

分類分析のための k -匿名化テーブル評価関数の提案

橋本 翔太[†] 上土井陽子^{††} 若林 真一^{††}

[†] 広島市立大学情報科学部 〒731-3194 広島県広島市安佐南区大塚東三丁目4番1号

^{††} 広島市立大学大学院情報科学研究科 〒731-3194 広島県広島市安佐南区大塚東三丁目4番1号

E-mail: [†]shotah@lcs.info.hiroshima-cu.ac.jp, ^{††}{yoko,wakaba}@hiroshima-cu.ac.jp

あらまし 本研究では分類解析に適した k -匿名化テーブルを作成することを目的として k -匿名化テーブルを評価する関数を定義する．分類分析への入力となる k -匿名化テーブルを作成することを目的とした先行研究は存在するが，作成された表の評価方法は明確になっていなかった．我々は先行研究で導入されたエントロピーを利用した特殊化評価尺度を，テーブル全体を評価する関数に拡張する．

キーワード 階層的 k -匿名化，特殊化，分類

1. はじめに

近年，情報技術の発展により，企業は顧客の機密データを含む大量の情報を所有し，他の企業とも共有できる環境になってきている．企業は，蓄積した大量の情報を顧客に対するサービス等を考える際に有用に利用したい．しかし，大量の情報をそのまま利用しても，良質なサービスを提供することにつながる可能性はある．一方，顧客データを利用するにはプライバシー保護できるようデータを変換しなければならない．従来研究[1]では，第三者へ公開することを想定し，分類ルールを作る際にできるだけ有用な情報へ加工する匿名化手法が提案されている．しかし，作成された情報に対する評価方法は明示されていなかった．本稿では，データマイニングの手法の1つである分類分析 (Classification: 図1) に適した匿名化テーブル変換問題を新たに定義し，階層的匿名化フレームワークで利用する評価関数を導出する．

2. 準備

本稿で扱うデータテーブル T は， ID を識別子の集合である識別子属性， $D_1, \dots, D_p$ を値の集合である準識別子属性， $Class$ をタイプの集合である型属性としたとき，テーブル $T(ID, D_1, \dots, D_p, Class)$ を各属性に属する値の $p+2$ 個の列 $(id, d_1, \dots, d_p, cls)(id \in ID, d_i \in D_i(1 \leq i \leq p), cls \in Class)$ の集合とする．また，テーブル T に属する $p+2$ 個の値の列をレコードと呼ぶ．また，各属性に含まれる値をドメイン値と呼ぶ．テーブルの例を表1に示す．表1では34個のレコードを準識別子属性の値の組合せが同じものを集約し，1つの行で表現している．例えば，表1の3行目は， $\langle Male, Carpenter, 35 \rangle$ である5つのレコードをもち，そのうち型属性 $Class$ が値 Y であるレコードが2つあり，型属性 $Class$ が値 N であるレコードが3つあることを表す．本稿では，表を表現する場合，表1のような集約した形式を用いる．

テーブル $T(ID, D_1, \dots, D_p, Class)$ の属性の列の部分列 $A_1, A_2, \dots, A_l$ の属性が与えられたとき， $T(A_1, A_2, \dots, A_l)$ によって，テーブル $T(ID, D_1, \dots, D_p, Class)$ の属性列 $A_1, A_2, \dots, A_l$

表1 テーブル T

ID	Class	Sex	Job	Salary(K)
1-3	0Y3N	Male	Janitor	30
4-7	0Y4N	Male	Mover	32
8-12	2Y3N	Male	Carpenter	35
13-16	3Y1N	Female	Technician	37
17-22	4Y2N	Female	Manager	42
23-25	3Y0N	Female	Manager	44
26-28	3Y0N	Male	Accountant	44
29-31	3Y0N	Female	Accountant	44
32-33	2Y0N	Male	Lawyer	44
34	1Y0N	Female	Lawyer	44

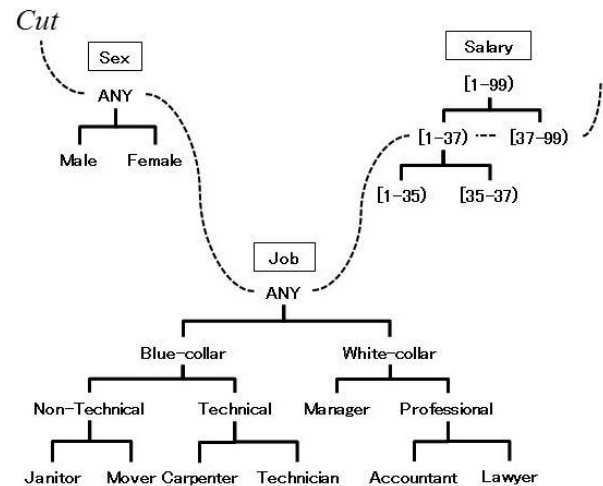


図2 分類木と森のカット

に関する射影を表すとする．

2.1 k -匿名性

テーブル $T(ID, D_1, \dots, D_p, Class)$ において，準識別子属性の集合を $QID = \{D_1, \dots, D_p\}$ としたとき， QID の並び $D_1, \dots, D_p$ において， p 個のドメイン値の各列 $(d_1, d_2, \dots, d_p)(d_i \in D_i)$ に対して，テーブル $T(ID, D_1, \dots, D_p, Class)$ に属するレコードで，準識別子属

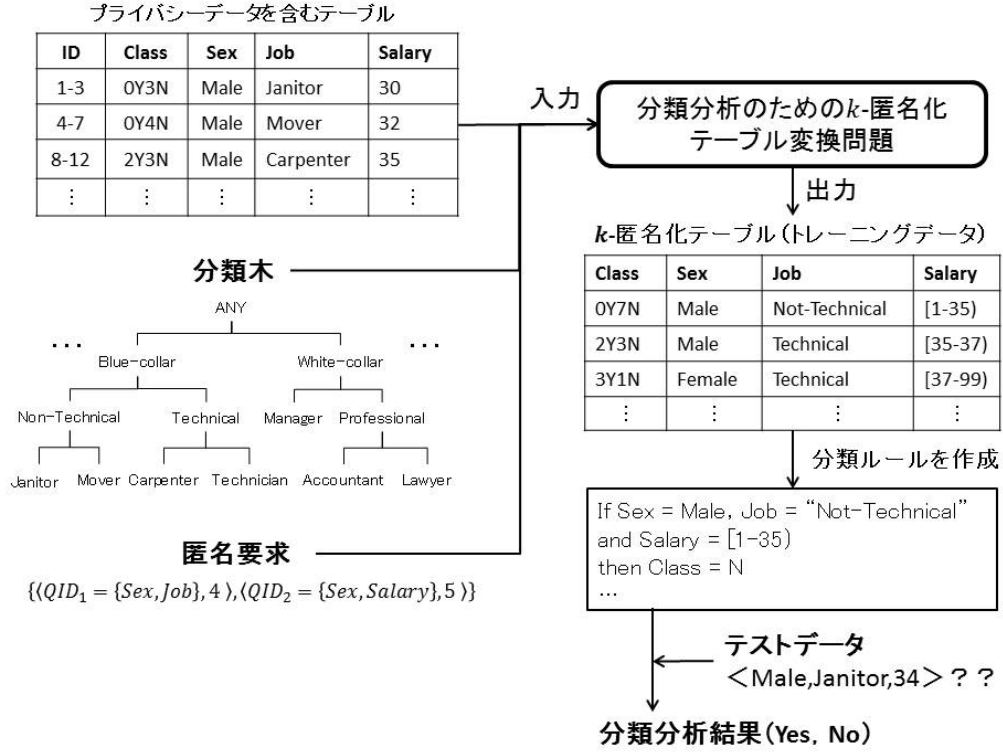


図 1 本研究で対象とする分類分析概要

性の列に対応する値の部分列が $(d_1, d_2, \dots, d_p)$ と一致するレコードの集合を $S_{(d_1, d_2, \dots, d_p)}$ とする．準識別子集合 QID の列 $D_1, \dots, D_p$ において，定義される全ての p 個のドメイン値の列 $(d_1, d_2, \dots, d_p)$ の中で， $S_{(d_1, d_2, \dots, d_p)}$ が空集合ではなく，サイズが最も小さいという条件を満たす列を $(d'_1, d'_2, \dots, d'_p)$ とする．このとき， $|S_{(d'_1, d'_2, \dots, d'_p)}|$ を準識別子集合 QID の匿名度 $A(QID)$ とする．また， k をユーザ定義の匿名性の閾値とすると， $A(QID) \geq k$ ならば，匿名要求 $\langle QID, k \rangle$ を満たしているとする．

さらに，これを準識別子属性集合 QID の部分集合に対応するよう拡張し，部分集合の列 $\{QID_1, QID_2, \dots, QID_j\}$ におけるテーブル $T(ID, D_1, \dots, D_p, Class)$ での各 QID_i の匿名性を $A(QID_1), A(QID_2), \dots, A(QID_j)$ とする．そのとき， $A(QID_1) \geq k_1$ かつ， $\dots$ ，かつ $A(QID_j) \geq k_j$ ならば，匿名要求 $\{\langle QID_1, k_1 \rangle, \dots, \langle QID_j, k_j \rangle\}$ を満たしているという．

2.2 分類木とカット

準識別子属性集合 $QID = \{D_1, D_2, \dots, D_p\}$ の各属性 $D_i (1 \leq i \leq p)$ に対して，分類木 $Tree_i$ を次のように定義する．図 2 に分類木の例を示す．

(1) 属性 D_i が分類別的 (categorical) な属性の場合：葉ノードはドメイン値，親ノードは子ノードよりも特定の度合いの低い値を表す木構造で，実行前に決定されている．

(2) 属性 D_i が連続的 (numerical) な属性の場合：各ノードは間隔を表し，葉でない各ノードは型属性 $Class$ における情報利得 (InfoGain [1]) を最大にする親の間隔のある最適な 2 分割を表す 2 つの子ノードをもち，葉ノードはテーブルに出現

しているドメイン値である木構造で，実行時に作成される．

また，分類木 $Tree_i$ における根から葉までの各パス上のノードを正確に 1 つ含むノードの集合をカットといい，属性 D_i の分類木 $Tree_i$ におけるカットを cut_i で表す．また，各分類木のカットを含む集合を分類木の森のカットと呼ぶ (図 2)．

2.3 一般化

テーブル $T(ID, D_1, \dots, D_p, Class)$ のあるレコード $(id, v_1, \dots, v_p, cls) \in T(ID, D_1, \dots, D_p, Class)$ を分類木の森のカット $Cut = \{cut_1, \dots, cut_p\}$ を使って一般化し，作成されたレコードを $(id, v'_1, \dots, v'_p, cls)$ とする．テーブル $T(ID, D_1, \dots, D_p, Class)$ の全てのレコードを一般化することで，一般化テーブルを作成する．以下では，値の一般化についてまず定義し，レコード，テーブルの一般化の定義に拡張する．

2.3.1 分類木 cut_i による値 v_i の一般化

準識別子属性 D_i の cut_i に含まれる値 v'_i で値 v_i を置き換えることを $general(cut_i, v_i) = v'_i$ と表す．ここで， v'_i は i 番目の準識別子属性 D_i の分類木 $Tree_i$ の値 v_i から根ノードまでのパス上のノード集合を $path_vertex(i, v_i)$ とするとき， $|path_vertex(i, v_i) \cap cut_i| = 1$ となる分類木 $Tree_i$ のカット cut_i において $v'_i \in (path_vertex(i, v_i) \cap cut_i)$ を満たす．値 v_i と値 v'_i は図 4 に示されている関係にある．

2.3.2 分類木の森 Cut によるレコードの一般化

レコード $(id, v_1, \dots, v_p, cls)$ ，分類木の森のカット $Cut = \{cut_1, \dots, cut_p\}$ が与えられたとき， Cut によって一般化して作成された一般化レコード $(id, v'_1, \dots, v'_p, cls)$ を次の関数により

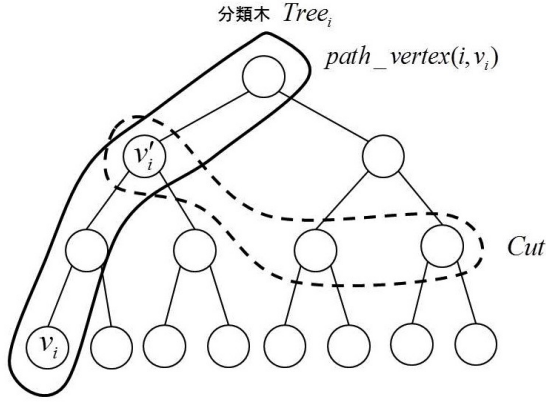


図 3 分類木 $Tree_i$ における値 v_i と値 v'_i の関係

定義する．

$$\begin{aligned} General(Cut, (id, v_1, \dots, v_p, cls)) &= (id, v'_1, \dots, v'_p, cls) \\ | v'_i &= general(cut_i, v_i) (1 \leq i \leq p) \quad (1) \end{aligned}$$

2.3.3 分類木の森 Cut によるテーブルの一般化

テーブル $T(ID, D_1, \dots, D_p, Class)$ ，分類木の森のカット $Cut = \{cut_1, \dots, cut_p\}$ が与えられたとき，テーブル $T(ID, D_1, \dots, D_p, Class)$ を一般化する関数 $GeneralTable$ を以下の式を満たす関数と定義する．またこの関数を適用し，得られたテーブルを匿名化テーブルという．以降の記述では，表 T の匿名化テーブルを T_g と表記する．

$$\begin{aligned} GeneralTable(Cut, T) &= \{(id, v'_1, v'_2, \dots, v'_p, cls) \\ | (id, v_1, \dots, v_p, cls) &\in T \wedge (id, v'_1, v'_2, \dots, v'_p, cls) \\ &= General(Cut, (id, v_1, \dots, v_p, cls))\} \quad (2) \end{aligned}$$

2.4 特殊化

元のテーブル $T(ID, D_1, \dots, D_p, Class)$ と準識別子属性 $D_1, D_2, \dots, D_p$ の分類木 $Tree_1, Tree_2, \dots, Tree_p$ と，それら分類木によって構成される森に対するソリューションカット Cut と Cut に対応する匿名化テーブル $T_g(Cut)$ が与えられているとき， Cut に含まれているノード v をその子ノードの集合 $child(v)$ に置き換え， Cut を Cut' に更新することを特殊化と呼ぶ．つまり， $Cut' = (Cut - \{v\}) \cup child(v)$ である．このとき， Cut' に対応する匿名化テーブル $T_g(Cut')$ ， Cut に対応する匿名化テーブル $T_g(Cut)$ を特殊化したテーブルと呼ぶ．また， Cut から Cut' への特殊化を $v \rightarrow child(v)$ と表す．

もし特殊化が特殊化のあとに匿名要求を満たすテーブルを結果として出力するならば，valid(妥当)である．特殊化は， v に一般化されていたレコード全てが同じクラスをもっていないなら，beneficial(有益)である．もしそうでないならば，そのとき特殊化は分類分析にとって役に立つ情報を導かない．よって特殊化は，妥当かつ有益である場合に限り，実行する．

3. 従来研究での匿名化テーブル変換問題

従来研究で対象とする分類分析のための匿名化テーブル変換

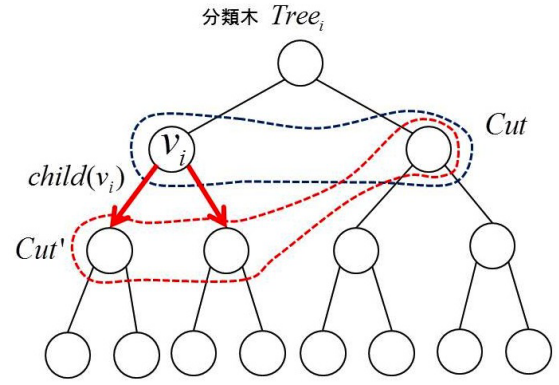


図 4 特殊化の例

問題は，テーブル T ，匿名要求 $\{\langle QID_1, k_1 \rangle, \dots, \langle QID_l, k_l \rangle\}$ ，準識別子属性集合 QID のそれぞれの属性に対する分類木を与えたとき，以下の制約条件を満たす匿名化テーブル T_g を作ることである．

- (1) 匿名化テーブル T_g は，匿名要求を満たしている．
- (2) 匿名化テーブル T_g は，分類分析のためにできるだけ多くの情報を含む．

3.1 階層的トップダウンフレームワーク

従来研究では，分類分析のための匿名化テーブル変換問題を解くためのフレームワークとして，Fung らが TDS(Top-Down Specialization) フレームワーク [3] を提案している．この TDS フレームワークのフローチャートを図 5 に示す．なお，図中の特殊化のよさを表す関数 $Score$ の引数は省略しているが，型属性 $Class$ に関してなるべくまとまったグループに分割できる特殊化を評価するための関数である．このフレームワークは，最初にテーブル T_g を分類木で各属性の根となる値で一般化してから，順々に特殊化を行い，匿名化テーブルを作成する．

3.2 特殊化の指標 $Score(v)$

特殊化 $v \rightarrow child(v)$ の前後における利得比を表す $Score(v)$ がある． $Score(v)$ は，値 v の上で特殊化の”よさ”を計ることができる． $Score(v)$ の計算式は式 (3) であり， $Score(v)$ を計算する際に必要とする $InfoGain(v)$ と $SplitInfo(v)$ の計算式について，順に示していく．

$$Score(v) = GainRatio(v) = \frac{InfoGain(v)}{SplitInfo(v)} \quad (3)$$

3.2.1 特殊化 $v \rightarrow child(v)$ における情報利得 $InfoGain(v)$

値 x に一般化された T のレコードの集合を $T[x]$ とする．各レコードは分類クラス集合 CLS に属するクラス cls をもっているとする．クラス cls をもっている $T[x]$ のレコードの個数を $freq(T[x], cls)$ とする． $c \in child(v)$ であるとき， $|T[v]| = \sum_{c \in child(v)} |T[c]|$ である． $I(T[x])$ が $T[x]$ のエントロピーであるとき，

$$I(T[x]) = - \sum_{cls \in CLS} \frac{freq(T[x], cls)}{|T[x]|} \times \log_2 \frac{freq(T[x], cls)}{|T[x]|} . \quad (4)$$

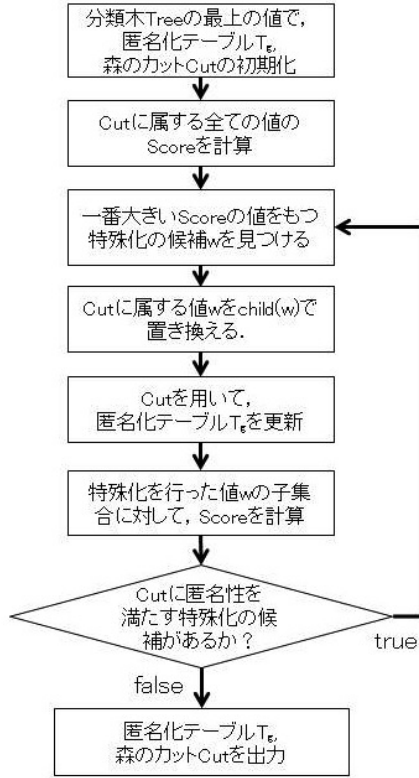


図 5 TDS フレームワーク

であるので,

$$InfoGain(v) = I(T[v]) - \sum_{c \in child(v)} \frac{|T[c]|}{|T[v]|} I(T[c]). \quad (5)$$

3.2.2 特殊化 $v \rightarrow child(v)$ における分割情報量 $SplitInfo(v)$

$InfoGain(v)$ の値は, 多数の子の値をもつ属性の特殊化では偏りができ, 他の属性よりも高い値をもつ. しかし, 多数の子をもつ属性を特殊化することは, 分類分析にとって有益ではないかもしれない. これは, 分割情報 $SplitInfo(v)$ で $InfoGain(v)$ を割ることにより避けうる.

$$SplitInfo(v) = - \sum_{c \in child(v)} \frac{|T[c]|}{|T[v]|} \times \log_2 \frac{|T[c]|}{|T[v]|}. \quad (6)$$

4. 新しい分類分析のための匿名化テーブル変換問題

従来研究にはなかった一般化テーブル T_g の同じ型属性 $Class$ のラベルをもつレコードを集めてグループ化したいので, クラスに関する乱雑さを目的関数とし, この目的関数値が最小である一般化テーブルを作成する問題を以下に明確に定義する.

4.1 問題定義

テーブル T , 匿名要求 $\{\langle QID_1, k_1 \rangle, \dots, \langle QID_l, k_l \rangle\}$, 準識別子属性集合 QID のそれぞれの属性に対する分類木を与えたとき, 匿名化テーブル変換問題は, 以下の制約条件を満たしながら, 後に詳述する目的関数 $TableInfo(T, Cut)$ を最小にする匿名化テーブル T_g を作る問題である.

制約条件

- 匿名化テーブル T_g は, 匿名要求を満たしている.

目的関数

テーブル $T(ID, D_1, \dots, D_p, Class)$ の型属性 $Class$ に関する乱雑さを計算する関数 $TableInfo(T, Cut)$ の計算式は式 (7) である. w はユーザが指定する重み係数である. $TableInfo(Cut, v)$ を計算する際に必要とする $ClassInfo(T, Cut)$ と $SplitInfo(T, Cut)$ の計算式について, 次節より順に示していく.

$$TableInfo(T, Cut) = w \times ClassInfo(T, Cut) + (1 - w) \times SplitInfo(T, Cut) \quad (7)$$

4.2 種々の関数

4.2.1 属性値によるテーブルのグループ化

テーブル T と分類木の森のカット $Cut = \{cut_1, \dots, cut_p\}$ が与えられたとき, p 個の準識別子属性 $D_1, D_2, \dots, D_p$ のカット $cut_1, cut_2, \dots, cut_p$ の積集合 $cut_1 \times cut_2 \times \dots \times cut_p$ に含まれている p 個の値の組 $(v'_1, v'_2, \dots, v'_p)$ において, テーブル T に含まれているレコード $(id, v_1, v_2, \dots, v_p, cls)$ の匿名化レコードでその準識別子属性の値の列が $(v'_1, v'_2, \dots, v'_p)$ と一致する匿名化レコードの集合をグループと呼ぶ. 以下にグループの定義を示す.

$$group((v'_1, v'_2, \dots, v'_p), T, Cut) = \{(id, v'_1, v'_2, \dots, v'_p, cls) \mid (id, v_1, v_2, \dots, v_p, cls) \in T \wedge (id, v'_1, v'_2, \dots, v'_p, cls) = General(Cut, (id, v_1, v_2, \dots, v_p, cls))\} \quad (8)$$

また, テーブル T の匿名化テーブルが n 種類の p 個の準識別子属性の一般化値の組合せ $(v'_1, v'_2, \dots, v'_p)$ の組合せ $(v'_{11}, v'_{21}, \dots, v'_{p1}), \dots, (v'_{1n}, v'_{2n}, \dots, v'_{pn})$ を持つとき, それぞれの p 個の値の組合せに関するグループを $group_1, \dots, group_n$ とし, このグループの集合を $Group(T, Cut) = \{group_1, group_2, \dots, group_n\} \subseteq cut_1 \times cut_2 \times \dots \times cut_p$ とすると, 集合 $Group(T, Cut)$ に属するグループの和集合の大きさ $|\cup_{g \in Group(T, Cut)} g|$ とテーブル T の大きさ $|T|$ は必ず等しい.

4.2.2 テーブル T のクラスに関する乱雑さ $ClassInfo(T, Cut)$

カット Cut で匿名化されたテーブル T のグループの集合を $Group(T, Cut)$ とするとき, $Group(T, Cut)$ に属する任意の $group$ において, クラス cls をもっている $group$ のレコードの個数を $freq(group, cls)$ とする. このとき, 各グループ $group$ のクラスに関する乱雑さ $E(group)$ を式 (9) で定義する.

$$E(group) = - \sum_{cls \in Class} \frac{freq(group, cls)}{|group|} \times \log_2 \frac{freq(group, cls)}{|group|} \quad (9)$$

また, テーブル T のクラスに関する乱雑さ $ClassInfo(T, Cut)$ は, 各グループのクラスに関する乱雑さ $E(group)$ に重み $\frac{|group|}{|T|}$ をかけて足し合わせたものであり, 以下に定義する.

$$\begin{aligned}
& \text{ClassInfo}(T, \text{Cut}) \\
&= \sum_{\text{group} \in \text{Group}(T, \text{Cut})} \frac{|\text{group}|}{|T|} \cdot E(\text{group}) \\
&= - \sum_{\substack{\text{cls} \in \text{Class} \\ \wedge \text{group} \\ \in \text{Group}(T, \text{Cut})}} \frac{\text{freq}(\text{group}, \text{cls})}{|T|} \cdot \log_2 \frac{\text{freq}(\text{group}, \text{cls})}{|\text{group}|}
\end{aligned} \tag{10}$$

4.2.3 テーブル T のレコードに関する乱雑さ $\text{SplitInfo}(T, \text{Cut})$

本小節では分類ルールを作成する上で、 unnecessary 特殊化が行われていないかを評価する関数としてレコードに関する乱雑さを定義する． unnecessary 特殊化とは、表 8 の 6 行目のレコード $\langle \text{Male}, \text{Professional}, [37-99] \rangle$ を例にすると、準識別子属性 Salary の値 Professional で特殊化を行うと、レコードが細かく分類されてしまうような特殊化といい、分類ルールを作るという目的に反するので unnecessary 特殊化という．

カット Cut で匿名化されたテーブル T のレコードの個数を $|T|$ 、テーブル T のグループの集合を $\text{Group}(T, \text{Cut})$ とする．このとき、テーブル T のレコードの乱雑さ $\text{SplitInfo}(T, \text{Cut})$ は式 (15) である．

$$\begin{aligned}
& \text{SplitInfo}(T, \text{Cut}) \\
&= - \sum_{\text{group} \in \text{Group}(T, \text{Cut})} \frac{|\text{group}|}{|T|} \times \log_2 \frac{|\text{group}|}{|T|}
\end{aligned} \tag{11}$$

4.3 匿名化テーブルの情報量比較

以上に定義した関数を表 1 から表 8 のテーブル $T_{\text{raw}}, T_{\text{init}}, T_1, T_2, T_3, T_4, T_5, T_{\text{final}}$ を例に計算してみる．テーブル T_{raw} は生のデータテーブル、テーブル T_{init} は最も一般化されている初期テーブル、 $T_i (1 \leq i \leq 5)$ は i 回目の特殊化後のテーブル、 T_{final} は最後に出力される匿名化テーブルである．

テーブル T のクラスに関する乱雑さ ClassInfo 、テーブル T のレコードに関する乱雑さ SplitInfo 、目的関数であるテーブルの乱雑さ TableInfo の計算結果を表 9 に示す．表 9 から、

- 特殊化が進むにつれて、関数 ClassInfo の値が減少、関数 SplitInfo の値が増加している．
 - 特殊化が進むにつれて、関数 TableInfo の値が減少する．
 - 生のデータテーブル T_{raw} と最後に出力される一般化テーブル T_{final} の乱雑さの差は、匿名要求による差である．
- といったことが分かる．

表 2 テーブル T_{init}

Class	Sex	Job	Salary(K)
21Y13N	ANY	ANY	[1-99]

表 3 テーブル T_1

Class	Sex	Job	Salary(K)
2Y10N	ANY	ANY	[1-37]
19Y3N	ANY	ANY	[37-99]

表 4 テーブル T_2

Class	Sex	Job	Salary(K)
2Y10N	ANY	Blue-collar	[1-37]
3Y1N	ANY	Blue-collar	[37-99]
16Y2N	ANY	White-collar	[37-99]

表 5 テーブル T_3

Class	Sex	Job	Salary(K)
0Y7N	ANY	Non-Technical	[1-37]
2Y3N	ANY	Technical	[1-37]
3Y1N	ANY	Technical	[37-99]
16Y2N	ANY	White-collar	[37-99]

表 6 テーブル T_4

Class	Sex	Job	Salary(K)
0Y7N	ANY	Non-Technical	[1-35]
2Y3N	ANY	Technical	[1-37]
3Y1N	ANY	Technical	[37-99]
16Y2N	ANY	White-collar	[37-99]

表 7 テーブル T_5

Class	Sex	Job	Salary(K)
0Y7N	Male	Non-Technical	[1-35]
2Y3N	Male	Technical	[35-37]
3Y1N	Female	Technical	[37-99]
11Y2N	Female	White-collar	[37-99]
5Y0N	Male	White-collar	[37-99]

表 8 テーブル T_{final}

Class	Sex	Job	Salary(K)
0Y7N	Male	Non-Technical	[1-35]
2Y3N	Male	Technical	[35-37]
3Y1N	Female	Technical	[37-99]
7Y2N	Female	Manager	[37-99]
4Y0N	Female	Professional	[37-99]
5Y0N	Male	Professional	[37-99]

表 9 関数値の例 (重み係数 $w = 98$)

テーブル	ClassInfo	SplitInfo	TableInfo
T_{raw}	0.4002	3.2010	0.4562
T_{init}	0.9596	0	0.9404
T_1	0.6012	0.9366	0.6079
T_2	0.5912	1.3792	0.6070
T_3	0.5046	1.7251	0.5290
T_4	0.5046	1.7251	0.5290
T_5	0.4750	2.1763	0.5090
T_{final}	0.4405	2.5168	0.4820

5. 組織的なトップダウン k -匿名化

5.1 新しい特殊化の指標 $\text{Score}(T, \text{Cut}, v)$

表 2 のテーブル T_{init} と表 3 のテーブル T_1 を例に評価関数値の比較を行い、特殊化によるスコアを導出する．表 9 のテーブル T_{init}, T_1 の関数 TableInfo の関数値に注目する．特殊化 $[1-99] \rightarrow \{[1-37], [37-99]\}$ を行うと、テーブル T_{init}, T_1

の関数値 $TableInfo$ は 0.9404 から 0.6079 に 0.3325 だけ減少していることが分かる．この関数値の差分は属性 $Salary$ の値 $[1 - 99]$ の特殊化 $[1 - 99] \rightarrow \{[1 - 37], [37 - 99]\}$ による，テーブルの目的関数値の最小化に関する貢献であると考えることができ，目的関数値の差分を値 $[1 - 99]$ の $Score$ とする．

また，目的関数値の差分 $Score$ は， $TableInfo(T_{init}, Cut) - TableInfo(T_1, Cut')$ と表現できるが，関数 $TableInfo$ の引数に特殊化をする値 v が明示されていないため，値 v を引数に含めた関数 $Score(T, Cut, v)$ として次節で定義する．本研究において，テーブルの乱雑さの最小化を目的としているので，特殊化前後における目的関数値の差分が大きいため，すなわち関数 $Score(T, Cut, v)$ の値が最も大きい値 v で特殊化を行うことを，制約条件を満たさなくなるまで繰り返すことにより，目的関数値を最小化する．

次節で，特殊化 $v \rightarrow child(v)$ の前後における利得差を表す関数 $Score(T, Cut, v)$ を定義する．また関数 $Score(T, Cut, v)$ は，分類分析について，現在の Cut における値 v の上で特殊化の“よさ”を示す．関数 $Score(T, Cut, v)$ の計算式は，式 (12) であり， $Score(T, Cut, v)$ を計算する際に必要となる $InfoGain(T, Cut, v)$ と $SplitInfoGain(T, Cut, v)$ の計算式について，順に示していく．

$$\begin{aligned} Score(T, Cut, v_i) &= w \times ClassInfoGain(T, Cut, v_i) \\ &+ (1 - w) \times SplitInfoGain(T, Cut, v_i) \end{aligned} \quad (12)$$

5.1.1 テーブル T のクラスに関する乱雑さの利得差関数

$$ClassInfoGain(T, Cut, v_i)$$

現在のカット Cut から，第 i 番目の準識別子の分類木のカット cut_i に属する値 v_i で特殊化 $v_i \rightarrow child(v_i)$ を行った後のカット $Cut' = (Cut - \{v_i\}) \cup child(v_i)$ とする．ここで， $child(v_i)$ は分類木における値 v_i の子の集合である．テーブル T のクラスに関する乱雑さの利得差関数 $ClassInfoGain(T, Cut, v_i)$ は，式 (13) になる．

$$\begin{aligned} ClassInfoGain(T, Cut, v_i) &= ClassInfo(T, Cut) - ClassInfo(T, Cut') \end{aligned} \quad (13)$$

値 v_i や子の値 $child(v_i)$ を用いて表すと式 (14) となる．式 (14) の関数 Val は，テーブル $T(ID, D_1, \dots, D_p, Class)$ の一般化テーブルにおけるグループ $group$ と整数 $i (1 \leq i \leq p)$ が与えられたときに， $group$ に属する全てのレコードが共通して持っている i 番目の準識別子属性に対応する値を返す関数とする．

$$\begin{aligned} ClassInfoGain(T, Cut, v_i) &= \sum_{\substack{group \in Group(T, Cut) \\ \wedge Val(group, i) = v_i}} \frac{|group|}{|T|} \cdot E(group) \\ &- \sum_{\substack{group \in Group(T, Cut') \\ \wedge Val(group, i) \in child(v_i)}} \frac{|group|}{|T|} \cdot E(group) \end{aligned} \quad (14)$$

5.1.2 テーブル T のレコードに関する乱雑さの利得差関数

$$SplitInfoGain(T, Cut, v_i)$$

テーブル T のレコードに関する乱雑さの利得差関数を $SplitInfoGain(T, Cut, v_i)$ とするとき，式は次のように定義される．

$$\begin{aligned} SplitInfoGain(T, Cut, v_i) &= \\ &= \sum_{\substack{group \in Group(T, Cut) \\ \wedge Val(group, i) = v_i}} \frac{|group|}{|T|} \cdot \log_2 \frac{|group|}{|T|} \\ &- \sum_{\substack{group \in Group(T, Cut') \\ \wedge Val(group, i) \in child(v_i)}} \frac{|group|}{|T|} \cdot \log_2 \frac{|group|}{|T|} \end{aligned} \quad (15)$$

5.2 組織的なトップダウン k -匿名化アルゴリズム

前節で定義した特殊化の指標 $Score$ を用いたアルゴリズムのフローチャートを図 6 に示す．最初に分類木 $Tree$ の最上の値でテーブル，森のカット Cut を初期化することにより，匿名化を行うトップダウンなアルゴリズムである．

このアルゴリズムと前述した図 5 の TDS フレームワークとの大きな違いは， $Score$ の更新に関する部分である．従来は特殊化する値の子集合に対して $Score$ を計算していたが，提案するアルゴリズムでは特殊化をするたびに Cut に属する全ての値に対して，新しい Cut を使って $Score$ を更新する．これにより，特殊化前後のテーブル全体がどれだけ分類分析に適しているかを表す目的関数値 $TableInfo$ を活かした特殊化を選択することが可能となる．

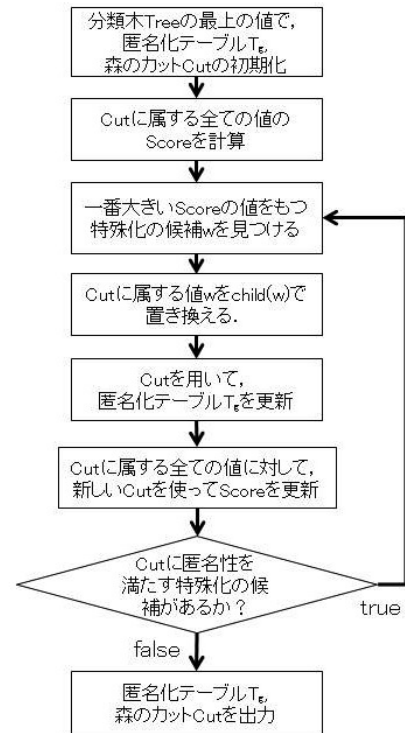


図 6 組織的なトップダウン k -匿名化アルゴリズムのフローチャート

6. まとめと今後の課題

本研究では，分類分析を目的として作成された k -匿名化テーブルを評価する関数を提案した．さらに，階層的匿名化フレームワークで利用する評価関数を導出した．

今後の課題としては，提案階層的 k -匿名化手法の実験的性能評価，および，集中的に管理されたテーブルのみならず，分散環境下にあるテーブルに対してもプライバシーを保護しながら k -匿名化テーブルを作成する手法の考案があげられる．

文 献

- [1] Noman Mohammed, Benjamin C. M. Fung and Mourad Debbabi “Anonymity meets game theory:secure data integration with malicious participants,” VLDB Journal, pp. 567-588, 2011.
- [2] Jiawei Han and Micheline Kamber, “Data Mining : Concepts and Techniques,” Morgan Kaufman Pub., 2001.
- [3] Benjamin C.M., Fung, Ke Wang, and Philip S. Yu, Fellow, IEEE “Anonymizingclassificationdata for privacy preservation,” IEEE TKDE 19(5), 711725, 2007.
- [4] Ke Wang, Benjamin C. M. Fung, and Guozhu Dong, “Integrating Private Databases for Data Analysis,” Intelligence and Security Informatics Lecture Notes in Computer Science Volume 3495, pp171-182, 2005.