集合値の一般化階層なし再符号化による複合データの k-匿名化

高橋　翼† 側高　幸治† 竹之内隆夫† 豊田　由起† 森　拓也†
興梠　貴英††

† 日本電気株式会社 情報・ナレッジ研究所　〒 211–8666 神奈川県川崎市中原区下沼部 1753
†† 東京大学医学部附属病院 健康医科学創造講座　〒 113–8655 東京都文京区本郷 7-3-1
E-mail: †t-takahashi@nk.jp.ne.com

あらまし　データセット中のデータ主体のプライバシ保護手段として，k-匿名化が知られている．単一値や集合値を持つ複合データを k-匿名化する際には，匿名性と情報損失の両方を属性間で統一の基準で評価し，効率的に属性毎の再符号化を実施する必要がある．これまで，多次元の単一値向けにはトップダウンアプローチが効率的な k-匿名化手段として知られてきた．複合データでも同様にトップダウンアプローチを採用するためには，集合値のトップダウンアプローチに適合可能な再符号化手法が必要となる．そこで，本稿では，集合値向けのトップダウンアプローチの再符号化手法を提案する．特に，既存の多くの集合値再符号化手法で必要とされてきた一般化階層が存在しない場合でも，情報損失を抑えながら効果的に再符号化可能な手法を提案する．これによって，複合データに対しても，プライバシ侵害の懸念を排除したデータ活用が実現可能となる．評価実験を通して，本手法の有効性を示す．

キーワード　k-匿名化, 集合値匿名化, 複合データ, 多次元データ

Multi-dimensional k-anonymization for Set-valued Data without Generalization Hierarchy

Tsubasa TAKAHASHI†, Koji SOBATAKA†, Takao TAKENOUCHI†, Yuki TOYODA†, Takuya MORI†, and Takahide KOHRO††

† Knowledge Discovery Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211–8666 Japan
†† Department of Translational Research for Healthcare and Clinical Science, University of Tokyo
7–3–1 Hongo, Bunkyo-ku, Tokyo, 113–8655 Japan
E-mail: †t-takahashi@nk.jp.ne.com

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情報が，サービスを受ける度に蓄積されている．近年，ビッグデータ活用のニーズが高まり，これらの蓄積されたパーソナル情報を他のサービスや事業に活用する二次活用の期待も高まっている．しかしながら，パーソナル情報にはデータ主体に関する機微な情報が記録されている場合が多いため，二次活用の際にはデータ主体のプライバシへの配慮が必要となる．特に診療履歴やレセプトのような医療情報は，機微性が非常に高く，積極的な二次活用が行われていない．

一方，データ主体のプライバシを保護するために，パーソナル情報のうち個人を特定し得る属性 (準識別子) を加工する技術であるデータ匿名化が研究されている．データ匿名化は，データセットに所望の匿名性の指標を充足させる処理である．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加工処理である k-匿名化は，準識別子の一般化 (Generalization) や削除 (Suppression) といった加工を用いて実現される．このとき，加工されたデータセットには情報の曖昧化や欠落といった情報損失が生じ，情報損失を小さく抑えながら所望の匿名性を保証することが望まれている．

また，パーソナル情報には，レコード毎に単一の値を取る単

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

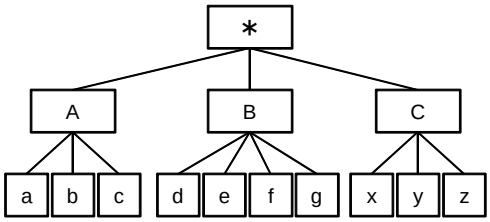

図 2　一般化階層

一値属性と，1 つ以上の値を取り，集合値 (トランザクション) 形式で表現される集合値属性が含まれる場合がある (図 1)．単一値属性の例として，生年や性別などが該当し，集合値属性の例として，傷病名や薬剤名などが該当する．レセプトデータやマーケットバスケットデータではこれらが共に含まれていることで，年代別の罹患頻度の分析や，購買傾向の分析等がデータマイニングによって実現できる．

既存の研究では，1 つ以上の単一値属性を含むパーソナル情報の匿名化と，1 つの集合値属性のみからなるパーソナル情報の匿名化技術が提案されていきた．しかしながら，それらが混在する複合データであるパーソナル情報の匿名化は広く研究されていない．

様々な属性が混在する複合データの k-匿名化は，多次元データの匿名化であり，組合せ爆発等を生じさせずに効率良く匿名化する必要がある．トップダウンアプローチは，多次元データを効率良く匿名化するためのフレームワークであり，考慮すべき加工パターンの組合せ爆発を生じさせずに匿名化を実現できる．しかし，既存の集合値属性の k-匿名化手法は，属性値中のアイテムの概念階層を定義した一般化階層 (図 2) が必要であり，一般化階層のない属性をトップダウンアプローチで k-匿名化することができない．

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，一般化階層が定義されていない集合値属性を，トップダウンアプローチで k-匿名化する手法を提案し，一般化階層が定義されていない集合値属性も，他の属性と共存した k-匿名化を実現する．

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，一般化階層のない集合値属性の k-匿名化手法を提案する．4 章では，4 章で提案した集合値属性の匿名化を，他の属性と共存させた k-匿名化手法を提案する．5 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 6 章にて，本稿の結論を述べる．

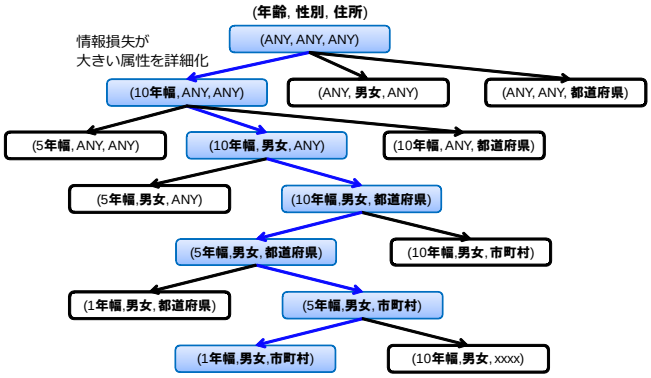

図 3　トップダウンアプローチ

2.　関連研究と準備

本稿で対象とする複合データ T は，レコード識別子と d 次元の準識別子を持つレコード ($t = \{id, a_1, \ldots, a_d\}$) の集合とする．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-匿名化と呼び，属性値の一般化や削除などが用いられる．数値属性に対する一般化は，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」に一般化される．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」へと一般化される．最適な k-匿名化は，NP-困難な問題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，従来の k-匿名性と同一である．

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
```

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 (図3)．トップダウンアプローチによる k-匿名化には，Top Down Specialization [2] や Mondrian [4] 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，属性毎に情報損失指標に基づいて詳細化を行う．このとき，詳細化によって匿名性が減少する．k-匿名性に違反したら詳細化を終了し，違反する前の状態にロールバックする．よって，詳細化処理が続く間は，常に k-匿名性が満たされているため，各詳細化は有効性の指標のみを考慮すればよい．一度に考慮する情報の組合せが小さいため，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]．

トップダウンアプローチとは異なり，元の属性値から徐々に一般化することで k-匿名化するボトムアップアプローチがある．ボトムアップアプローチでは，一般化によって必ずしも k-匿名性が満たされるとは限らない．そのため，k-匿名性を充足しつつ有効性の指標が高い一般化パターンを探索する必要がある．この探索はトップダウンアプローチと比較すると非常に深く，計算コストが高いため，高次元データの匿名化には適しているとは言い難い．

以上より，本稿では以下のようなトップダウンアプローチにより，効率的な複合データの匿名化の実現を目指す．

（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） 情報損失指標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．

本稿におけるトップダウンアプローチの k-匿名化アルゴリズムを Algorithm1 に示す．

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 [5]．

［定義 2］ (k^m-匿名性 [5]) 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として，いくつかの手法が提案されている．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 [5]．この手法では，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) する．匿名化されたデータの情報損失は大きいが，アイテムの一般化の粒度を均一化でき，集計や分析が容易である．大域的再符号化によるアイテムの加工では，他の属性を含む複合データでは，多くのデータが加工されやすく情報損失が大きい．Terrovits らの手法を改良し，情報損失を低減した手法も提案されている [8]．

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 [6]．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．その一方で，元の属性値が同じアイテムであっても，レコード毎に異なる値に一般化される可能性があるため，集計や分析が困難になるというデメリットもある．また，この方式は再帰的なトップダウンアプローチを採用しており，深さを優先した詳細化処理が行われるため，他の属性の詳細化処理と共存させることが困難である．

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 [7]．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．そのため，情報損失は非常に大きいが，残存するアイテムの真実性は完全に保たれる．また，ボトムアップアプローチであるため，他の属性を含む k-匿名化との共存が困難である．

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．他の属性の詳細化と公平に詳細化を行うためには，各属性を一段階ずつ段階的に詳細化可能な集合値属性の詳細化手法が必要で求められる．本稿では，複合データの複数の属性値を効率良く再符号化可能な匿名化の実現を目指す．特に，多くの集合値属性で適用可能とするために，一般化階層が与えられてない前提における再符号化を扱う．

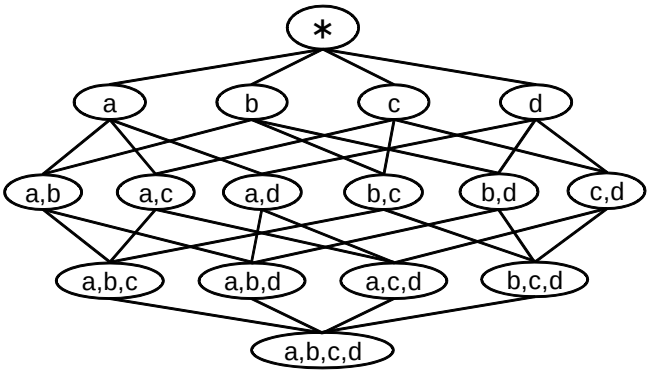

図 4　集合値のアイテム集合に基づく階層 (束)

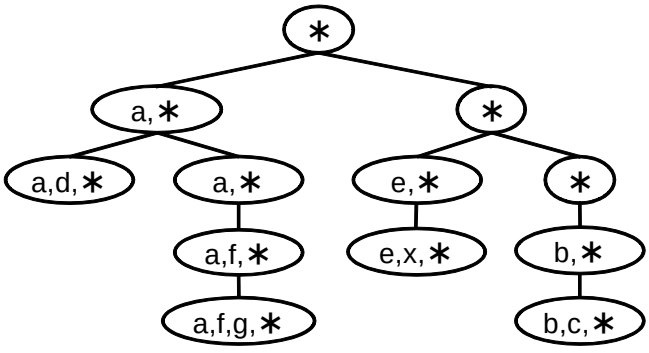

図 5　アイテムの共起関係に基づく集合値の再符号化

3. 集合値の一般化階層なし再符号化

集合値は，集合値を成すアイテム集合の種類によって，図 4 のような概念階層を考えることができる．ルートノードである「*」は，すべてのアイテムを削除した状態であり，集合値を最も一般化した状態である．各ノードの持つアイテム集合を，ある集合値を匿名化した状態とする．このとき，1 つのアイテムを削除して上位概念のノードへと再符号化することで，集合値は一般化される．また，1 つのアイテムを追加 (開示) して下位概念のノードへと再符号化することで，集合値は詳細化される．このように集合値を成すアイテムの開示/削除によって，集合値の再符号化が実現できる．

また，集合値に対して共通のアイテム集合を複数のレコードが持つとき，共通のアイテム集合からは個々のレコードを識別することができない．よって，共通のアイテム集合だけを開示し，それ以外のアイテム集合を削除する再符号化を行うことで，ある一定の匿名性が保証できる．この操作を削除ベースの再符号化と呼ぶ．以降，特筆しない場合，集合値属性に対する再符号化は，削除ベースの再符号化とする．さらに，k レコードが共通のアイテム集合だけから成るとき，当該レコード群は k-匿名性を満たすと言える．再符号化によって削除されるアイテム数が少ないほど，元のデータをよく表していると言える．しかしながら，削除アイテム数が最小となる最適なレコードの組合せの探索は NP-困難である．

そこで，他の属性と共存した効率良く匿名化を実現するために，段階的なトップダウンアプローチを採用することを前提としたヒューリスティック手法について考える．

まず，集合値を最も一般化された状態にし，集合値に関する知識からはレコードを区別できない状態へと再符号化する．集合値における最も一般化された状態への再符号化は，すべてのアイテムの削除，またはすべてのアイテムを包含する概念への一般化である．本稿では，一般化階層の存在を前提としないため，すべてのアイテムの削除を用いる．これによって，全てのレコードが同じ集合値を共通に持ち，k-匿名性を満たすクラスタ (レコードの集合) が生成される．

トップダウンアプローチに適した詳細化を実現するために，すべてのアイテムを削除した状態から，アイテムを徐々に開示していくことで，詳細化を行う (図 5)．情報損失を低減するという観点から，最終的な匿名化結果ではできるだけ多くのアイテム集合が開示されることが望ましい．また，他の属性の詳細化によって k-匿名化が終了することを考えると，類似性の高いレコードがただ集められるだけでなく，確実に多くのアイテムが開示されることも望まれる．

3.1 詳細化による情報利得と分割点決定

クラスタ c に対してアイテム α を基準に，α を含む/含まないことによってレコードを 2 つのクラスタへと分割すると，この分割によって，α を含むレコード数だけ，情報損失を低減できる．この 1 度の 2-分割による情報損失の低減度合いを情報利得 δ とする．情報利得 δ は以下のように表すことができる．

$$\delta = f(\alpha) \tag{1}$$

$f(\alpha)$ はクラスタ c のレコード群における α を含むレコード数 (出現頻度) である．本稿では，上述のようにアイテムの出現頻度に基づいてクラスタ分割の基準となるアイテム α_{piv} を選択し，α_{piv} を含むクラスタと含まないクラスタへの 2 分割によって集合値を詳細化することを考える．分割の基準としたアイテム α_{piv} を分割点と呼ぶ．

出現頻度が高いアイテムは多くのレコードを詳細化可能であり，情報利得が高い．ただし，出現頻度が最大のアイテムの情報利得は必ずしも最大ではない．

クラスタ c のメンバであるすべてのレコードに共通のアイテム集合を β とする．クラスタ c をそれ以上詳細化せずに匿名化結果として出力した場合，少なくとも β をクラスタ c に属するレコードの匿名化結果として開示できる．よって，クラスタ c の時点における開示可能アイテム数は，以下の通りである．

$$pub(c) = |\beta||c| \tag{2}$$

上述のように α_{piv} を基準に 2-分割した場合，開示可能アイテムの総数は以下の通りである．

$$\delta(c, \alpha_{piv}) = pub(c) + f(\alpha) = |\beta||c| + f(\alpha_{piv}) \tag{3}$$

よって，出現頻度の高いアイテムを分割点 α_{piv} として選ぶことで，多くのアイテムを開示できる．しかしながら，詳細化によって，k-匿名性に違反する場合にはこの限りではない．$f(\alpha_{piv})$ が $|c| - k$ よりも大きいとき，α_{piv} を含まないレコード群から

作られるクラスタ $c^{\#}$ は，属するレコード数が k 未満となるため k-匿名性を満たさない．よって，クラスタ $c^{\#}$ のレコードはすべて削除する．そのため，$f(\alpha_{piv}) \geqq |c| - k$ のとき，開示可能アイテムの総数は以下のようになる．

$$\delta(c, \alpha_{piv}) = (|\beta| + 1) f(\alpha_{piv}) \tag{4}$$

以上より，詳細化後の開示可能アイテムの総数は以下のようになる．

$$\delta(c, \alpha_{piv}) = \begin{cases} |\beta||c| + f(\alpha_{piv}) & (f(\alpha_{piv}) < |c| - k) \\ (|\beta| + 1) f(\alpha_{piv}) & (otherwise) \end{cases} \tag{5}$$

出現頻度が最大のアイテムを α_{max}，$f(\alpha_{max}) < |c| - k$ を満たすアイテムの中で出現頻度が最大のアイテムを α'_{max} とする．このとき，開示アイテム数を最大化する α_{piv} は α_{max} と α'_{max} から選べばよいことが分かる。

例として，アイテム集合 $\beta = \{x, y\}$ を共通に持つクラスタ c について考える．ここで k=3，クラスタ c のレコード数は $|c| = 10$ とする．クラスタ c のメンバであるレコードに含まれるアイテム a は出現頻度が 8 と最も高く，アイテム b は出現頻度が 6 とする．$alpha_{piv}$=a として詳細化すると，$\{x, y, a\}$ を共通に含むクラスタ c' と，$\{x, y\}$ を共通に含み，a を含まないクラスタ $c^{\#}$ に分割される．しかしながら，クラスタ $c^{\#}$ のレコード数は 2 であり，k=3 の k-匿名性を充足しないため削除の対象となる．一方，アイテム b で分割した場合は，分割後の 2 つのクラスタは共に k=3 の k-匿名性を充足する．よって，$\delta(c, a) = 24$，$\delta(c, b) = 26$ より，開示可能アイテム数を最大化するという観点からは，b が a よりも良い分割点である．

さらに，本稿では削除可能なレコード数に上限 θ_{sup} を与え，その上限を超えない範囲で，削除を許容した詳細化を行う．言い換えると，削除されたレコード数が θ_{sup} を超えない間，α_{max} と α'_{max} から分割点を決定し，θ_{sup} を超過した場合は常に α'_{max} を分割点とする．

3.2 削除/開示ベースの詳細化

上述のように，抽出した分割点を用いて 1 つのクラスタを 2 つのクラスタに分割し，集合値を段階的に詳細化する．クラスタ分割の結果，レコード数が k 未満のクラスタは削除し，このクラスタに含まれるレコードは匿名化済みデータに記録しない．このレコードの削除は，削除許容割合を超えるまで可能とする．また，削除許容割合はあらかじめ設定しておくものとする．

図 6 は，属性：薬剤名に対する詳細化結果をイテレーション毎に示している．以降では，図 6 を用いて具体例を示しながら説明する．イテレーション 1 では，アイテム a が分割点として選択された．レセ ID1〜4 は a を含むクラスタへ，レセ ID5〜9 はもう一方のクラスタへと分割される．分割の結果，アイテム a を全てのレコードが共通に持つクラスタと，全てのレコードに共通の属性値を持たないクラスタとに分割される．この 2 つのクラスタは，それぞれサイズ 4，5 であり，k-匿名性 (ここでは k=2) を満たす．イテレーション 2 では，イテレーション 1 で作られた a を含むクラスタをアイテム d を分割点として分割する．イテレーション 3，4 でも同様に分割することで詳細化を行う．分割処理の収束後，各クラスタの共通アイテムを匿名化済み結果として出力する．

		再符号化のイテレーション			
薬剤名	初期状態	1回目	2回目	3回目	4回目
a, b, d	*	a	a, d	a, d	a, d
a, f, g	*	a	a	a, f	a, f, g
a, d, f, y, z	*	a	a, d	a, d	a, d
a, b, f, g	*	a	a	a, f	a, f, g
b, c, f	*	*	*	b	b, c
c, e, x	*	*	e	e, x	e, x
e, x	*	*	e	e, x	e, x
b, c	*	*	*	b	b, c
c, e, x	*	*	e	e, x	e, x

図 6　クラスタ分割例

4. 複合データの k-匿名化

既存の研究では独立して提案されてきた単一値の匿名化と，集合値の匿名化とを統一的に処理する手法を提案する．特に，単一値と集合値の双方でトップダウンアプローチを採用し，共存させた匿名化手法を提案する．

単一値と集合値の情報損失の評価を，統一的な指標を用いて行い，詳細化の対象を公平に決定する．その上で，各属性の詳細化を段階的に行うことで，属性ごとに異なる詳細化手法を共存させる．

単一値に対する詳細化は，既存の任意のトップダウンアプローチによる再符号化手法を用いてよい．ただし，すべての属性で統一的な有効性指標に基づく必要がある．この指標に基づいて，どの属性によって詳細化を行うべきかを決定する．

4.1 情報損失指標

本稿では，有効性指標として，情報損失の大きさを採用し，情報損失が最大の属性を詳細化の対象とする．詳細化対象を適切に決定するためには，全ての属性の情報損失の大きさを統一的な指標で扱う必要がある．

NCP [3] は単一値の一般化度合いを表す指標であり，0〜1 の値を取る．0 のとき，属性値は全く一般化されていないことを表す．1 のとき，最大まで一般化されたことを表す．

属性ごとの NCP 値は，レコード毎の NCP 値の総和で表わされ，0〜$|T|$ の範囲の値をとる．ここで，$|T|$ はテーブル T 中のレコード数を表す．

レコード t の数値属性 A の NCP は以下の式で表される．

$$NCP_A(t) = \frac{|a_{max} - a_{min}|}{|A|} \tag{6}$$

ここで，a はレコード t の属性 A の値であり，a_{max} は a の最大値，a_{min} は a の最小値であり，$|A|$ は属性 A のレンジを表す．

一般化階層を持つカテゴリカル属性 A に関するレコード t の NCP は以下の式で表される．

表 1　属性：傷病名のアイテム数

データセット	単一値 QI 数	集合値 QI 数	レコード数	総アイテム数	ユニークアイテム数	平均アイテム数
レセプト 10K	3	1	10,000	52,021	169	5.202
レセプト 100K	3	1	100,000	441,940	4,156	4.442
BMS-WebView-2	0	1	77,512	358,278	3,340	4.622

$$NCP_A(t) = \frac{descendants(a)}{|A|} \tag{7}$$

ここで，$descendants(a)$ は属性値 (ノード)a の子孫であるリーフノード数を表す．また，$|A|$ は属性 A の全リーフノード数を表す．

4.2　集合値属性の情報損失

一般化を用いた集合値属性の匿名化をするときの集合値属性値に対する NCP は He らによって提案されている [6]．本節では，集合値属性の NCP [6] を削除のみを用いた匿名化に適用した NCP_{sv} を定義する．まず，あるレコード中のアイテム α の NCP を以下の式で定義する．

$$NCP_{sv}(\alpha) = \begin{cases} 1 & \alpha \ is \ suppressed \\ 0 & otherwise \end{cases} \tag{8}$$

$NCP_{sv}(\alpha)$ を用いてレコード t の NCP_{sv}，テーブル T の NCP_{sv} は以下のように定義する．

$$NCP_{sv}(t[A]) = \frac{\sum_{\alpha \in t} NCP_{sv}(\alpha)}{|t[A]|} \tag{9}$$

$$NCP_{sv}(T[A]) = \sum_{t \in T[A]} NCP_{sv}(t[A]) \tag{10}$$

ここで，$|t|$ は t に含まれるアイテム α の数を表す．式 (9)，(10) は，レコード毎のオリジナルのアイテム数 $|t|$ および削除されたアイテム数を必要とする．したがって，レコード数が $|T|$ のとき，$|t|$ の計算コストもしくは計算結果を記憶しておく空間コスト $O(|T|)$ が必要となる．さらに，u 個の集合値属性がある場合，$O(u|T|)$ の空間が必要となる．

4.3　効率の良い集合値属性の情報損失計算

そこで，効率良く集合値属性の情報損失を計算可能な手法を導出する．$r(t[A])$ を $t[A]$ のアイテムのうち，開示されている (削除されていない) アイテムの集合とする．

$$NCP_{sv}(T[A]) = \sum_{t \in T[A]} \frac{|t[A]| - |r(t[A])|}{|t[A]|} \tag{11}$$

$$NCP_{sv}(T[A]) = \sum_{t \in T[A]} \left(1 - \frac{|r(t[A])|}{|t[A]|}\right) \tag{12}$$

$$NCP_{sv}(T[A]) = |T| - \sum_{t \in T[A]} \frac{|r(t[A])|}{|t[A]|} \tag{13}$$

ここで，τ_A をレコードの持つアイテム数の平均値とする ($\tau_A = \frac{|t[A]|}{|T|}$)．$|t[A]|$ を τ_A で置換して近似すると，

$$NCP_{sv}(T[A]) \simeq |T| - \sum_{t \in T[A]} \frac{|r(t[A])|}{\tau_A} \tag{14}$$

以上を整理すると，集合値属性の NCP は以下のようになる．

$$NCP^*_{sv}(T[A]) = |T| - \tau_A^{-1} N_r(A) \tag{15}$$

式 (15) では，$|T|$ と τ_A^{-1} は定数項であり，あらかじめ計算しておくことができる．よって，開示されているアイテム数 $N_r(A)$ の関数となり，開示アイテム数の計数だけで導出できるため，高速に計算可能である．

5. 評価実験

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，集合値属性において削除されたアイテム数，情報損失量を評価し，匿名化の実行時間を計測した．

5.1　評価環境

本評価で用いたデータセットの統計情報を表 1 に示す．

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(注1)が提供するレセプトデータを用いた．約 10 万人の患者を含む約 400 万件のレセプトデータからランダムに抽出した 1 万件，10 万件のレセプトを対象とした．匿名化対象のテーブルとして，(レセプト ID, 患者 ID, 生年，性別，診療年月, 傷病名) を含むテーブルを用いた．準識別子は，生年，性別，診療年月，傷病名とした．

また，KDD CUP 2000 で提供されている相関ルールマイニング用データセットの 1 つである BMS-WebView-2 も匿名化の対象データとして用いた．BMS-WebView-2 はトランザクションデータであるため，準識別子は集合値属性が 1 つのみである．

5.2　削除アイテム数

情報の損失の大きさを評価するために削除されたアイテム数を算出した．情報損失の大きさの指標である NCP は，式 (15) の通り，削除アイテム数と正の相関がある．1 万件のレセプトデータに対して，k=2, 5, 10, 20, 50 で匿名化した際の，属性：傷病名が削除されたアイテムの割合を図 7 に示す．1 万件のレセプトデータに対して，k=10 で準識別子の種類を変更して匿名化した際の，属性：傷病名が削除されたアイテムの割合を図 8 に示す．BMS-WebView-2 に対して，k=2, 5, 10, 20, 50 で匿名化した際の，属性：傷病名が削除されたアイテムの割合を図 9 に示す．

図 7 からは，k の値が大きいほど，k-匿名性の充足に多くのアイテムの削除を必要としていることがわかる．これは，k が大きいほど，同一のアイテム集合の組合せを持たせることが困難になるためと考えられる．

図 8 は，準識別子の数を変更した場合の削除アイテム数の変

(注1)：http://www.jmdc.co.jp/

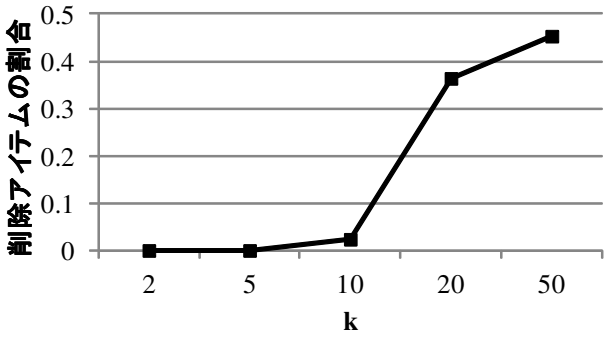

図 7　削除アイテムの割合 (レセプト 10K)

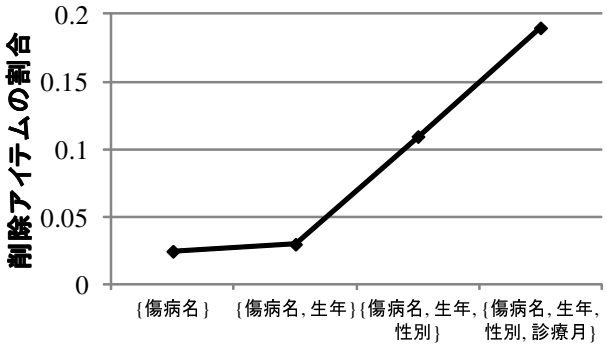

図 8　削除アイテムの割合 v.s. QI 数 (レセプト 10K)

化を示している．図 8 の評価では，単一値属性を増やした場合について評価した．準識別子の種類数が増加するに従って，削除されるアイテム数が増えている．準識別子の種類が増加すると，すべての準識別子の値の組が k-匿名性を満たす必要があるため，k-匿名性の充足が難しくなり，多くのアイテムが削除されやすいと考えられる．

図 9 は，BMS-WebView-2 に対して提案手法による匿名化を実施した結果を示している．図 9 では，50% 以上のアイテムが削除されている．また，レセプトデータと同様に k が大きくなると，多くのアイテムが削除されている．BMS-WebView-2 はレセプトデータに比べて，アイテムの組合せの種類が多いため，k-匿名性の充足に，多くのアイテムの削除が必要だったのではないかと考えられる．

次に 1 万件と 10 万件のレセプトデータに対して，k=10 で匿名化した際の属性：傷病名の削除されたアイテム数と割合を表 2 に示す．10 万件のレセプトデータでは 13.3% の削除で k-匿名化を実現できた．一方，1 万件のレセプトデータでは約半数のアイテムの削除が行われた．このことから，アイテムの出現頻度分布にも依存するが，データセットのサイズが大きいほど容易に k-匿名性を充足できることが分かる．

提案手法は，削除のみを用いるため，所望の匿名性が高い (k の値が大きい) 場合や，対象データセットが小さい場合に，非常に多くのアイテムの削除を必要とすることが分かる．ただし，タキソノミー等を持たず，一般化階層の定義が難しい集合値属性に対しても k-匿名化を実現でき，幅広い集合値属性を匿名化できることは利点の 1 つである．

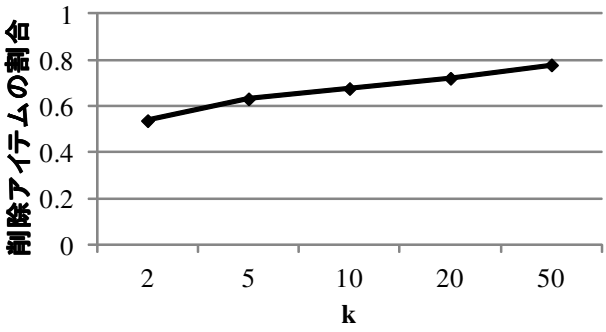

図 9　削除アイテムの割合 (BMS-WebView-2)

表 2　削除アイテム数の比較 (データサイズ)

レコード数	削除アイテム数	削除アイテムの割合
10,000	26383	50.7%
100,000	58847	13.3%

5.3　匿名化処理時間

次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．評価には，32GB の主記憶と，2.4GHz の CPU(2 コア) の仮想マシンを利用した．仮想マシンのホストは，192GB の主記憶と 2.4GHz の CPU(12 コア) を持つ計算機である．Java 1.6.0_24，PostgreSQL 9.1.8 を利用して提案手法を実装した．

まず，k-匿名性の k の値を変化させたときの，処理時間の変化を評価する．1 万件のレセプトデータに対して，k=2, 5, 10, 20, 50 で匿名化した際の，匿名化に要した処理時間を図 10 に示す．また，BMS-WebView-2 に対して同様に処理時間を計測した結果を図 11 に示す．

図 10，11 より，k の値が増加すると，処理時間が短くなることがわかる．これは，k の値が大きくなるほど，k-匿名性の充足が難しく，属性が詳細化不可能となりやすいためではないかと考えられる．

次に，スケーラビリティを評価するために，1 万件と 10 万件のサイズの異なるデータセットの匿名化に要した処理時間を計測した．k=10 の場合の処理時間を比較する．比較した結果を表 3 に示す．

10 万件のデータセットを匿名化した場合には，1 万件の場合の約 70 倍の処理時間を要している．10 万件のデータセットは，アイテムの種類数が 1 万件のデータセットの約 25 倍であり，考慮すべきアイテムの種類数や組合せが多いため，匿名化にもより多くの時間が掛かってしまったのではないかと考えられる．特に，本手法では，アイテムの種類数を M, 平均アイテム数を μ とすると，${}_M C_\mu + \ldots {}_M C_1$ 回の程度の詳細化が必要となる．1 回の詳細化の計算コストは大きくないが，アイテムの種類数の増加や，集合値の長さの伸長によって詳細化の回数が膨大になり計算コストが増大すると考えられる．

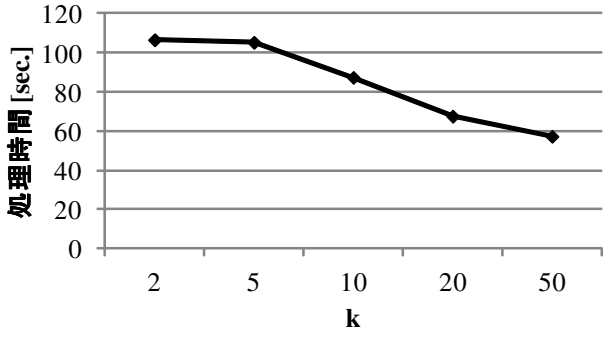

図 10　匿名化処理時間 (レセプト 10K)

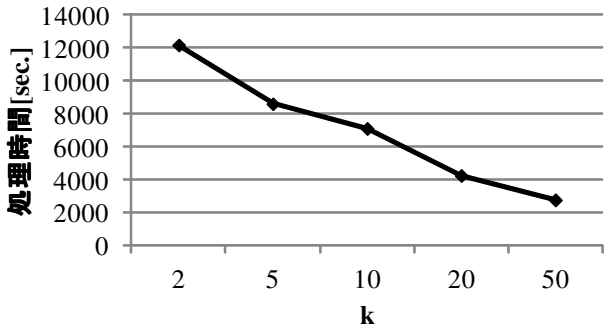

図 11　匿名化処理時間 (BMS-WebView-2)

表 3　匿名化処理時間の比較

レコード数	匿名化処理時間
10,000	87.4 sec.
100,000	7005.4 sec.

6. ま と め

本稿では，単一値属性と集合値属性が混在するデータセットを匿名化する問題に取り組んだ．まず，一般化階層のない集合値属性をトップダウンアプローチで匿名化する手法を提案した．この手法を用いて，さらに，単一値属性と集合値属性を統一的に k-匿名化する手法を提案した．その際，集合値属性の情報損失の度合いを高速に計算する手法についても示した．評価実験では，レセプトデータを用い，削除されたアイテム数や匿名化に必要な処理時間を示し，それらの傾向が確認された．

今後は，複数のデータセットを用いてさらに詳細な評価を行う予定である．また，既存の集合値匿名化手法や，トップダウンアプローチとの比較も行う予定である．加えて，本稿では単純な 2-分割のみで集合値の詳細化を実現したが，複数分割や並列化によって高速化が期待できると考えており，今後の課題の 1 つである．

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] 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).

[3] 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).

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

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

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

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

[8] Liu, J. and Wang, K.: Anonymizing Transaction Data by Integrating Suppression and Generalization. Proc. $PAKDD$2010 (2010).

[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] KDD CUP. `http://www.kdd.org/kddcup/index.php`