データを極小歪曲しk-匿名性を保持したデータに変換するプライバシー保護アルゴリズム

A Privacy Protection Algorithm to Convert Data into Data Having k-Anonymity with Minimal Distortion

村本 俊祐† 上土井 陽子†† 若林 真一†††

Shunsuke MURAMOTO Yoko KAMIDOI Shin'ichi WAKABAYASHI

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．

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*.

† 学生会員 広島市立大学大学院情報科学研究科博士前期課程
shun@lcl.ce.hiroshima-cu.ac.jp
†† 会員 広島市立大学大学院情報科学研究科
yoko@ce.hiroshima-cu.ac.jp
††† 非会員 広島市立大学大学院情報科学研究科
wakaba@ce.hiroshima-cu.ac.jp

表 1: 2-匿名性保持を目的とした一般化
Table1: Generalization for maintaining 2-anonymity.

(a) 初期テーブル PT

	Race	Birth	Gender	ZIP
t1	Black	1964	female	02138
t2	Black	1964	female	02138
t3	Black	1967	male	02141
t4	White	1971	female	02139
t5	White	1967	male	02141
t6	White	1971	male	02139
t7	White	1965	male	02141

(b) 一般化テーブル RT

	Race	Birth	Gender	ZIP
t1′	Black	1964	female	02138
t2′	Black	1964	female	02138
t3′	Person	196*	male	02141
t4′	White	1971	human	02139
t5′	Person	196*	male	02141
t6′	White	1971	human	02139
t7′	Person	196*	male	02141

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. k-匿名性

データテーブルは表 1 のような有限個のタプル (行に対応) と属性 (列に対応) からなるものを考慮する．ここで各タプルは各属性に属するデータ値の属性数 n 個の組とする．また，個人を特定する単独の識別子ではないが組み合せることで同じ働きをする恐れのある属性の集合を準識別子 QI と呼ぶ．

従来手法 [1][2] ではテーブルに k-匿名性を持たせるために一般化や，抑制というデータ操作を使用していた．まず抑制とはデータ値がすべて隠された状態を指す．データ値の状態を大きく分けると初期状態と抑制状態に分けられる．一般化とは，その二つの状態の中間の状態を示すために，データ値の一部分を隠す，またはより広い値域を指す値に変換するデータ操作である．

ここで k-匿名性を以下のように定義する．

> **データテーブル中の各タプルにおいて，そのタプルのもつデータ値情報 (各属性値の組合せ) と同じデータ値情報を持つタプルが自分自身を含め k 個以上存在する状態**

k-匿名性の例を挙げる．表 1(a) のテーブル PT が与えられたとき，表 1(b) のテーブル RT に変換したとする．テーブル PT のタプルに注目すると，この状態では $t1$，$t2$ のタプルは同一データ値組合せであるが，その他のタプルは独立している．一方，テーブル RT では，$t1$，$t2$ のタプルが同一データ値組合せを持っており，同様に $t3$，$t5$，$t7$ の 3 つのタプル，$t4$ と $t6$ の 2 つのタプルがそれぞれ同一データ値組合せを持っている．よって，テーブル RT では全てのタプルにおいて同一データ値組合せをもっているタプルが自分を含め 2 個以上存在する．このとき，テーブル RT は 2-匿名性を保持していると言う．k-匿名性 ($k \geq 2$) を保持しているテーブルではどのタプルも公開前データのタプルに一意に対応していないので複数データ項目の組合せによる，データ推測が防止されているといえる．

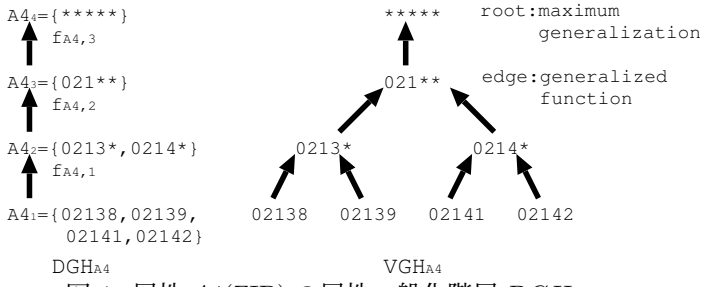

図 1: 属性 $A4$(ZIP) の属性一般化階層 DGH, 値一般化階層 VGH

Fig1: DGH and VGH for Domain A4(ZIP).

3. データ歪曲度算出関数 *DIS*

2 節で紹介した k-匿名性保持を目的とした一般化をアルゴリズムに取り入れることにより，複数データ項目の組合せによるデータ推測を防ぐことを考える．このとき前述のとおり，データテーブルに k-匿名性を保持させるためには，一般化等のデータ操作を必要とする．しかし一般化等のデータ操作は元のデータを歪曲してしまう．本アルゴリズムでは，データを利用する際に解析などがしやすいように元のデータになるべく近い形で k-匿名性を保持したデータに変換することを考えた．したがって，より歪曲の少ない結果を出す必要があった．よって，本研究では一般化を行うことで得られたデータテーブルが元のデータテーブルに対して，どの程度変化したか (データ歪曲度) を評価するために文献 [3] を元に，データ歪曲度算出関数 *DIS* を新たに数式を用いて定義した．

提案アルゴリズムは属性を初期値の集合から最大一般化値までに一般化された回数で階層的に分ける属性一般化階層 DGH(Domain Generalization Hierarchies) と，一般化前の値と後の値の関係を木 (最大一般化値を根とする) で表現した値一般化階層 VGH(Value Generalization Hierarchies) という一般化表現を使用している．表 1 のデータテーブル中の属性 *ZIP* に関しての属性一般化階層 DGH および値一般化階層 VGH の例を図 1 に示す．属性一般化階層 DGH，値一般化階層 VGH はデータテーブル上の各属性それぞれに存在し，複数階層から成るものと定義する．基本的に属性一般化階層 DGH と値一般化階層 VGH はデータテーブルの管理者が任意に作成可能である．また DGH_{Ai}，VGH_{Ai} は属性 Ai の属性一般化階層 DGH と値一般化階層 VGH という意味をもつとする．

テーブル PT が一般化テーブル RT に変換されたときのデータ歪曲度算出関数 *DIS* の定義式を以下に示す．

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

式中の tj はタプルを指し，$tj(Ai)$ でタプル tj 中の属性 Ai に対応する値を示し，関数 $h(tree, v)$ は木 $tree$ 中の値 v の高さを返す関数である．また DGH の絶対値をとることでその属性一般化階層関数 DGH の階層数が得られるとする．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* 等の情報が得られない状態) だと $DIS(RT)$ は 1 となる．したがって，データ歪曲度算出関数 *DIS* は 0 から 1 の値を取る．

Input: テーブル PT; 準識別子 $QI = (A1, ..., An)$, 整数 $k(k \geq 2 \wedge |PT| \geq k)$, DGH_{Ai}, VGH_{Ai}, ここで $i = 1, ..., n$
Output: k-匿名性を保持したテーブル MGT
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に選ぶ.
  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 2: 提案アルゴリズム *MinDIS*

Fig2: Proposed algorithm *MinDIS*.

4. 提案アルゴリズム *MinDIS*

データ歪曲度算出関数 *DIS* を導入することでデータ歪曲度による一般化を評価することが可能となり，評価を基に歪曲の少ない一般化を選択することが可能になった．また提案アルゴリズムではテーブルを k-匿名性を保持したテーブルに変換する過程で行ったどの一般化を欠いても出力テーブルが k-匿名性を保持しなくなるという性質を満たす．よって k-匿名性保持に不必要な一般化は行わないので，このような一般化により得られたテーブルを k-極小歪曲なテーブルと定義する．確実に k-匿名性を保持し，データ歪曲度の低いテーブルを出力することを目的に提案したアルゴリズム *MinDIS* を図 4 に示す．図 4 での頻度リスト *freq* は各タプルの属性値組合せが同一なタプルの個数を保持したリストである．

提案アルゴリズム *MinDIS* の動作を例を挙げて説明する．与えられる初期テーブルは表 2 のテーブルとする．また属性の VGH として，Race 及び Gender は図 3，BirthDate は図 4，ZIP は図 1 をそれぞれ使用するとする．ここの例では k の値を 2 として実行する．

まず，*step*1 で初期テーブルが k-匿名性を満たしているか確認する．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 ではテーブルから頻度リストを作成する．表 2 のテーブル右側の *occurs* がタプルの出現頻度を示している．表 2 の初期テーブルは全てのタプルが独立しているので *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に選ぶ．ここでは上から 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属性値組合せ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 ti とを一般化させたときの一般化データ $RT_{t2,ti}$ のデータ歪曲度は，

$DIS(RT_{t2,t1}) = 0.100$, $DIS(RT_{t2,t3}) = 0.392$
$DIS(RT_{t2,t4}) = 0.392$, $DIS(RT_{t2,t5}) = 0.442$
$DIS(RT_{t2,t6}) = 0.442$, $DIS(RT_{t2,t7}) = 0.442$
$DIS(RT_{t2,t8}) = 0.516$, $DIS(RT_{t2,t9}) = 0.442$
$DIS(RT_{t2,t10}) = 0.442$, $DIS(RT_{t2,t11}) = 0.442$
$DIS(RT_{t2,t12}) = 0.442$

となり，データ歪曲度の最も小さい $t1$ 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

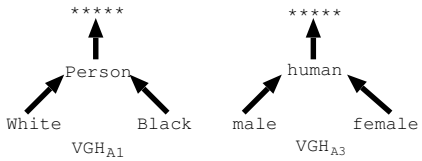

図 3: 属性 $A1$($Race$)，$A3$($Gender$) 値一般化階層 VGH

Fig3: VGH for Domain A1(Race), A3(Gender).

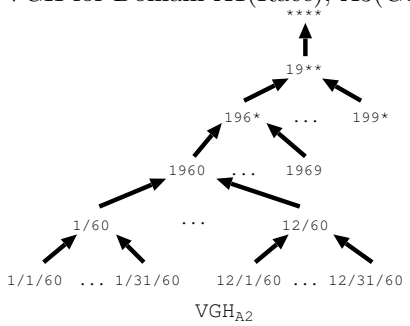

図 4: 属性 $A2$($BirthDate$) の値一般化階層 VGH

Fig4: VGH for Domain A2(BirthDate).

2 以上) になるテーブルを作成する．例えば次は $t5, t6$ 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，初期テーブルは最終的に 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表 3(a) に示す．また，比較として従来手法 *Datafly* の出力結果を表 3(b) に示す．*Datafly* では，テーブル中の他のタプルと同一データ値組合せを持っていないタプルが k 個未満になったら一般化を終了し，そのタプルを完全に抑制する．表 3(b) の最下行 2 つのタプルが完全に抑制されているのはそのためである．

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ランダム作成した入力データテーブルを使用した．シミュレーション実験結果を表 4 に示す．提案アルゴリズム *MinDIS* はアルゴリズム中で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 のテーブルは実行回数を 25 回にすると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 4 よりすべてのタプル数と属性数の組合せにおいて，提案

表 2: 入力テーブル T

Table2: Input Table T.

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		

表 3: 出力テーブル

Table3: Output Tables.

(a)$MinDIS$

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

(b)$Datafly$

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

アルゴリズム *MinDIS* は従来手法 *Datafly* よりデータ歪曲度の小さい結果を出力した．以上より提案アルゴリズム *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，それらの結果の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 5 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 5 においても表 4 のようにすべてのデータにおいて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．ランダムデータでの結果との大きな違いはデータによりデータ歪曲度の改善度合いがとても大きいということである．data2, data3 のデータ歪曲度はランダムデータのそれらと比べ，かなり低い値を示していると言える．

5.3 考察

表 5 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ことがわかる．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，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．しかし，

表 4: データ歪曲度に関する比較
Table4: Comparison of output data distortion degrees.

TUPLE	ATT	MinDIS k=2 min	MinDIS k=2 max	MinDIS k=5 min	MinDIS k=5 max	MinDIS k=10 min	MinDIS k=10 max	Datafly k=2	Datafly k=5	Datafly 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		0.121		0.141		0.200	0.200	0.250
	10	0.225		0.342		0.343		0.475	0.475	0.475
	100	0.608		0.754		0.755		0.870	0.871	0.881

表 5: ベンチマークデータにおけるデータ歪曲度の比較
Table5 : Comparison of output data distortion degrees on benchmark data.

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

作成可能な階層には条件があり，一般化階層に含まれる値は，一般化を行う過程で必ず最終的には一つの値 (最大一般化された値) に一般化されることがあげられる．また最大一般化された値は一つの属性につき一つだけ存在するということも重要な条件である．

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 6 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相関関係が疎と思われるので，実行時間が実データより長い．タプル間の相関関係が密な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．データサイズが大きいデータにおける実行時間が長くなる一番の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．タプルの持つデータ値と一般化した際に起こるデータ歪曲の関係について研究がまだ必要な点があり，比較を行う対象のタプルを絞り込むことに成功していない．よって現在のアルゴリズムでは、全タプルを対象に比較すること無しではタプルを共に一般化する他のタプルを選出することができない．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．$data3, data4$ 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同一データ値組合せを持つタプルの数が多くなり，比較の際には同一データ値組合せを持つタプル群の中の 1 つと比較を行えばよいので，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．これより，k の値が大きくなった場合でも実行時間が短くなる可能性があるということがわかった．実行時間の結果より，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．

表 6: 実行時間の比較
Table6: Comparison of execution times.

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．また一般化階層に使用した一般化は全て等しい重みであった．それぞれの一般化に独自の重みを付加するなど，データ歪曲度自体についても改良の余地があると考えている．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.

村本 俊祐 Shunsuke MURAMOTO
広島市立大学大学院情報科学研究科在学中．2007 広島市立大学情報科学部情報工学科卒業．データベース上でのプライバシー保護法の研究に従事．日本データベース学会学生会員．

上土井 陽子 Yoko KAMIDOI
広島市立大学大学院情報科学研究科講師． 1994 広島大学大学院工学研究科博士課程後期修了．博士 (工学)．主にデータマイニング，クラスタリングの研究に従事．日本データベース学会，電子情報通信学会，IEEE，ACM，SIAM 各会員．

若林 真一 Shin’ichi WAKABAYASHI
広島市立大学大学院情報科学研究科教授．1984 広島大学大学院工学研究科博士課程後期修了．工学博士．日本アイ・ビー・エム (株) 東京基礎研究所副主任研究員，広島大学工学部助教授を経て，2003 より現職．主として，VLSI CAD，VLSI 設計，組合せ最適化に関する研究に従事．情報処理学会，IEEE，ACM 各会員．