2Z-2

k-匿名化手法の効率向上に関する一提案

渡邉 奈津美† 土井 洋‡ 趙 晋輝†

中央大学大学院理工学研究科† 情報セキュリティ大学院大学‡

1. はじめに

プライバシ保護手法のひとつとして *k*-匿名化が知られている[1]. 匿名化前後のデータ間の歪度はデータ歪曲度と呼ばれ, *k*-匿名性を満たしつつデータ歪曲度を小さくすることが望ましい. 大域的一般化を用いた既存手法[2]は処理時間が短くて済む一方, データを過度に歪曲してしまうという欠点がある. これに対し, 局所的一般化を用いた既存手法[3]はデータの過度な歪曲を防ぐ一方で, 処理に時間を要するという欠点がある. 本研究では, 必要に応じて局所的一般化手法と大域的一般化手法を組み合わせることにより, データの過度な歪曲を防ぎつつも処理時間の短い *Hybrid k*-匿名化手法を提案し, 実装結果も示す.

2. *k*-匿名性

本稿では, 有限個のレコード(行に対応)と属性(列に対応)からなるテーブル形式の DB を扱う. テーブルの各セルに格納されている値は属性値と呼ばれる. *k*-匿名性とは“テーブルのどのレコードに関しても, 同じレコードが自身を含め *k* 個以上存在する”ということを保証するプライバシ保護指標である. *k*-匿名性を満たすようにデータの一般化や属性の削除等を行うことを *k*-匿名化という. 既存手法[2][3]では, データの一般化や抑制により *k*-匿名化を行っている. 一般化とはデータ値の一部分を隠す, またはより広い値域を示す値に置き換える操作のことである. 抑制とはデータ値すべてを隠してしまうことである. 図 1 は一般化による 2-匿名化の例である.

匿名化前のテーブルT_P

郵便番号	性別
02138	女
02139	女
02141	男
02142	男

匿名化後のテーブルT_R

郵便番号	性別
0213*	女
0213*	女
0214*	男
0214*	男

図 1 : 2-匿名化の例

3. データ歪曲度算出関数

データ歪曲度は, 一般化を行うことで得られたテーブルと元のテーブルとの変化の度合いを測るための指標である. 文献[3]にて提案されたデータ歪曲度算出関数 *DIS* は一般化階層木の深さの差に基づいて算出される.

一般化階層木は, 一般化前後でのデータ値の関係を階層的に示す木のことであり, 属性ごとに与えられる. 図 2 は一般化階層木の例である. 一般化階層木をどのように作成するかは, 匿名化処理を行うテーブルの管理者に委ねられている.

A Proposal of Efficiency Improvement of k-Anonymization Schemes
Natsumi Watanabe†, Hiroshi Doi‡, Junhui Chao†
† Graduate School of Science and Engineering, Chuo University
‡ Institute of Information Security

レコード数 m, 属性数 n のテーブルT_Pを一般化して得られたテーブルT_Rの *DIS* は次式で与えられる.

$$DIS(T_R) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{h\left(GH_{A_j}, v_{i,j}\right) - h\left(GH_{A_j}, v'_{i,j}\right)}{|GH_{A_j}|}}{mn}$$

GH_{A_j}は属性A_jの一般化階層木であり, $v_{i,j}$はテーブルT_Pの i 行 j 列目, $v'_{i,j}$はテーブルT_Rの i 行 j 列目の値である. また, *h*(*tree*,*value*)は一般化階層木 *tree* における値 *value* の根からの深さを返す関数であり, |*tree*|は一般化階層木の根からの最大の深さである.

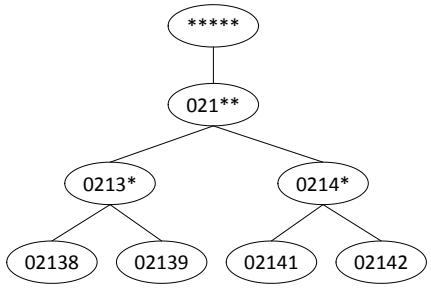

図 2 :一般化階層木 GH_{A_1} (A_1: 郵便番号) の例

4. 既存手法

大域的一般化では, テーブルのすべてのレコードに対し, ある属性の 2 つ以上の値を一般化階層木の共通する祖先の値に置き換える. これに対し局所的一般化では, テーブルのある 2 つ以上のレコードについて, 各属性が同一の値になるように必要に応じて共通する祖先の値に置き換える.

レコードの重複度を頻度と呼び, 各レコードの頻度を数え上げたものを頻度リストと呼ぶ.

既存手法 *Datafly*[2]は頻度 *k* 未満のレコードが *k* 個未満になるまで大域的一般化を繰り返し, 残りの頻度 *k* 未満のレコードを抑制する *k*-匿名化手法である. この手法は処理時間が短い一方で, データを過度に歪曲してしまうという問題があった. それを解決するために提案されたのが *MinDIS*[3]である.

MinDIS はランダムに選ばれた頻度 *k* 未満のレコードと局所的一般化を行った際に, 最も *DIS* が小さくなるレコードを探して局所的一般化を行う. これを頻度 *k* 未満のレコードが存在する限り繰り返すことにより, *k*-匿名性を満たすテーブルを作成する. この手法はデータの過度な歪曲を防ぐ一方で, 処理に時間がかかる. なぜなら, どのレコードと一般化した際に *DIS* が最小になるかを, 頻度 *k* 未満のレコードが選ばれる度に計算する必要があるからである.

そこで本稿では, *MinDIS* を行う前に大域的一般化を用いて各レコードの重複度を高めることによって, すなわち *MinDIS* において局所的一般化の処理の対象となるレコードを削減することによって, データの過度な歪曲を防ぎつつも処理時間の

短い *Hybrid k*-匿名化手法を提案する.

5. 提案手法

Hybrid k-匿名化手法では各属性の取り得る値の数が$\frac{m}{k}$以下になるまで大域的一般化を行った後, *MinDIS* にてテーブル全体を *k*-匿名化する. 各属性の取り得る値の数が$\frac{m}{k}$以下であることは, テーブル全体が *k*-匿名性を満たすための必要条件である.

(定理 1) テーブル T が k-匿名性を満たすならばテーブル T の各属性の取り得る値の数は$\frac{m}{k}$以下である.
(証明)自明なので省略する.

提案手法のアルゴリズムを図 3 に示す. なお, $|A_j|$は属性$A_j(j = 1,2, \dots, n)$の取り得る値の数であり, 大域的一般化を行う度に減少する.

Input : テーブルT_P, 整数 $k(2 \leqq k \leqq m)$, 一般化階層木 $GH_{A_j}(j = 1,2, \dots, n)$
Output : *k*-匿名性を満たすテーブルT_R
Step1. **If** (テーブルT_Pが *k*-匿名性を満たす) **then do**
　　$T_R \leftarrow T_P$, Step6 へ.
Step2. **else do** T_Pから頻度リスト *freq* を作る.
/* 各属性が *k*-匿名性を満たすよう大域的一般化 */
Step3. **For each** j $(j = 1,2, \dots, n)$ **do**
　　While($|A_j|$>$\frac{m}{k}$) A_jを大域的一般化する.
Step4. 頻度リスト *freq* を更新する.
/* テーブル全体が *k*-匿名性を満たすよう局所的一般化 */
Step5. **While**(頻度 *k* 未満のレコードが存在)
　Step5.1. 頻度 *k* 未満のレコードをランダムに選ぶ.
　Step5.2. 選んだレコードと局所的一般化を行った際に, *DIS* が最小になるようなレコードを探す.
　Step5.3. 選ばれた 2 つのレコードを局所的一般化し, *freq* を更新する.
Step6. *freq* からT_Rを作成する.
Step7. **Return** T_R

図 3 : *Hybrid k*-匿名化手法のアルゴリズム

6. 実装結果

既存手法と提案手法を実装し, データ歪曲度および実行時間を評価した結果を以下に示す. 実装環境は CPU が Intel® Core™ i7-2600K CPU(3.4GHz), 実装メモリが 4GB である. 人工的に作成した一様ランダムなデータおよび, 実データを用いて実験を行った. 実データは sample1 として[4]の ticdata2000.txt, sample2 として[5]の ae.test を用いた. *MinDIS* および *Hybrid* はランダム性を持つため, 100 回実行した際の平均値を示す. なお, 一般化階層木は入力データをもとに 2 分木を生成している.

7. 考察

提案手法においては, random1, random2, sample2 を入力として与えた場合, データの過度な歪曲を防ぎつつも処理時間を短縮することができた. しかしながら, sample1 に対しては, 処理時間が増えてしまった. これは, sample1 のいずれの属性も取り得る値の数が小さいため, テーブルが入力された時点で各属性の取り得る値の数が$\frac{m}{k}$以下である (大域的一般化の必要がない)にも関わらず, 提案手法の step3 にて, 各属性の取り得る値の数が$\frac{m}{k}$以下か否かの判定を行った後に *MinDIS* を実行するためである. 逆に各属性の取り得る値の数が多いとき, 提案手法は有効であると考えられる.

8. おわりに

本稿では, データの過度な歪曲を防ぎ, なおかつ処理時間も短くなるような *Hybrid k*-匿名化手法を提案した. 提案手法では各属性の取り得る値の数が$\frac{m}{k}$以下になるまで大域的一般化を行っているが, この条件が最適であるとは言えない. 大域的一般化から局所的一般化へ切り替える適切な条件を算出することが今後の課題である.

表 1 : データ歪曲度の比較

入力データ (m, n)	k	*Datafly*	*MinDIS*	*Hybrid*
random1 (5000, 5)	2	0.933	0.605	0.608
	5	0.917	0.816	0.816
	10	0.917	0.870	0.869
random2 (5000, 10)	2	0.967	0.697	0.700
	5	0.958	0.887	0.886
	10	0.959	0.929	0.928
sample1 (5822, 86)	2	0.982	0.059	0.059
	5	0.982	0.204	0.205
	10	0.972	0.324	0.324
sample2 (5687, 12)	2	0.974	0.171	0.274
	5	0.974	0.371	0.366
	10	0.968	0.479	0.478

表 2 : 実行時間の比較

入力データ (m, n)	k	*Datafly*	*MinDIS*	*Hybrid*
random1 (5000, 5)	2	1.389	2.469	2.130
	5	1.392	2.890	2.513
	10	1.387	2.889	2.541
random2 (5000, 10)	2	4.110	5.166	4.092
	5	4.098	6.035	4.872
	10	4.111	6.006	4.923
sample1 (5822, 86)	2	15.334	12.845	15.424
	5	15.304	15.099	17.697
	10	15.350	15.339	18.015
sample2 (5687, 12)	2	3.500	10.394	3.898
	5	3.490	12.232	4.669
	10	3.520	12.337	4.776

[参考文献]

[1] 経済産業省:情報大公開プロジェクト「パーソナル情報保護・解析基盤」のご紹介, http://www.meti.go.jp/policy/it_policy/daikoukai/igvp/cp_jp/cat305/post-3.html
[2] L. Sweeney:Guaranteeing anonymity when sharing medical data, the Datafly system, Journal of the American Medical Informatics Association, pp.1–5, (1997)
[3] 村本俊祐, 上土井陽子, 若林真一: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, 日本データベース学会Letters, Vol.6, No.1, pp.97–100 (2007)
[4] http://kdd.ics.uci.edu/databases/tic/
[5] http://kdd.ics.uci.edu/databases/JapaneseVowels/