Computer Security Symposium 2013
21 - 23 October 2013

効率的な集合値再符号化手法による複合データのk-匿名化

髙橋 翼† 側高 幸治† 竹之内 隆夫† 豊田 由起† 森 拓也†

†日本電気株式会社
211-8666 神奈川県川崎市中原区下沼部1753
t-takahashi@nk.jp.nec.com

あらまし パーソナルデータ，特に単一の値や集合値等が混在した複合データの二次活用によって，様々なサービスや研究の発展が期待されている．二次活用の際には，データ主体のプライバシ保護のためにk-匿名化等の匿名性保証が求められる．既存の集合値のk-匿名化は複合データのような多次元データのk-匿名化への適用や，効率性に課題がある．本稿では，複合データに対する効率的なk-匿名化を実現するために，複合データの匿名化に適用可能かつ計算効率を向上した集合値の再符号化手法を提案する．また，いくつかの計算機環境で提案手法の有効性を評価し，1億件規模のデータセットを数時間でk-匿名化できることを確認した．

Efficient Set-value Recoding for Anonymizing Complex Data

Tsubasa Takahashi† Koji Sobataka† Takao Takenouchi† Yuki Toyoda†
Takuya Mori†

†NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666, JAPAN
t-takahashi@nk.jp.nec.com

Abstract Complex data, which has various types of attributes, is expected to leverage in wide range of services and researches. In such utilizations, ensuring anonymity of data owners by data anonymization such as k-anonymization is required to preserve privacy of them. However, existing k-anonymization for set-values is difficult to apply multi-dimensional anonymization methods. This paper proposes an efficient set-value recoding which follows multi-dimensional k-anonymization framework for anonymizing complex data. In our evaluation, we measure the efficiency of proposed method in several computational environments. As a result, we confirm that proposed method anonymizes 100 million records in a half of day.

1 はじめに

診療履歴やサイト訪問履歴といったパーソナルデータが蓄積され，近年のビッグデータ活用のニーズの高まりから，パーソナルデータを他のサービスや事業に活用する二次活用の期待が高まっている．しかしながら，診療履歴等のパーソナルデータにはデータ主体に関する機微な情報が記録されているため，取扱いに際してはデータ主体のプライバシへの配慮が必要となる．特に二次活用に際しては，データ主体の匿名性を保証した上でパーソナルデータを提供することが求められる．

データ主体の匿名性を保証する技術として，パーソナルデータのうち個人を特定し得る属性(準識別子)を加工し，一定の匿名性を保証する

データ匿名化が研究されている．データセットが持つ匿名性の指標として，同一の準識別子の組を持つレコードが k 個以上出現することを表す k-匿名性 [1] が広く知られている．k-匿名性を充足させる処理である k-匿名化は，準識別子の汎化 (Generalization) やレコードの削除 (Suppression) といった加工を用いて実現される．このとき，加工による情報損失が生じ，データの有用性が低下する．よって k-匿名化では，匿名性を保証しつつ有用性を最大限維持することが望まれる．

さらに，ビッグデータに対する二次活用の期待の高まりから，大規模なパーソナルデータの匿名化が求められるようになってきた．k-匿名化はパーソナルデータのレコード数や属性数の増加に従って計算コストが増加する傾向にあるため，ビッグデータを対象にした k-匿名化には，高い効率性やビッグデータを前提とした計算機環境での動作が必要である．

一方で，パーソナルデータは様々な種類の属性が混在する複合データである．本稿では，年齢や性別などの一レコードに単一の値を取る単一値と，傷病名や処方薬などの一レコードに複数の値を取り得る集合値を含むパーソナルデータを対象とする．集合値は，単一値と比較すると k-匿名性を充足させるために，多くの汎化や削除が必要になる．特に複合データに対する k-匿名化では，すべての属性値の組み合わせに対して k-匿名性の保証が必要となりより多くの情報損失が生じ得る．よって，情報損失を低減するためには，集合値の加工は計算コストが小さい大域的な手法 [8] よりも，細やかなキャリブレーションが可能な局所的な手法によって実施されることが望ましい．

本稿では，単一値属性と集合値属性が混在する複合データを統一的に k-匿名化する問題を扱う．特に，集合値に対する効率的な局所的再符号化手法を提案し，複合データの効率的な k-匿名化を実現する．さらに，提案手法をカラムストア型 DWH と従来の RDB 上で提案手法の効率性を検証する．

本稿の以降の構成は以下の通りである．2 章では，本稿の関連研究と提案手法のベースとなる技術を紹介する．3 章にて，効率性を高めた集合値の再符号化手法を提案する．4 章では，提案手法の有効性について評価実験を通して検証する．最後に 5 章にて，本稿の結論を述べる．

患者ID	生年	性別	傷病名	薬剤名
1	1970	男性	A, B, C	a, b, d
2	1971	男性	A, B, C	a, f, g
3	1974	女性	D, E	a, d, f, y, z
4	1980	男性	D, E	a, b, c, f, g
5	1960	女性	A, D	b, c, f
6	1999	女性	E, F	c, e, x
7	1982	男性	E, F	b, e, x
8	2001	女性	A, D	b, c
9	1984	男性	E, F	c, e, x

図 1: 複合データ

2 関連研究と準備

本稿で対象とする複合データ T は，属性としてレコード識別子 ID と準識別子 $A_i, \ldots, A_d$ を持つ，タプル形式 $(t = (id, a_1, \ldots, a_d))$ のレコードの集合である．ここで，a_i を属性 A_i の属性値である．$(a_i \in A_i)$．属性 A_i は単一値属性もしくは集合値属性である．単一値属性は，すべてのレコードの属性値がサイズ 1 である $(|a| = 1, a \in A)$．集合値属性は，アイテムの集合からなる集合値 $a = \{\alpha_1, \ldots, \alpha_m\}$ を属性値として持ち，そのサイズはレコード毎に異なる．

2.1 k-匿名化

Sweeney は，同一の準識別子の組を持つレコードが k 個以上存在するというレコード識別の困難さを表す k-匿名性を提案した [1]．データセットに対して k-匿名性を充足させるための加工を k-匿名化と呼ぶ．k-匿名化では，属性値の加工によってデータの有用性に損失が生じる．数値属性に対する汎化は，元の属性値を包含する属性値への加工である．

例えば，生年「1974」は「1970-1974」に汎化される．非数値属性に対しては，元の属性値の上位概念である値への加工である．例えば，性

別「男性」は上位概念である「Any」へと汎化される．

上述のような汎化によって匿名化が実現される．しかし，過度な属性値の加工は情報損失を生み，データの利用価値を棄損する．そのため，匿名性を保証しつつ有用性を最大限維持することが望まれる．有用性の指標として，属性値の汎化度合いの小ささを表すNCP(Normalized Certainty Penalty)[4] 等がある．また，有用性を最大化する最適な k-匿名化は，NP-困難な問題 [2] として知られている．

k-匿名性を拡張した匿名性の指標として，ℓ-多様性 [9] や t-近接性 [10]，m-不変性 [11]，Pk-匿名性 [13] 等が知られており，想定する攻撃や保護指針に併せて様々な指標を用いることができる．

本稿でテーブル T に対して充足させる k-匿名性を以下のように定義する．

定義 1 *(*複合データに対する k-匿名性*)* 準識別子の集合を $QI = \{A_1, \ldots, A_d\}$ とする．任意のレコード $t \in T$ に対して，同一の準識別子の値の組 $\{a_1, \ldots, a_d\}$ を持つ他のレコードが $k-1$ 以上存在しているとき，複合データ T は k-匿名性を満たす．

これは，A_i に集合値が含まれること以外は，[1] の k-匿名性と等価である．

2.2 トップダウンアプローチ

複数の単一値属性の加工を共存させながら k-匿名化するアプローチとして，トップダウンアプローチが知られている．トップダウンアプローチによる k-匿名化には，Top Down Specialization[3] やMondrian[5] といった貪欲手法が知られている．

トップダウンアプローチでは，まず，最も一般化された状態に初期化し，テーブルを k-匿名化された状態に加工する．その後，属性の有用性指標に基づいて属性値の詳細化を行い，有用性を向上させる．このとき，詳細化によって匿名性が減少する．トップダウンアプローチでは，所望の匿名性を満たす間詳細化を繰り返し，高い有用性を持つ匿名化データセットを生成する．

Algorithm 1 Top Down Anonymization(T)

```
1: T* ← initialize(T)
2: while |QI*| ≥ 1 do
3:    T*_tmp ← T*
4:    A_div ← choose_dimension(QI*)
5:    T* ← specialize(A_div)
6:    if T* is not k-anonymous then
7:       T* ← T*_tmp
8:       QI* ← QI*\{A_div}
9:    end if
10: end while
```

有用性の高い匿名化テーブルを探索するためには，匿名性を満たす匿名化テーブルを探索し，有用性の高いものを選択する必要がある．トップダウンアプローチでは，匿名性を満たす匿名化テーブルを探索する必要がなく，詳細化処理による有用性の向上 (情報利得) と匿名性違反の有無だけを考慮すればよい．よって，一度に考慮する情報の組合せが小さく，複合データのような高次元データの匿名化に適している．また，属性ごとに優先度を与えてバイアスをかけた手法も提案されている [12].

本稿では Algorithm1 に示したトップダウンアプローチの匿名化アルゴリズムにより，効率的な複合データの匿名化の実現を目指す．本アルゴリズムでは，有用性指標として汎化の度合いによって情報損失の大きさを表す NCP を用いる．以下に各ステップについて概説する．

1. 各属性を最も一般化した状態に初期化し，1つのクラスタを生成．(Line 1)
2. 情報損失 (NCP) が最大の属性を詳細化の対象として選択．(Line 4)
3. 属性固有の詳細化手法を用いて対象属性を詳細化し，クラスタ分割 (Line 5)
4. クラスタ分割後，k-匿名性を検証．k-匿名性に違反したら，対象属性を詳細化不可とし，ロールバック．(Line 6〜9)
5. 全ての属性が詳細化不可になるまで，2.〜4. を繰り返す．

2.3 集合値属性の k-匿名化

一般に，集合値属性の k-匿名化では，非数値のアイテムの集合から成る集合値 (またはトランザクション) を属性値として想定する．k-匿名性が保証されるためには，同一の集合値を持つレコードが k 以上存在することが求められる．これは，単一値属性の k-匿名化と同様に，属性値の一般化や削除を用いて実現する．また，集合値に対する匿名性の指標として k^m-匿名性が提案されている [6].

定義 2 *(k^m-匿名性 [6])* 任意のレコード $t \in T$ の属性 A の集合値において，任意の m-アイテム集合と同一のアイテム集合を持つレコードが *k-1* 以上存在するとき，T は A において，k^m-匿名である．また，$m = \infty$ のときはすべての組合せを考慮する．よって，$m = \infty$ のときは k-匿名性と同一である．

集合値に対して k-匿名性や k^m-匿名性を充足させる手法がいくつか提案されている．

Terrovits らは，属性値の概念階層を定義した一般化階層を用いて，アイテムを一般化することで k-匿名化する手法を提案している [6]．この手法は，ある同一のアイテムのすべてを同じ値に一般化 (大域的再符号化) である．

He らは，一般化階層を用いて，特定のレコードの特定のアイテムだけを一般化 (局所的再符号化) することで k-匿名化する手法を提案している [7]．この手法では，アイテムの一般化を局所的範囲に限定することができ，情報損失を抑制することができる．

Xu らは，k-匿名性の違反の要因となるアイテムをテーブルから削除することで，一般化階層を用いずに k-匿名化を実現する手法を提案している [8]．この手法では，あるアイテムがあるレコードの k-匿名性違反の要因となったとき，当該アイテムをすべてのレコードから削除する．

以上のように，既存の集合値属性の k-匿名化手法は他の属性と共存した統一的な k-匿名化に適していない．本研究では，複合データに対する k-匿名化を実現するために，トップダウンアプローチに適用可能な集合値の再符号化を取り扱う．トップダウンアプローチに適用するためには，集合値の段階的な再符号化によって，他の属性に対する再符号化と連動しながら高い有用性を持つ匿名化テーブルを生成することが求められる．

2.4 トップダウンアイテム集合再符号化

我々の先行研究では，複合データのトップダウンアプローチによる k-匿名化を実現するために，トップダウンのアイテム集合再符号化を提案している [14].

先行研究では，まず集合値中のすべてのアイテムが秘匿された状態に加工し，1 つの初期クラスを作る．集合値属性が詳細化の対象となったら，各クラスから頻度の高いアイテムを 1 つ選択し，選択したアイテム (分割基準アイテム) を含むレコード群からなるクラスと含まないレコード群からなるクラスへと分割する (図 2)．このとき分割基準アイテムを含むクラスが k-匿名性を充足していれば，分割基準アイテムを開示し，違反した場合には分割前の状態に戻す．

この手法では，分割基準アイテムを含むレコード群だけに対して開示処理が行われる．そのため，匿名化の各ステップにおいてレコード毎に集合値の詳細化の度合いが異なり，開示されたアイテムの数量や情報損失に大きな差が生じる．このような状態において，他の属性に対する詳細化処理によって k-匿名化が収束すると詳細化の機会を活かせず，集合値が十分に詳細化されない可能性がある．また，一度の詳細化でアイテムの開示が試行されるレコードは少数である．よって，テーブル全体が十分に詳細化されるまでに多数回のイテレーションを要し，集合値に対する再符号化処理に多くの計算時間を費やしてしまう．

3 提案手法

3.1 基本方針

先行研究の再符号化手法の課題を解決するために，すべてのレコードから少なくとも 1 つのアイテムに対して詳細化が試行されるような再符号化を考える．一度の詳細化ですべてのレコー

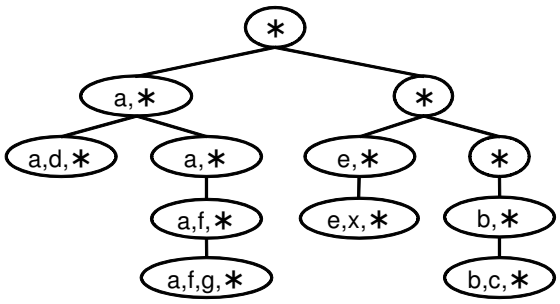

図 2: 先行研究のトップダウン集合値再符号化

ドに対してアイテム開示が試行されることで，集合値の情報損失が低減される可能性が先行研究の手法よりも高まる．また，一度に多くのアイテムを開示できることで詳細化の効率性も高まる．

先行研究の手法においても，複数回の詳細化処理を一度の再符号化と捉えることで一度に多くのアイテムの開示が実現できる．この手法では，分割基準アイテムの有無によって 2 つのクラスへの分割が行われる．この分割を分割基準アイテムを含まないクラスに対して再帰的に実施することで，すべてのレコードに対する少なくとも 1 回のアイテム開示の試行が可能である．しかし，最大でアイテムの種類数回の詳細化が行われる可能性があり，計算効率に対する課題が依然として存在する．

本稿では，一度の詳細化ですべてのレコードのアイテム開示が試行できるような手法について提案する．

提案手法では，先行研究の再符号化手法と同様に，トップダウンアプローチによる再符号化を行う．初期状態は先行研究の再符号化手法と同様に，すべてのアイテムを秘匿状態にすることで行う．集合値属性が詳細化の対象として選択された際には，以下のステップにて再符号化を行う．

1. アイテムのクラス内ランキング
2. 分割基準アイテムの選択
3. クラスの分割

3.2 アイテムのクラス内ランキング

有用性の高い匿名化テーブルを生成するためには，それを生成し得るアイテムから順に開示されていくことが望ましい．特に本稿では，開示アイテム数を有用性指標とし，分割基準アイテムの候補をランキング形式で抽出する．

まず，クラス毎に開示の対象と成り得る複数のアイテムを絞り込む．そのために，各アイテムを分割基準アイテムとした際に，どれだけの情報損失の低減 (情報利得) が得られるかをアイテム毎に評価してランキングを作成する．

アイテムの評価は，クラス中のアイテムの出現頻度 $f(\alpha)$ によって行う．

$$f(\alpha) = |\{t \in c | \alpha \in t.A\}| \tag{1}$$

ランキングは $f(\alpha)$ の高さの降順で作成する．なお，本稿では多くのアイテムを開示することを目的としているため，出現頻度によってアイテムを評価しているが，他の基準によって情報損失を規定する場合には，他の評価基準によってアイテムを評価し，ランキングを作成すればよい．

評価値の高いアイテムが開示されるほど，有用性の向上が期待できる．よって，ランキング上位のアイテムほど，分割基準アイテムに適したアイテムと考えることができる．

クラス中の出現頻度が極度に小さいアイテムが分割基準アイテムとなると，たとえ k-匿名性を充足できたとしても極度に小さなクラスを生成してしまい，他の属性の詳細化を阻害してしまいかねない．そこで，頻度がクラス c のレコード数の β% 未満のアイテムはランキングから削除する．このアイテムの枝刈りの閾値 θ は以下で与えられる．

$$\theta = max(\beta|c|, k) \tag{2}$$

ランキングにおいて出現頻度が θ 以上のアイテムを分割基準アイテムの候補とする．ランキングに含まれるアイテムの集合を R とし，各アイテムの順位を $rank(\alpha)$ とする．

3.3 分割基準アイテムの選択

次に，各レコードの集合値から最も有用性の向上が期待できるアイテムを 1 つ選択する．

各レコードは，所属するクラスのランキングを参照し，詳細化対象の集合値中のアイテム群から，ランキング最上位のアイテムを選択する．このアイテムが開示の候補 (開示候補アイテム p) となる．

$$p(t.A) = arg\ min_{\alpha \in R \cap t.A} rank(\alpha) \quad (3)$$

3.4 サブクラスの生成

最後に，各レコードの集合値に対して詳細化 (アイテムの開示) を行う．

まず，クラス内において，開示候補アイテムが同一であるレコード群からサブクラスを生成する．詳細化後に k-匿名性を充足するためには，開示候補アイテムが同じレコードが k 以上存在する必要がある．そこで，開示候補アイテムごとにレコード数を計数し，k-匿名性を満たし得るかを判定する．

レコード数が k 未満の開示候補アイテムが存在するとき，それらのレコード数の合計値が k 以上であれば，レコード数が k 以上の開示候補アイテム群によるアイテムの開示が k-匿名性を満たす．このとき，レコード数が k 以上の開示候補アイテム群によってアイテムの開示を行い，開示候補アイテムごとに新たなクラスを生成する．

一方，レコード数が k 未満の開示候補アイテムのレコード数の合計値が k 未満のときは，k-匿名性に違反するレコード群が生じる．k-匿名性違反が生じる場合には，対象のレコード群を削除可能な場合は削除して，他のレコード群に対するアイテム開示を行う．削除が必要なレコード数が削除許容レコード数を超えない範囲で削除を許容する．削除が可能でない場合には当該クラスの詳細化を行わず，当該クラスを詳細化不可とする．

本稿では，詳細化処理において総レコード数 $|T|$ の ϵ% までのレコードの削除を許可し，削除レコード数が $\epsilon|T|$ を超えたらレコードの削除を許可しない．

4 評価

本稿の提案手法の有効性を評価するために評価実験を行った．評価実験では，集合値の開示アイテムの割合，情報損失 (NCP)，実行時間を計測した．

4.1 評価環境

4.1.1 評価用データセット

評価用のデータセットとして，株式会社日本医療データセンター[1]が提供するレセプトデータと，人工的に生成したレセプトデータを用いた．前者は約 400 万件，後者は 100 万〜1.5 億件のレセプトである．匿名化対象のテーブルは，(レセプト ID, 患者 ID, 生年，性別，診療年月, ベッド数, レセプト種別, 診療開始日, 傷病名, 薬剤名) を属性として持つ．レセプト ID は主キー，患者 ID はデータ主体識別子であり，これら以外の属性を準識別子とした．生年，性別，診療年月, ベッド数, レセプト種別, 診療開始日は単一値であり，傷病名, 薬剤名は集合値である．

4.1.2 評価用計算機

2 つの計算機で評価を行った．一方は匿名化アルゴリズムを制御するサーバ (匿名化サーバ) と，データの格納と加工を担当する DWH という構成を用いた．匿名化サーバには仮想マシン (CPU4 コア，メモリ 32GB) を用い，DWH には IDA(InfoFrame DWH Appliance)[2] と PostgreSQL を用いた．IDA は大規模データ向けのカラムストア型の DWH である．もう一方は，匿名化サーバ上に PostgreSQL をインストールした環境である．

4.2 開示アイテム数

提案手法の有効性を評価するために，集合値属性の開示アイテムの割合を計測した．

[1] http://www.jmdc.co.jp/
[2] http://jpn.nec.com/infoframe/dwhappliance/

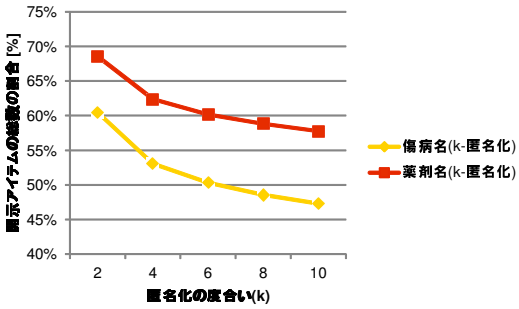

図 3: 開示アイテム数の割合

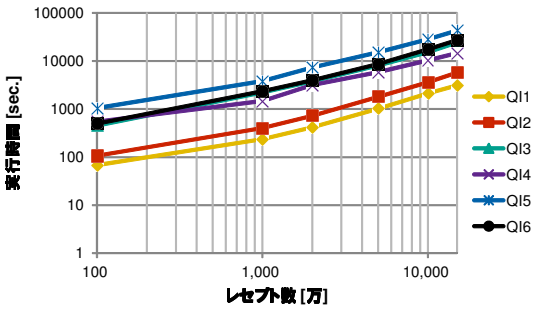

図 4: 実行時間 (IDA)

集合値属性である傷病名と薬剤名に対して，k=2, 4, 6, 8, 10 で匿名化した際に開示アイテムの割合を計測した結果を図 3 に示す．いずれの集合値属性も，k の値の増加に従って削除アイテム数は増加し，開示アイテム数が減少している．

4.3 匿名化処理時間

最後に提案手法の効率性を評価するために，匿名化に要した処理時間を計測した．ワーストケースに近い計算時間を計測するために，すべての準識別子をオリジナルの状態まで詳細化するように匿名化アルゴリズムを修正して実行時間の計測を行った．

人工的に生成した模擬レセプトデータを対象とし，IDA を用いて匿名化を実行した際の実行時間を図 4 に示す．この評価では，準識別子のパターンとして以下の組み合わせを用いた．

- QI1: 生年, 性別, 診療年月
- QI2: 生年, 性別, 診療年月, ベッド数, レセプト種別, 診療開始日
- QI3: 傷病名
- QI4: 薬剤名
- QI5: 傷病名, 薬剤名
- QI6: 生年, 性別, 診療年月, 傷病名

100 万，1000 万，2000 万，5000 万，1 億，1 億 5000 万件のデータセットを用いた．

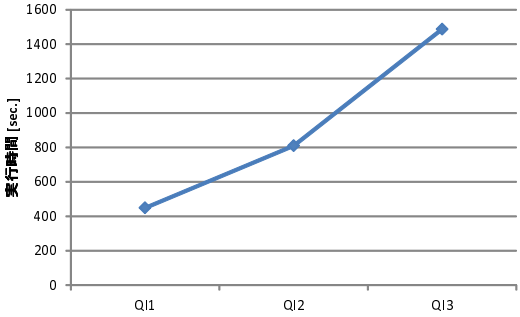

図 5: 実行時間 (PostgreSQL)

次に，100 万件の JMDC データに対して PostgreSQL を用いて k-匿名化した際の実行計算時間を図 5 に示す．JMDC データに対しては以下の 3 種類の準識別子のパターンを対象とした．

- QI1: 生年, 性別, 診療年月
- QI2: 生年, 性別, 診療年月, 傷病名
- QI3: 生年, 性別, 診療年月, 傷病名, 薬剤名

図 4 より，100 万件程度の規模であればいずれの準識別子の組み合わせパターンに対しても，数分程度で匿名化可能であることがわかる．また，データセットの規模が増加してもほぼ線形に近い計算時間の増加傾向であることがわかる．さらに，本評価で用いたデータセットでは，1 億件を超える大規模なデータセットに対しても 1 時間から 10 時間程度で匿名化を完了できることが確認できた．

図 4 と図 5 を比較すると，IDA を用いた場合の実行時間は PostgreSQL を用いた場合と比較して 5 倍以上高速であることが分かる．仮に

PostgreSQL を導入した計算機上で 1 億件程度のデータセットを効率よく扱えたとしても，1 億件程度のデータセットの匿名化に少なくとも数日を要することが類推される．

なお，先行研究では 10 万件程度のデータセットの匿名化に数時間を要しており，それと比較しても提案手法は非常に高速である．以上より，トップダウンの再符号化手法としての効率性は改善されたと言える．

5 まとめ

本稿では，複合データに対する k-匿名化の問題を扱い，集合値を効率良く再符号化する手法を提案した．評価実験では，提案手法の実行時間を計測した．高性能な DWH を用いた際に，1 億件程度のデータセットを 1 時間〜数時間で匿名化可能であることを確認した．今後の課題として，大規模データに対するスケーラビリティを維持しながら，高い有用性を維持可能な k-匿名化の実現が挙げられる．

参考文献

[1] Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5), pp. 555–570 (2002).

[2] Meyerson, A. and Williams, R.: On the Complexity of Optimal K-Anonymity. Proc. $PODS$2004, pp. 223–228 (2004).

[3] Fung, B.C.M., Wang, K. and Yu, P.S.: Top-down specialization for information and privacy preservation. Proc. $ICDE$2005, pp. 205–216 (2005).

[4] Xu, J., Wang, W., Pei, J., Wang, X., Shi, B. and Fu, A.: Utility-based anonymization using local recoding. Proc. $SIGKDD$2006, pp.785–790 (2006).

[5] LeFevre, K., DeWitt, D.J., and Ramakrishnan, R.: Mondrian multidimensional k-anonymity. Proc. $ICDE$2006 (2006).

[6] Terrovits, M. Mamoulis, N. and Kalnis, P.: Privacy Preserving Anonymization of Set-valued Data. Proc. $VLDB$2008 (2008).

[7] He, Y. and Naughton, F.: Anonymization of set-valued data via top-down, local generalization. Proc. $VLDB$2009 (2009).

[8] Xu, Y., Wang, K., Fu, A. and Yu, P. S.: Anonymizing Transaction Databases for Publication. Proc. KDD2008 (2008).

[9] Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M.: ℓ-Diversity: Privacy Beyond k-Anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD), (2007).

[10] Li, N., Li, T. and Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. Proc. ICDE 2007, pp. 106–115, (2007).

[11] Xiao, X. and Tao, Y.: m-invariance: towards privacy preserving re-publication of dynamic datasets. Proc. $SIGMOD$, pp.689–700, (2007).

[12] Kiyomoto, S., Miyake, Y. and Tanaka, T.: Privacy Frost: A User-Oriented Data Anonymization Tool. Proc. $AREAS$, pp. 442–447, (2011).

[13] 五十嵐大, 千田浩司, 高橋克巳: k-匿名性の確率的指標への拡張とその応用例. CSS2009 (2009).

[14] Takahashi, T., Sobataka, K., Takenouchi, T., Toyoda, Y. and Mori, T.: Top-down Itemset Recoding for Releasing Private Complex Data. Proc. PST, pp. 373–376, (2013).