Vol.2010-CSEC-50 No.47
2010/7/2

一般化階層木の自動生成と情報エントロピーによる歪度評価を伴う k-匿名化手法

原 田 邦 彦[†1]　佐 藤 嘉 則[†1]

k-匿名化手法は，個票データから同一タプルが k 件以上存在することを保証したデータを生成するプライバシ保護手法である．Sweeney[1] を代表とする既存手法は入力のデータとは別に一般化階層木を入力として与える必要があった．また，匿名化前後のデータ間の歪度を一般化階層木の深さを基準とした指標で与えていた．この指標では，データの失う情報量を正しく扱えない場合がある．本稿では，各属性の属性値の頻度情報を取得して一般化階層木をデータから自動生成する方法と，一般化を行う際のデータの歪度を頻度分布を用いた情報エントロピーで評価する方法を提案する．

k-anonymization Schemes with Automatic Generation of Generalization Trees and Distortion Measuring using Information Entropy

Kunihiko Harada[†1] and Yoshinori Sato[†1]

k-anonymization schemes are ones of the well-known methods to protect privacy. They generate anonymized data, each tupple of which appears at least k times in the whole records, from input microdata. In the existing schemes as typified by Sweeney[1], system users must prepare the generalization tree for each attribute in addition to the input microdata. Moreover, the schemes evaluate the distortion of data after anonymization by measurements based on the tree depths. There are cases the measurements cannot treat the information loss in an accurate way. In this paper, firstly, we propose k-anonymization schemes which automatically construct generalization trees by using frequencies of the attribute values. Secondly, We propose schemes to measure the distortion based on information entropy.

1. はじめに

近年，個人にまつわる膨大なデータの集積化に伴い，プライバシ侵害の危険性が増大してる．こういった社会背景に鑑みていわゆる個人情報保護法が施行され，個人情報を管理する企業等は収集・利用等の個人情報に関する適切な管理を義務付けられている．個人情報保護法の経産省ガイドライン[2] によれば，対応措置の一つに個人情報の匿名化を掲げている．ここでの匿名化とは，個人の識別情報（名前など）を非開示にすることを意味する．しかし，このように匿名化された個人情報ですら，いくつかの属性情報を組み合わせることによって個人が特定されてしまうことによるリスクが指摘されている[3]．

このリスクを回避する方法の一つに，Sweeney[1),4)] によって提案された k-匿名化 (k-anonymization) がある．

これまで提案されてきた各種 k-匿名化手法[1),5)–9)] はアルゴリズムの入力として，匿名化を行う対象データとは別に，属性値を一般化する構造を表す一般化階層木を各属性に対して与える必要があった．しかし，属性値の総数が大きいデータに対してこれを生成するのには，運用上のコストを必要とする．そこで本稿では，各属性値の頻度情報を利用して，自動的に木を構成する手法を提案する．

また，匿名化前後のデータ間の歪度として，文献 1)，5)，9) では，属性値の対応する一般化階層木の深さの差を基準とした指標を用いていた．しかし，この方法では頻度に偏りがある場合に直感に合致しない評価をしてしまう．そこで，本稿では頻度情報を用いた情報エントロピーをにより歪度を評価する手法を提案する．さらに，匿名化されたデータの既存アプリケーションへの適用可能性を向上させる方法に言及する．

2. 準　　備

本節では本稿で用いる用語と概念を定義する．

2.1 個票データと匿名化

文献 3) を参考に，本稿で扱う個票データと匿名化について整理する．ある 1 個体は住所や年齢など様々な属性情報を持つ．属性の取りうる値を属性値と呼び，ある 1 個体に対するいくつかの属性を説明する属性値のタプルを個票と呼ぶ．本稿では，同じ属性集合に対して

†1 （株）日立製作所 システム開発研究所
Systems Development Laboratory, Hitachi, Ltd.

複数個体の個票を集めたものを個票データと呼ぶ．個票データをテーブル形式で表現した場合，1 レコード（1 行）が各個票を意味し，1 カラムが各属性を表すことになる．

個票データを開示することにより，1 個票が現実世界（母集団）のどの個体に相当するかを特定されることを識別リスクと呼ぶ．匿名化とは，個体の持つ各属性値に一般化（曖昧化，generalization），欠損化 (suppression)，ノイズ付加 (adding noise) などの操作を加えることで識別リスクを低減させることを指す．このように，操作を加えた値に置き換えることを再符号化と呼ぶ．個票データを構成する属性のうち，氏名・住所・電話番号など個体を直接識別できるように設計されている属性を直接識別子 (identifier) と呼ぶ．個票データの匿名化においては，このような直接識別子は予め除外されていることが前提である．しかし，年齢・性別・職業などを組み合わせることで個体を特定できるようなケースは少なくない．このような属性を準識別子 (quasi-identifier) と呼ぶ．本稿で扱うリスクは準識別子を組み合わせて個体を特定されることである．したがって，以降本稿では簡単のために，個票データの属性は準識別子のみから構成され，それ以外は除外したものを指すものとする．

なお，住所の 1 属性値である「横浜市戸塚区吉田町２９２番地」を「横浜市戸塚区吉田町」にすることで特定可能性が大きく下がることからもわかるように，何が直接識別子，準識別子であるか，あるいはそのどちらでもないかは一概に決められるものではない．匿名化システムの利用者が，入力の個票データの機微度から判定を行うものである．

2.2 k-匿 名 化

識別リスクを低減させる匿名化手法の 1 つとして，一般化を用いた k-匿名化がある．一般化とは 2.1 節に示した住所の例のように，真の値ではあるがその情報を曖昧にすることを指す．個票データの安全性を量る指標として，次の k-匿名性 (k-anonymity) が提案されている．

定義 1 (k-匿名性[4])．個票データの中で，出現する各々のデータタプルによって表される個票が k 件以上存在するとき，個票データは k-匿名性を持つという．

一般化を行ってデータに k-匿名性を持たせることを k-匿名化と呼ぶ．

2.3 一般化階層木

一般化階層木は k-匿名化を行う前に各属性に対して 1 つずつ与える木であり，各属性値の再符号化候補をその情報量の順序に階層構造が従うよう木構造を用いて表したものである．**図 1** に 10 歳以上を対象とした年齢の一般化階層木の一例を示す．13 歳の属性値を持つ個票は，そのデータが “10–14”, “10 代”, “*” のどれかに一般化されうることを意味し，この順に従い曖昧になる．なお，“*” は欠損値を意味する．

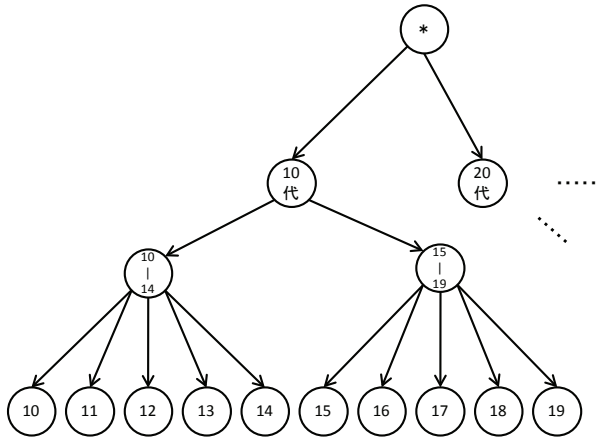

図 1 一般化階層木の例

最後に本節では，一般化階層木におけるいくつかの用語を定義する．葉 (leaf) とは子を持たない節点のことである．すなわち，再符号化される前の個票データの持つ属性値に相当する．根 (root) とは親を持たない節点のことであり，欠損値に相当する．内部節点とは葉でない節点を指す．節点 a と根を結ぶパスが経由する節点の個数を節点 a の深さと呼ぶ．ただし，根の深さは 0 である．本稿を通じて，内部節点にはその状態を示すラベル（再符号化値）が割り当てられているものとする．例えば図 1 の “10–14” といったものがラベルである．本稿では，内部節点のラベルを準属性値と呼ぶ．

2.4 歪 度 関 数

k-匿名化においては個票データの匿名性とその損失する情報量損失とのトレードオフを考えることが重要である．そこで，本節では損失する情報量を量る指標の満たすべき性質を考え，これを歪度関数として定義する．

定義 2 (歪度関数)．属性の一般化階層木を T とし，T を構成する節点集合を V_T とする．$v_1, v_2 \in V_T$ に対して v_2 が v_1 の祖先であることを $v_1 \triangleright v_2$ または $v_2 \triangleleft v_1$ で表すことにする．$v_1 \triangleright v_2$ を満たす (v_1, v_2) 全体の集合を $\Delta \subset V_T \times V_T$ とする．このとき，T に対する歪度関数 $f_T : \Delta \to \mathbb{R}$ は次の 2 つの性質を満たす関数である．ただし，$\mathbb{R}$ は実数全体の集合である．

(1)　非負性：$f_T(v_1, v_2) \geq 0, \forall (v_1, v_2) \in \Delta$

(2) 加法性：$v_1 \triangleright v_2 \triangleright v_3$ を満たす任意の v_1, v_2, v_3 に対して，$f_T(v_1, v_3) = f_T(v_1, v_2) + f_T(v_2, v_3)$ を満たす．

歪度関数は，ある個票の当該属性が $v_1 \in V_T$ に相当する属性値または準属性値である場合に，これをさらに $v_2 : v_2 \triangleleft v_1$ に相当する準属性値に再符号化したときにデータが曖昧になる程度を表す関数を意図している．非負性は，一般化を行えば必ず曖昧になる（情報を損失する）ことを意味する．なお，非負性と加法性により $f_T(v_1, v_1) = 0$ がすぐに導ける．これは，v_1 を v_1 に再符号化しても情報を損失しないことを意味する．また，加法性はある個票のある属性の属性値または準属性値が v_1 に相当するものであった場合に，これを $v_2 : v_2 \triangleleft v_1$ に再符号化した後，さらに $v_3 : v_3 \triangleleft v_2$ に再符号化した場合の合計の情報損失が，はじめから v_1 を v_3 に再符号化した場合の情報損失に等しいことを意味しており，歪度関数が自然に満たすべきものである．

k-匿名化アルゴリズムの出力データは，元の個票データとの間の歪度関数値の全属性および全個票にわたる総和が，k-匿名性を満たす任意の他の匿名化データよりも小さいことが望ましい．しかし，一般には計算量の問題から，局所最適な匿名データを出力したり，一般化を行う上での制約を加えることによったりするアルゴリズムが用いられる．

f_T を属性によって異なる関数で定義することも考えられるが，異なる属性をまたいで f_T を統一的に扱うことが困難であるため，一般化階層木の構造を除いて f_T は T に依らないものとし，以降単純に f で表す．

3. 提 案 手 法

これまで提案されてきた k-匿名化手法[1),5)–10)] は一般化階層木を別途用意して匿名化アルゴリズムに与える必要があった．そこで，本稿では入力の個票データから，各々の属性値の頻度情報を取得し，これを用いて各属性の一般化階層木を自動的に生成する過程を伴う k-匿名化を行う手法を提案する．3.1 節では手法の概要を説明し，3.2 節では一般化階層木の自動生成手法を説明する．

一方，Sweeney[1)] に代表される既存手法は，匿名化前後のデータ間の歪度を一般化階層木の深さを基準とした指標で与えていた．しかし，この指標では，データの失う情報量を正しく評価できない問題がある．3.3 節では，この問題を例を挙げて説明し，取得した頻度情報を利用して匿名化前後のデータの歪度を評価する手法を提案する．

3.1 手法の概要

一般化を行う指針は，大域的再符号化と局所的再符号化に分類される．

大域的再符号化 (global recoding) とは，個票データ全体を通じて，ある属性の 2 つ以上の属性値または準属性値を，共通する 1 つの祖先節点に対応する準属性値に再符号化することを指す．例えば，図 1 に示した年齢の一般化階層木を用いて k-匿名化を行った結果，大域的再符号化では，本来 13 歳であって 13 歳のままとなる個票と，本来 13 歳であったのに 10–14 歳に再符号化される個票が共存することがない．

一方，局所的再符号化 (local recoding) とは，個票データ全体ではなく，2 つ以上の個票について，ある属性の 2 つ以上の属性値または準属性値を，共通する 1 つの祖先節点に対応する準属性値に再符号化することを指す．したがって，前記の例では，本来 13 歳であって 13 歳のままとなる個票と，本来 13 歳であったのに 10–14 歳と再符号化される個票の共存が起こりうる．局所再符号化によって生成された匿名化データは適用するアプリケーションをよく吟味する必要があるが，大域的再符号化に比べてデータの歪度を小さくできることが期待できる．

本稿では文献 9) を参考に局所的再符号化を行う場合をアルゴリズム 1 に述べる．

アルゴリズム 1.
入力： 個票データ（属性数：m），歪度関数 $f(v_1, v_2)$
出力： 匿名化データ
Step 1: 個票データから各属性に対する一般化階層木 $T_1, T_2, \ldots, T_m$ を自動生成する．（手法は 3.2 節に記載.）
Step 2: データ D を入力の個票データに初期化する．
Step 3: D が k-匿名性を持つかどうかチェックする．k-匿名性を持つ場合には D を出力して終了する．
Step 4: k-匿名性を満たさないデータタプル A を 1 つランダムに選択する．
Step 5: A でない全てのデータタプル B に対して，仮に B と A を同じデータタプルとなるように再符号化した場合の歪度を次のように計算する．
(1) $c(A)$ をデータタプル A の頻度とする．
(2) 各属性 i に対して A の持つ属性値 a_i と B の持つ属性値 b_i を再符号化して同じ再符号化ができる属性値または準属性値を c_i とする．c_i は一般化階層木での a_i, b_i の共通の祖先のうち最も深いところにあるものである．
(3) 歪度を次式を用いて計算する．
$$c(A)\sum_i f(a_i, c_i) + c(B)\sum_i f(b_i, c_i)$$
Step 6: 上記 Step 5 で計算した歪度が最も小さいデータタプル B とデータタプル A を再符号化する．
Step 7: Step 3 に戻る．

本稿では，局所的再符号化にしか言及しないが，本稿で提案する一般化階層木の自動生成法および歪度関数は大域的再符号化にも局所的再符号化にも適用可能である．

3.2 一般化階層木の自動生成手法

3.2.1 葉の順序を保存する必要がない場合

葉（属性値）の順序を保存する必要がない場合には Huffman 符号木[11] を用いる．Huffman 符号木は無歪データ圧縮符号化に用いる符号木として提案されたもので，出現頻度の高い属性値を浅い階層に，出現頻度の低い属性値を深い階層に割り当てる．Huffman 符号木は，同じ属性値集合を持つ場合に考えられるあらゆる符号木の中で，当該頻度分布に対して葉の深さの期待値を最小にすることが知られている．Huffman 符号木構成アルゴリズムを用いて属性の一般化階層木を作成する方法をアルゴリズム 2 に述べる．

アルゴリズム 2 (葉の順序を保存しない一般化階層木の自動生成法[11])．
入力： 個票データ，一般化階層木を生成する属性の ID
出力： 指定した属性に対する一般化階層木
Step 1: 指定した属性の全ての属性値の頻度を個票データから数え上げる．
Step 2: 全ての属性値に対応する節点を作成し，キュー Q の頻度が小さい順になる場所に入れる．
Step 3: Q の要素が 1 個ならば，それを一般化階層木の根として出力し，処理を終了する．
Step 4: Q の最初の 2 個の節点 a,b を取り出し，Q から削除する．
Step 5: 新たに節点 c を作成して a,b を c の子とする．また，c の頻度を a,b の頻度の和とする．
Step 6: c を Q の頻度が小さい順となる場所に挿入する．
Step 7: Step 3 に戻る．

アルゴリズム 2 を用いて属性の一般化階層木を生成すれば，頻度の低い属性値には深い階層が割り当てられる．頻度が低い属性値を持つ個票ほど k-匿名性の障害となる可能性が高い．即ち，頻度が低い属性値に深い階層が割り当てられることにより，頻度の小さいもの同士を同じ準属性値に再符号化しようとする方向に k-匿名化アルゴリズムが動くため，過度の一般化を抑制する効果が期待できる．

注意 1. Huffman 符号木の代わりに，Shannon-Fano 符号木[12),13)] を用いる方法も考えられる．Huffman 符号木がボトムアップに木を構成する手法であるのに対し，Shannon-Fano 符号はトップダウンに構成するアルゴリズムである．一般に，Shannon-Fano 符号木よりも Huffman 符号木のほうが葉の深さの期待値が小さくなることが知られている．

注意 2. Huffman 符号木や Shannon-Fano 符号木が全二分木 [⋆1]であることに注意する．k-匿名化では，一般化階層木の全ての内部節点に再符号化されうるが，実用上の観点からは全二分木の場合にはそれが好ましくないことも考えられる．このような場合には，再符号化を行っても良い節点（ラベルを与える節点）のみを残した多分木へと変換すれば良い．

⋆1 全ての節点が葉であるか子を二つ持つ二分木のこと．

3.2.2 葉の順序を保存する必要がある場合

図 1 に示す例など，年齢等に代表されるように範囲を準属性値として指定したい場合や，できるだけ順序の近いものを優先的に一般化したい場合などには葉（属性値）の順序を保存した一般化階層木を構成した方が都合がよい．この制約を持つ符号木の中で，葉の深さの期待値を最小にする符号木を生成するアルゴリズムとして Hu-Tucker アルゴリズム[14] が知られている．本稿では，Hu-Tucker アルゴリズムを用いて一般化階層木を自動生成することを提案する．Hu-Tucker アルゴリズムを用いた一般化階層木の自動生成法をアルゴリズム 3 に述べる．

アルゴリズム 3 (葉の順序を保存する一般化階層木の自動生成法[14])．
入力： 個票データ，一般化階層木を生成する属性の ID
出力： 指定した属性に対する一般化階層木
Step 1: 指定した属性の全ての属性値の頻度を個票データから数え上げる．
Step 2: 全ての属性値に対応する節点を作成し，リスト L に入れる．L は属性値の順序の通りにしておく．
Step 3: L が要素を 1 つしか含まなければ Step 6 に行く．
Step 4: L から以下を満足するペアを見つけ出し，L から削除する．新たな節点を生成しペアをその節点の子とする．ペアの頻度の和を生成した節点の頻度とし，ペアが存在した L 内の位置に格納する．
(1) L の中で間に葉を含まない節点のペアを隣り合った節点と呼ぶ．
(2) 隣り合った節点ペア (a,b) において，a と隣り合った節点の中で最も頻度が少ないものが b で，b と隣り合った節点の中で最も頻度が少ないものが a である節点ペアが対象のペアである．
Step 5: Step 3 に戻る．
Step 6: L の 1 番目の要素を根とする木に対して，全ての葉の深さを取得する．
Step 7: Step 6 で取得した深さを利用して葉の順序を保存する木を生成し，これを出力として終了する．

3.3 情報エントロピーを用いた歪度関数

本節では，頻度分布から計算する情報エントロピーを用いて，一般化を行うときに損失する情報量を計算する歪度関数を与える．提案する指標は定義 2 の性質を満足する．

Sweeney[1] によって提案された元の個票データと匿名化された個票データとの間で曖昧

Vol.2010-CSEC-50 No.47
2010/7/2

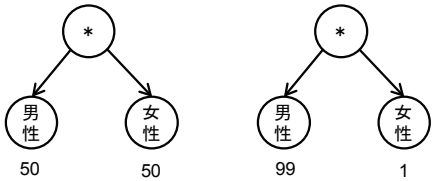

図 2 歪度評価の一例

になった程度を表す指標 *Prec* を参考に考えられた，村本－上土井－若林が提案した *DIS*[9] も定義 2 を満足する歪度関数である．*DIS* は一般化階層木の節点の深さの差をベースとした指標であるが，これを用いると例 1 のように損失する情報量を正しく評価できない場合がある．

例 1．図 2 は 2 つの 100 名の個票データの性別属性に関する一般化階層木である．左側は男性 50 名，女性 50 名から成る個票データから，右側は男性 99 名，女性 1 名から成る個票データから生成されたものである．*DIS* のように匿名化前後の深さの差を歪度とする．大域的再符号化を行い，これらのデータ全てが欠損値となった場合，いずれも $50+50 = 99+1 = 100$ の歪度となる．しかし，右側のようにデータの殆どが男性である場合にそれを欠損値としたところでデータの持つ情報量がほとんど変わらないのは感覚的に明らかである．対して左側は男女半々であるので，これら全てを欠損値とすることは大きな情報の損失となる．

上記の例に鑑みて，情報エントロピーを用いた損失情報量計量関数を歪度関数として用いることを提案する．

定義 3 (損失情報量計量関数)．一般化階層木 T の節点 v の頻度 $c(v)$ を次のように定義する．

(1) 節点が葉の場合にはその属性値の出現頻度とする．
(2) 節点が内部節点の場合には，その節点の全ての子の頻度の和とする．((1) により再帰的に定義される．)

このとき，T 内の節点 v_1 が表す属性値または準属性値を持つ個票 1 つを $v_2 : v_2 \lhd v_1$ が表す準属性値に再符号化した時の損失情報量計量関数 g を次のように定義する．ただし log の底は 2 とする．

$$g(v_1, v_2) = -\log \frac{c(v_1)}{c(v_2)}$$

このとき，損失情報量計量関数は次の定理を満たす．

定理 1． 損失情報量計量関数は歪度関数である．

証明．損失情報量計量関数 g が非負性と加法性を満足することを示せば良い．$v_2 \lhd v_1$ と節点の出現頻度の定義より $c(v_1) \leq c(v_2)$ を満足することは明らか．これにより，$g(v_1, v_2) \geq 0$ を満足する．等号成立の必要十分条件は $c(v_1) = c(v_2)$ である．すなわち，頻度 0 の葉の存在を認めず，全二分木を仮定すれば $v_1 = v_2$ が等号成立の必要十分条件である．以上で非負性が示された．加法性については次式より成立する．

$$\begin{aligned} g(v_1, v_2) + v(v_2, v_3) &= -\log \frac{c(v_1)}{c(v_2)} - \log \frac{c(v_2)}{c(v_3)} \\ &= -\log \left(\frac{c(v_1)}{c(v_2)} \cdot \frac{c(v_2)}{c(v_3)} \right) \\ &= -\log \frac{c(v_1)}{c(v_3)} = g(v_1, v_3) \end{aligned}$$

以上により，損失情報量計量関数は歪度関数である． □

例 2． 損失情報量計量関数を歪度関数として採用し，図 2 の例を再考する．左側の個票データを再符号化した場合の歪度は $-50 \cdot \log 50/100 - 50 \cdot \log 50/100 = 100$ である．一方で，右側の個票データを再符号化した場合の歪度は，$-99 \cdot \log 99/100 - 1 \cdot \log 1/100 \simeq 8.079$ であり，左側の方がより多くの情報を損失したと評価されることが分かる．

次に，女性 1 個票が局所的再符号化されて欠損値に置き換えられる場合の歪度を考える．左側の場合が 1 であるのに対し，右側はおよそ 6.644 となり，右側の個票データの方が「女性」というデータを重視されていることがわかる．

以上の例にも示されるように，損失情報量計量関数を用いることで，より精度の高いデータを生成できることが期待される．

4. 一般化後の属性値再割り当て法

一般化を用いた k-匿名化アルゴリズムが出力する匿名化データは，入力の一般化階層木の内部節点に割り当てられた準属性値を含む．例えば，人種と言う属性に対して，準属性値が {コーカソイド，モンゴロイド} といった集合値として与えられる．しかし，匿名化後のデータの適用アプリケーションがこのような集合値や範囲値に予め対応しているような場合は多くはない．そこで，これらの値を代表値に置き換えると言う対策が講じられる．一般には，最頻値・最小値・最大値・メディアン・平均値を用いることが考えられる．特に集合値に関しては，この中では最頻値に置き換えるしか方法がない．こういった背景から，本節で

は頻度分布を利用した，より柔軟な属性値の再割り当て方を提案する．

手法は次のようなものである．まず，一般化を行う際に，どの節点の情報がその節点の情報に再符号化されたかを示す頻度情報を，入力の一般化階層木の各節点に対応させて保存しておく．k-匿名化が終了した後に，この頻度分布に従って属性値をランダムに生成することで，その節点に再符号化された準属性値を置き換える．具体的には，節点に保存された分布に従い，節点をランダムに決定する．決定した節点が葉である場合には，対応する属性値に置き換える．内部節点の場合には，再度その節点に保存された頻度分布に従って次の節点をランダムに決定することを繰り返す．

この方法で生成した匿名データは，各属性の頻度分布が元の個票データとほとんど変わらないことから，有用性が高いことが期待できる．

5. 実　　験

本節では，アルゴリズム 1 に 3.3 節で提案した歪度関数を適用し，実行時間を評価した結果を記す．なお，参考までに結果には元の個票データの持つ情報エントロピーと，与えたパラメタを満たすように匿名化した場合に損失する情報エントロピーを記載する．

評価は CPU が Core 2 Duo E8400 (3.00GHz) ⋆1, メインメモリが 3GB の汎用 PC 上に，Java ⋆2にてプロトタイプを構成して行った．また，テストデータとなる個票データとしては，各属性値を正規乱数で生成し，整数値化した擬似データを用意した．各々のデータは 1 回のみの実行で測定した結果である．単位は処理時間に関してはミリ秒，情報エントロピーに関してはビットである．

属性数を 5，個票データの件数を 50,000 件に固定した場合に，各属性の属性値数およびパラメタ k を変化させた場合の結果を**表 1** に記す．各属性の属性値数を 100，個票データの件数を 50,000 件に固定した場合に，属性数を変化させた場合の結果を**表 2** に記す．属性数を 5，各属性の属性値数を 50 に固定した場合に，個票データの件数を変化させた場合の結果を**表 3** に記す．

6. 考　　察

実験結果から，以下の傾向を確認できる．まず，一般化階層木の構成に構成に要する時間

⋆1 Core 2 Duo は米国およびその他の国におけるインテルコーポレーションまたはその子会社の商標または登録商標です．

⋆2 Java は米国およびその他の国における米国 Sun Microsystems, Inc. の商標または登録商標です．

表 1　属性値の個数と処理時間及び損失情報エントロピーの関係

属性値数	k	葉の順序	木生成時間（読み込み時間）	総実行時間	元データエントロピー	損失エントロピー	損失割合
10	2	非保存	297 (297)	170,687	8.07E+05	4.24E+04	5.3%
		保存	281 (281)	170,109		4.24E+04	5.2%
	10	非保存	203 (203)	272,750		2.71E+05	33.6%
		保存	219 (203)	275,906		2.73E+05	33.8%
	20	非保存	219 (203)	276,500		3.65E+05	45.2%
		保存	219 (204)	279,391		3.66E+05	45.3%
100	2	非保存	313 (313)	942,469	1.64E+06	6.47E+05	39.4%
		保存	406 (390)	945,531		6.47E+05	39.4%
	10	非保存	219 (203)	1,073,531		1.18E+06	71.9%
		保存	250 (218)	1,073,797		1.18E+06	72.0%
	20	非保存	235 (219)	1,074,688		1.29E+06	78.3%
		保存	234 (218)	1,073,640		1.29E+06	78.3%
1000	2	非保存	312 (250)	1,448,375	2.47E+06	1.86E+06	75.3%
		保存	3,250 (250)	1,457,438		1.85E+06	75.1%
	10	非保存	328 (266)	1,438,125		2.02E+06	81.6%
		保存	3,250 (250)	1,460,829		2.02E+06	81.7%
	20	非保存	313 (250)	1,455,375		2.08E+06	84.4%
		保存	3,250 (250)	1,435,735		2.08E+06	84.3%

表 2　属性の個数と処理時間及び損失情報エントロピーの関係

属性数	k	葉の順序	木生成時間（読み込み時間）	総実行時間	元データエントロピー	損失エントロピー	損失割合
5	2	非保存	313 (313)	942,469	1.64E+06	6.47E+05	39.4%
		保存	406 (390)	945,531		6.47E+05	39.4%
10	2	非保存	359 (344)	2,615,797	3.29E+06	2.48E+06	75.5%
		保存	375 (343)	2,608,765		2.49E+06	75.6%

表 3　個票の件数と処理時間及び損失情報エントロピーの関係

個票数	k	葉の順序	木生成時間（読み込み時間）	総実行時間	元データエントロピー	損失エントロピー	損失割合
100	2	非保存	16 (0)	47	2.57E+03	1.82E+03	70.9%
		保存	15 (0)	47		1.76E+03	68.6%
	10	非保存	16 (0)	47		2.45E+03	95.3%
		保存	15 (0)	47		2.38E+03	92.5%
1000	2	非保存	31 (15)	531	2.76E+04	1.41E+04	51.1%
		保存	32 (32)	532		1.41E+04	51.1%
	10	非保存	31 (31)	594		2.35E+04	85.1%
		保存	31 (31)	594		2.37E+04	85.6%
10000	2	非保存	78 (78)	34,125	2.78E+05	1.04E+05	37.5%
		保存	78 (63)	33,984		1.04E+05	37.3%
	10	非保存	79 (79)	40,110		2.06E+05	74.1%
		保存	78 (63)	39,735		2.07E+05	74.4%
100000	2	非保存	453 (453)	2,784,796	2.78E+06	7.42E+05	26.7%
		保存	500 (485)	2,746,563		7.40E+05	26.6%
	10	非保存	454 (454)	3,415,079		1.74E+06	62.5%
		保存	469 (453)	3,374,094		1.74E+06	62.6%

Vol.2010-CSEC-50 No.47
2010/7/2

に関して述べる．一般化階層木の構成に要する時間は，ファイルの読み込みに用する時間を除いて個票の総数には依存しない．属性数と一般化階層木生成の実行時間の関係がほぼ線形関係であることも直感に合う．属性値の総数に対しても，今回のパラメタの取り方では，総実行時間に対して無視できる程度の実行時間で実行できる．今回は愚直な方法で実装したが，Huffman 符号木構成には属性値が頻度の重み順にソートされていること（$O(n \log n)$）を前提に時間計算量が $O(n)$ のアルゴリズムが知られており，Hu-Tucker 符号木構成に関しては $O(n^2)$ の時間計算量のアルゴリズムが知られている．ただし，n は属性値の総数である．個票として，属性値数がそれほど大きいものはそれほどあるとは考えられず，十分に現実的なものである．

損失情報エントロピーに関しては，葉の順序を保存する場合と保存しない場合で結果的にあまり変化がないのは，局所的再符号化を用いたためであると考えられる．大域的再符号化を用いた場合には，頻度の大きいものと頻度の小さいものを再符号化すると一度に大きく情報を損失すると考えられるため，Huffman 符号木のほうが有利に働くと予想できる．大域的再符号化を含めた評価は今後の課題である．

7. おわりに

本稿では第一に，k-匿名化を行う際に課題であった，一般化階層木を別途用意しなくてはならないという運用上の問題を，頻度情報を用いて一般化階層木を自動生成することで解決した．システムの運用者は自動生成された一般化階層木をそのまま用いても良いし，編集して都合のよいものに手直ししてもよく，運用コストの削減が見込める．さらに 4 節の方法を併用することで，既存アプリケーションの入力として適切なデータを生成できる．

第二に，情報エントロピーを用いた歪度関数を提案した．これにより，例 1，2 に示したように，直感にかなった歪度を与えることができた．実際，提案した指標は歪度の満たすべき自然な性質を満たすものである．

今回の実験評価では，局所的再符号化を用いて提案手法を評価したが，大域的再符号化にも問題なく適用可能なものである．今後の課題としては，実運用上でのアプリケーションへの適用可能性の評価，実データを用いた評価や大域的再符号化を含めた評価，属性間の相関を考慮に入れた手法の検討などがある．

参 考 文 献

1) Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, Vol.10, No.5, pp.571–588 (2002).
2) 経済産業省：個人情報の保護に関する法律についての経済産業分野を対象とするガイドライン (平成 21 年 10 月改正).
3) Takemura, A.: Current Trends in Theoretical Research of Statistical Disclosure Control Problem, *Institute of Statistical Mathematics*, Vol.51, No.2, pp.241–260 (2003).
4) Sweeney, L.: k-anonymity: A Model for Protecting Privacy, *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, Vol.10, No.5, pp.557–570 (2002).
5) Iyengar, V.: Transforming Data to Satisfy Privacy Constraints, *8th ACM SIGKDD International Conf. Knowledge Discovery and Data Mining (ICDM2004)*, pp.279–288 (2002).
6) Wang, K., Yu, P. and Chakraborty, S.: Bottom-up generalization: a data mining solution to privacy protection, *4th IEEE International Conf. Data Mining (ICDM2004)*, pp. 279–288 (2004).
7) Fung, B., Wang, K. and Yu, P.: Top-Down Specialization for Information and Privacy Preservation, *21st International Conf. Data Engineering (ICDE2005)*, pp.205–216 (2005).
8) LeFevre, K., DeWitt, D. and Ramakrishnan, R.: Incognito: Efficient Full-Domain K-Anonymity, *2005 ACM SIGMOD International Conf. Management of Data*, pp. 49–60 (2005).
9) 村本，上土井，若林：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).
10) Sweeney, L.: Guaranteeing anonymity when sharing medical data, the Datafly system, *Proceedings, Journal of the American Medical Informatics Association*, pp.51–55 (1997).
11) Huffman, D.: A method for the construction of minimum-redundancy codes, *Institute of Radio Engineers*, Vol.40, No.9, pp.1098–1102 (1952).
12) Shannon, C.: A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol.27, pp.379–423 (1948).
13) Fano, R.: The transmission of information, Technical Report 65, Reesearch Laboratory of Electronics at MIT (1949).
14) Hu, T. and Tucker, A.: Optimal computer search trees and variable length alphabetic codes, *SIAM Journal of Applied Mathematics*, Vol.21, No.4, pp.514–532 (1971).