪曲度について，2 つの手法を比較する．また，データ歪曲度算出関数 *DIS* において抑制状態のデータ歪曲度を定義していなかったが，抑制状態は最大一般化状態と同じとみなしてデータ歪曲度を算出した．

5.1　人工データによる実験

タプル数及び属性数を変化させた場合の 2 つのアルゴリズムのデータ歪曲度を比較するために，テーブルのどの個所を取り出しても属性の種類の出現頻度が同じになるように入力データテーブルをランダム作成し使用した．シミュレーション実験結果を表 1 に示す．提案アルゴリズム $MinDIS$ はアルゴリズム中にランダムにタプルを選択するので，データ歪曲度 DIS の数値はタプル数 10000 以外のテーブルについては 25 回実行した結果の最小値と最大値を示した．タプル数 10000 のテーブルは満足な実行回数を消化するのに時間がかかりすぎるので参考として 1 回実行した結果を載せている．

表 1 の結果よりすべてのタプル数と属性数の組合せにおいて，提案アルゴリズム *MinDIS* は従来手法 *Datafly* よりデータ歪曲度が小さい数値を示した．結果から提案アルゴリズム *MinDIS* はタプル数，属性数に関わらず従来手法 *Datafly* よりデータ歪曲度が小さい数値結果を出力できるであろうと予測される．

5.2　実データによる実験

ランダムに作成したデータではなく実データ (ベンチマークデータ) を入力したときに提案アルゴリズム *MinDIS* と従来手法 *Datafly* を用いて k- 匿名性を保持させる一般化を行い，データ歪曲度を算出した．シミュレーション実験結果を表 2 に示す．データは *University of California, Irvine* の *KDD(Knowledge Discovery in Databases)* アーカイブ (http://kdd.ics.uci.edu) からの *coil1999*(analysis.data) の河川物質データ (data1) と *coil2000*(ticdata2000.txt) の保険会社のデータ (data2, data3) と *Japanese Vowels*(ae.test) データ (data4) を使

用した．

表 2　実データ (ベンチマークデータ) における提案手法 *MinDIS* と従来手法 *Datafly* のデータ歪曲度の比較

name (|PT|,n)	MinDIS k=2 min	MinDIS k=2 max	MinDIS k=10 min	MinDIS k=10 max	Datafly k=2	Datafly k=10
data1 (200,18)	0.642	0.652	0.831	0.845	0.995	0.933
data2 (1455,86)	0.156	0.163	0.254	0.262	0.919	0.930
data3 (5822,86)	0.080	0.089	0.148	0.156	0.944	0.944
data4 (5687,12)	0.602	0.604	0.689	0.698	0.861	0.889

表 2 の結果においても図 1 の結果のようにすべてのデータを入力したときの結果一般化テーブルのデータ歪曲度は提案アルゴリズム *MinDIS* の値のほうが従来手法 *Datafly* の値より小さくなった．ランダムデータとの大きな違いはデータによりデータ歪曲度の改善具合いがとても大きいということである．data2, data3 のデータ歪曲度はかなり低い値を示していると言える．

5.3 考　　察

表 2 より，各データにおける結果として出力されるテーブルのデータ歪曲度にかなりの差がある．なぜこのようにデータ間の結果に差が出たかというと，計算機上に実装した提案アルゴリズム *MinDIS* 及び従来手法 *Datafly* では属性一般化階層 *DGH* と値一般化階層 *VGH* を自動で作成される簡易的な階層として使用していたことが原因であると考えられる．自動で作成された簡易的な一般化階層に属性が合っているデータについてはデータ歪曲度が比較的低く，そうでないデータについては高くなってしまった．この問題点の解決法の一つとして 2.2.1 節で述べたように各々のデータに合った属性一般化階層 *DGH* と値一般化階層 *VGH* をデータ管理者が作成することが考えられる．

節 5.1, 5.2 の結果から提案アルゴリズム *MinDIS* は確実に k-匿名性を保持してデータ推測を防ぎ，従来手法 *Datafly* よりデータ歪曲度の低い結果一般化テーブルを出力できるアルゴリズムとわかった．

また表 3 に提案アルゴリズム *MinDIS* 及び従来手法 *Datafly* における実行時間を示した．ランダム作成したデータは実データと違いタプル間の因果関係が小さいと思われるので，実行時間が実データより長い．タプル間の因果関係が大きい実データはランダム作成データに比べて実行時間が短いが，タプル数及び属性数が大きくなるにつれて実行時間が長くなることは，どのデータにおいても言える．実行時間が長くなる原因はタプルを一般化する際に最良の一般化対象を全タプルグループから探していることだと考えられる．$data3, data4$ において $k = 2$ より $k = 10$ の場合のほうが実行時間が短いのは，k の値が大きいほど段々とタプルグループ数が多くなり，局所最適な一般化対象を探すための比較が少なくなったためだと考えられる．最良の一般化対象タプルをある程度絞り込んで探す手法を提案アルゴリズムに取り入れる等の改善が今後必要である．

また，従来手法 $Datafly$ の計算量 $O(mn^2)$(m はタプル数，n は属性数) に比べ提案アルゴリズムの計算量は $O(kmn^2)$ であった．データの内容と規模にもよるが，今回の実験においては実行時間に大きな差はなかった．

表 3　提案アルゴリズム *MinDIS* と従来手法 *Datafly* における実行時間

data	TUPLE	ATT	MinDIS[sec] k=2	MinDIS[sec] k=10	Datafly[sec] k=2	Datafly[sec] k=10
random data	100	5	0.04	0.04	0.05	0.05
	100	10	0.04	0.05	0.17	0.21
	1000	5	15.02	27.88	4.98	4.99
	1000	10	15.64	22.32	17.08	20.27
	1000	100	40.69	49.57	1354.25	1409.69
data1	200	18	0.42	0.63	17.29	16.23
data2	1422	86	101.25	123.94	2121.70	2047.74
data3	5822	86	4285.32	1542.87	1927.85	2001.23
data4	5687	12	3659.25	2229.36	1599.11	1842.30

6. おわりに

本稿では確実に k-匿名性を保持してデータ推測を防ぐ，従来手法 *Datafly* よりデータ歪曲度の低い結果一般化テーブルを出力できるアルゴリズムを提案した．しかし，提案アルゴリズムはあくまでデータ歪曲度が極小な結果を出すのであって，一般化の関係をもたないより小さな歪曲度をもつ結果が存在する可能性がある．また入力テーブルが大きくなるほどタプル及び属性数に比例し一般化の回数は増加するので実行時間は大きくなってしまう．今回はただ単にデータ歪曲度が低いテーブルほどデータ解析者にとって有用なテーブルであるとし，出力テーブルを作成していた．しかしこのようなテーブルが常に全ての解析者にとって有用であるかはわからないので，解析者にとっての解析しやすさについても考慮し，出力結果テーブルを作成する必要がある．これらの点に着目したアルゴリズムの改良は今後の課題である．

文　　献

[1] A. Hundepool, L. Willenborg, “ARGUS for protecting microdata and tables,” Seminar on New Techniques & Technologies for Statistics, 1998.

[2] L. Sweeney, “Guarranteeing anonymity when sharing medical data, the Datafly system,” Journal of the American Medical Informatics Association, pp.1–5, 1997.

[3] L. Sweeney, “Achieving k-anonymity privacy protection using generalization and suppression,” International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5), pp.571–588, 2002.