プライバシーを保護するカウント演算の多値属性分類への適用

高見澤秀久†* 有次 正義††

† 群馬大学大学院工学研究科電子情報工学専攻 〒376–8515 群馬県桐生市天神町 1–5–1
†† 群馬大学工学部情報工学科 〒376–8515 群馬県桐生市天神町 1–5–1
* 東芝ソリューション（株）IT 技術研究所 〒183–8512 東京都府中市片町 3–22
E-mail: †takamiza@dbms.cs.gunma-u.ac.jp, ††aritsugi@cs.gunma-u.ac.jp

あらまし プライバシーを保護しながらデータを効果的に処理することは重要な課題である．本稿では，プライバシー保護のために摂動（perturbation）されたテーブルから，目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を再構築する手法を提案する．目的属性が 3 値以上の場合，従来手法では目的属性の各値の演算結果をそれぞれ独立に再構築しなければならない．そこで，本稿では従来手法を拡張し，目的属性の各値の演算結果を一括して再構築する手法を提案する．そして，提案手法はテーブルを摂動する割合が高い場合の再構築エラーを従来手法よりも低減させることを実験により示す．

キーワード プライバシー保護，データマイニング

Applying Privacy Preserving Count Aggregate Queries to k-Classification

Hidehisa TAKAMIZAWA†* and Masayoshi ARITSUGI††

† Department of Computer Science, Graduate School of Engineering, Gunma University
1–5–1 Tenjin-cho, Kiryu, Gunma, 376–8515 Japan
†† Department of Computer Science, Faculty of Engineering Gunma University
1–5–1 Tenjin-cho, Kiryu, Gunma, 376–8515 Japan
* Advanced IT Laboratory, Toshiba Solutions Corporation
3–22 Kata-machi, Fuchu, Tokyo, 183–8512 Japan
E-mail: †takamiza@dbms.cs.gunma-u.ac.jp, ††aritsugi@cs.gunma-u.ac.jp

Abstract It is important to process data effectively while preserving privacy. In this paper, we propose a reconstruction technique of count aggregate queries, which are necessary for building a decision tree, from a perturbed table in cases where a target attribute is more than binary. In the conventional technique, we must reconstruct the results of target values from those of each value calculated independently when a decision tree has a non-binary target attribute. In this paper, we borrow and extend the conventional technique to reconstruct the results of target values at once. We also report some experimental results showing that our proposal can reduce reconstruction errors compared to the conventional technique in cases where perturbation ratio is high.

Key words Privacy Preserving, Data Mining

1. はじめに

近年，プライバシー保護データマイニングの研究が盛んに行われている [1]〜[4]．データマイニングは膨大なデータから有用な知識を抽出するための技術として幅広く活用されている．しかしながら，近年のインターネットの発達に伴い，データマイニングによって個人のプライバシーが侵害される可能性あるからである [5].

1.1 モデル

本研究で扱うモデルを図 1 に示す．ここでは，サーバと，そのサーバに接続された n 個のクライアント $Client_1$，$Client_2$，$\cdots$，$Client_n$ により構成されるモデルを考える．

各クライアントは $m+1$ 個の属性 $Attr_0$，$Attr_1$，$Attr_2$，$\cdots$，$Attr_m$ からなるレコードをそれぞれ 1 つずつもっている．ここで，属性 $Attr_0$ は決定木分類における目的属性でカテゴリ属性である．他の属性 $Attr_1$，$\cdots$，$Attr_m$ は説明属性である．各クライアントは自身がもつレコードのプライバシーを保護するた

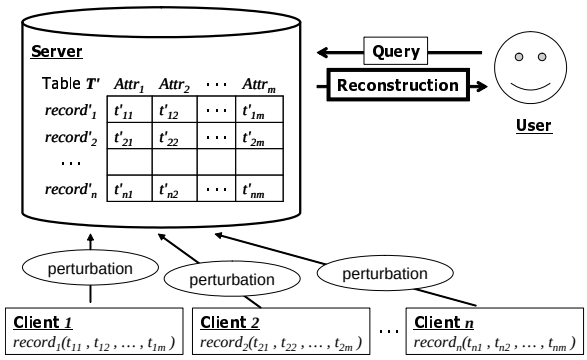

図 1 モ デ ル
Fig. 1 Model.

めに，摂動（perturbation）されたレコードをサーバに送信する．サーバは，すべてのクライアントから受信した n 個の摂動されたレコードから，テーブル T' を構成する．すなわち，サーバは各クライアントにおいて摂動されたレコードの値は知ることができるが，各クライアントが所持する元のレコードの値を知ることはできない．したがって，このモデルでは摂動されていない元のテーブル T が構成されることはない．ただし，以降は説明を簡略化するため，テーブル T の各レコードを独立に摂動することで，テーブル T' を構成するものと考える．

1.2 目　　的

本研究の目的は，摂動されたテーブルから，目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を再構築することである．本研究と同じモデルで摂動されたテーブルからカウント演算結果を再構築する手法は文献 [4] においても提案されている．文献 [4] では，ある一定の確率で同じ属性の定義域内の別のランダムな値に置き換える方法（維持 - 置換摂動：Retention Replacement Perturbation）により摂動されたテーブルから，決定木を構築するために必要なカウント演算結果を再構築する手法を提案している．

しかしながら，文献 [4] で提案されている手法を用いて，摂動されたテーブルから目的属性が 3 値以上である決定木を構築するために必要なカウント演算結果を再構築する場合，目的属性の各値の演算結果をそれぞれ独立に再構築しなければならない．そこで，本稿では文献 [4] で提案されている再構築手法を拡張して，目的属性の各値の演算結果を一括して再構築する手法を提案する．そして，従来手法と提案手法の再構築エラーを実験により比較する．

1.3 本稿の構成

本稿の構成を以下に示す．2. では関連研究及び本研究の基となる文献 [4] の摂動手法および再構築手法を説明する．3. では，2. で紹介した再構築手法により，摂動されたテーブルから決定木を構築するために必要なカウント演算結果を再構築する手順を説明する．そして，4. で，摂動されたテーブルから目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を一括して再構築する手法を提案する．5. では従来手法と提案手法とによる再構築エラーを比較する実験を行い，6. でまとめとする．

2. 準　　備

2.1 関 連 研 究

統計データベースの分野では，統計量を取得する問合せに対するプライバシー保護についての研究が行われている [6]．統計データベースのモデルでは，プライバシー情報を含むテーブルをもつサーバが，自身のもつテーブルに対してプライバシーを保護するための処理を施す．一方，本研究で扱うモデルでは，レコードをもつ各クライアントが，それぞれプライバシー保護のための摂動処理を行う．すなわち，サーバは摂動されたレコードの値しか知りえないプライバシー保護モデルであるため，従来の統計データベースにおけるプライバシー保護モデルとは異なる．

文献 [2] では複数のデータベース間でお互いがもつ個々のレコードの内容を秘匿したまま，決定木を構築するための計算を行う手法を提案している．しかしながら，文献 [2] におけるモデルでは各データベースが一定量のレコードをもつ必要があるため，本研究で扱うモデルとは異なる．

本研究におけるプライバシー保護モデルは，文献 [1], [4] でも扱われている．文献 [1] では，各クライアントがもつレコードに対して，属性ごとに独立な乱数をノイズとして加算することでプライバシー保護を行う．そして，摂動されたテーブルから各属性の分布を再構築する．しかしながら，文献 [1] では，元の分布の再構築は各属性で独立に行わなければならないという問題がある．文献 [4] では，ある一定の確率で同じ属性の定義域内の別のランダムな値に置き換える手法により，各クライアントがもつレコードを摂動する．そして，摂動されたテーブルから，決定木を構築するために必要なカウント演算結果を再構築する．本研究は文献 [4] を基に考える．以下では，文献 [4] で提案されている摂動手法と再構築手法について説明する．

2.2 プライバシー保護 OLAP

2.2.1 維持 - 置換摂動（Retention Replacement Perturbation）

文献 [4] では，各レコードを摂動するために維持 - 置換摂動を用いている．維持 - 置換摂動は，レコードの各属性値を，属性ごとに共通なある一定の確率で同じ属性の定義域内の別の値に置換する手法である．すなわち，この手法では摂動後も元の値がある一定の確率で維持される．この確率を維持確率（retention probability）として属性ごとに設定する．ここでは，ある属性 $Attr_j$ における維持確率を $rp_j (0 \leqq rp_j \leqq 1)$ と表現する．維持確率が小さい，つまり元の値が維持されずに他の値に置換される確率が高くなるほど摂動の割合も高くなり，プライバシーが保護されることになる．

維持 - 置換摂動は以下のように定式化することができる．ここで，t_{ij} は元のテーブル T の i 番目のレコードにおける j 番目の属性値，そして t'_{ij} は摂動されたテーブル T' の i 番目のレコードにおける j 番目の属性値である．

$$t'_{ij} = \begin{cases} t_{ij}, & (\text{確率 } rp_j) \\ \text{定義域内の他の値に置換}, & (\text{確率 } 1 - rp_j) \end{cases} \quad (1)$$

なお，図 1 のモデルでは，各クライアントはすべての属性の維持確率と定義域を知っているものとし，それにより各クライアントは自身がもつレコードを独立に摂動することができる．

2.2.2 反復ベイズ手法（Iterative Bayesian technique）

続いて，文献 [4] で提案されている，維持 - 置換摂動により摂動されたテーブル T' からカウント演算結果を再構築する手法である，反復ベイズ手法について説明する．ここでは，テーブル T における k 個の属性に対してカウント演算 $COUNT(P_1 \wedge P_2 \wedge \ldots \wedge P_k)$ を実行した結果を再構築することを考える．なお，P_j はテーブル T における j 番目の属性に対する条件式であり，例えば数値属性の場合は $25 \leqq age \leqq 40$，カテゴリ属性の場合は $risk = high$ といった条件式となる．

ここで，文献 [4] では，反復ベイズ手法だけでなく，逆行列の計算を用いた再構築手法である行列反転手法（Matrix Inversion technique）も提案している．しかし，行列反転手法では，属性数の増加に伴い再構築エラーが大きくなることが実験により示されているため，本稿では反復ベイズ手法のみを用いる．

まず，元のテーブルにおけるカウント演算結果を表すベクトル $\boldsymbol{x} = (x_0, x_1, ..., x_t)$ と，摂動されたテーブルにおけるカウント演算結果を表すベクトル $\boldsymbol{y} = (y_0, y_1, ..., y_t)$ を以下のように定義する．

$$\begin{cases} x_i = COUNT(\wedge_{r=1}^{k} Q(r, bit(i,r))) \\ \qquad\qquad \text{in } T, \text{ for } 0 \leqq i \leqq t \\ y_i = COUNT(\wedge_{r=1}^{k} Q(r, bit(i,r))) \\ \qquad\qquad \text{in } T', \text{ for } 0 \leqq i \leqq t \end{cases} \tag{2}$$

ここで，$t = 2^k - 1$ である．また，$Q(r,i)$ は r 番目の属性に対する条件式であり以下の式により計算する．

$$Q(r,i) = \begin{cases} \neg P_r, & \text{if } i = 0 \\ P_r, & \text{if } i = 1 \end{cases} \tag{3}$$

$bit(i,r)$ は，整数 i を k ビットの二進数で表現した値の左から r 番目のビットである．そして，あるレコードが条件 $\wedge_{r=1}^{k} Q(r, bit(i,r))$ を満たすとき，そのレコードの状態を i とする．

例として，2 つの属性に対するカウント演算と，そのカウント演算結果に対応する x_i および y_i を表 1 に示す．ここでは，x_i および y_i の添字 i がレコードの状態を表している．

表 1　カウント演算と $\boldsymbol{x}$，$\boldsymbol{y}$

Table 1　Vectors $\boldsymbol{x}$,$\boldsymbol{y}$, and count aggregate queries.

カウント演算	$\boldsymbol{x}$	$\boldsymbol{y}$
$COUNT(\neg P_1 \wedge \neg P_2)$	x_{00}	y_{00}
$COUNT(\neg P_1 \wedge P_2)$	x_{01}	y_{01}
$COUNT(P_1 \wedge \neg P_2)$	x_{10}	y_{10}
$COUNT(P_1 \wedge P_2)$	x_{11}	y_{11}

次に，元のテーブル T での状態が p であったレコードが，摂動されたテーブル T' での状態が q となる遷移確率 a_{pq} は，以下の式により計算することができる．

$$a_{pq} = \prod_{r=1}^{k} \left\{ (1 - rp_r) \cdot R_{r,bit(q,r)} + rp_r \cdot \delta_{(bit(p,r),bit(q,r))} \right\} \tag{4}$$

ここで，$R_{(r,i)}$ は以下の式により計算する．

$$R_{(r,i)} = \begin{cases} 1 - b_r, & \text{if } i = 0 \\ b_r, & \text{if } i = 1 \end{cases} \tag{5}$$

b_r は条件式の範囲を表すものであり，以下の式により計算する．

$$b_r = \begin{cases} \dfrac{high_r - low_r}{max_r - min_r}, & (Attr_r \text{ が数値属性の場合}) \\ \dfrac{1}{c_r}, & (Attr_r \text{ がカテゴリ属性の場合}) \end{cases} \tag{6}$$

属性 $Attr_r$ が数値属性の場合，$high_r$，low_r は，それぞれ属性 $Attr_r$ に対する条件式 P_r の上限値と下限値であり，max_r，min_r は，それぞれ属性 $Attr_r$ の定義域の上限値と下限値である．属性 $Attr_r$ がカテゴリ属性の場合，c_r は属性 $Attr_r$ の属性値の個数である．$\delta_{(i,j)}$ はクロネッカのデルタ（Kronecker's delta）であり，以下の式により計算する．

$$\delta_{(i,j)} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{7}$$

続いて，これらの式を用いて，摂動されたテーブル T' におけるカウント演算結果 $\boldsymbol{y}$ から元のテーブル T におけるカウント演算結果 $\boldsymbol{x}$ を再構築する手順を示す．まず，テーブル T における n 個の各レコードの状態をそれぞれ $U_1, U_2, ..., U_n$，テーブル T' における n 個の各レコードの状態をそれぞれ $V_1, V_2, ..., V_n$ とする．すなわち，$0 \leqq p, q \leqq 2^k - 1$ および $1 \leqq i \leqq n$ に対して，$P(U_i = p) = x_p/n$，$P(V_i = q) = y_q/n$ となる．また，$P(U_i = p)$ は以下の式で表すことができる．

$$P(U_i = p) = \sum_{q=0}^{t} P(V_i = q) P(U_i = p | V_i = q) \tag{8}$$

ここで，$P(U_i = p | V_i = q)$ は，ベイズの定理を用いて以下の式により表すことができる．

$$\begin{aligned} P(U_i = p | V_i = q) &= \frac{P(V_i = q | U_i = p) P(U_i = p)}{P(V_i = q)} \\ &= \frac{P(V_i = q | U_i = p) P(U_i = p)}{\sum_{r=0}^{t} P(V_i = q | U_i = r) P(U_i = r)} \\ &= \frac{a_{pq} x_p}{\sum_{r=0}^{t} a_{rq} x_r} \end{aligned} \tag{9}$$

そして，式 (8) に，式 (9) を代入することで，以下の式を求めることができる．

$$x_p^{T+1} = \sum_{q=0}^{t} y_q \frac{a_{pq} x_p^T}{\sum_{r=0}^{t} a_{rq} x_r^T} \tag{10}$$

ここで，$\boldsymbol{x}^T$ は，T 番目の繰返しを，$\boldsymbol{x}^{T+1}$ は，$T+1$ 番目の繰返しをそれぞれ示しており，初期値を $\boldsymbol{x}^0 = \boldsymbol{y}$ として，連続する $\boldsymbol{x}$ の違いが少なくなるまで，繰返し $\boldsymbol{x}$ を計算する．

3. 摂動されたテーブルからの決定木の構築

本研究の目的は，目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を摂動されたテーブルから再構築することである．そこで本節ではまず，決定木を構築するために必要となるカウント演算について説明する．続いて，2.2.2 で説明した手法を用いて，目的属性が 2 値および 3 値以上の決定木を構築するために必要なカウント演算結果を摂動されたテーブルから再構築する手順をそれぞれ説明する．

3.1 決定木の構築に必要なカウント演算

決定木 [7]〜[9] における最適な分割点を決定するための評価指標としては，一般に相互情報量（mutual information），ジニ係数（gini index），χ^2（chi square）値等が用いられる [9]〜[11]．表 2 にそれぞれの評価関数（$Ent, Gini, Chi$）を示す．なお，S をデータベース中の全レコードの集合，$p_i(S_j)$ を S の部分集合 $S_j \subseteqq S$ 中の目的属性が C_i であるレコードの割合とする．また，目的属性の属性値の個数を c とする．

表 2　一般的な評価関数
Table 2　General objective functions.

評価指標	関数
相互情報量	$Ent = H(S) - \frac{\|S_1\|H(S_1)+\|S_2\|H(S_2)}{\|S\|}$ $H(S) = -\sum_{i=1}^{c} p_i(S)\log p_i(S)$
ジニ係数	$Gini = G(S) - \frac{\|S_1\|G(S_1)+\|S_2\|G(S_2)}{\|S\|}$ $G(S) = 1 - \sum_{i=1}^{c} p_i^2(S)$
χ^2 値	$Chi = C(S,S_1) + C(S,S_2)$ $C(S,S_i) = \sum_{j=0}^{c} \frac{(p_j(S_i)-p_j(S))^2}{p_i(S)}$

例として，集合 S を条件 P_1 により二つの集合 S_1 および S_2 に分割する場合を考える．ここで，条件 P_0^i を目的属性が C_i であるとすれば，分割された各集合 S_1，S_2 が P_0^i を満たす割合 $p_i(S_1)$，$p_i(S_2)$ を，以下の式により表すことができる．

$$\begin{cases} p_i(S_1) = \dfrac{COUNT(P_0^i \wedge P_1)}{COUNT(P_1)} \\ p_i(S_2) = \dfrac{COUNT(P_0^i \wedge \neg P_1)}{COUNT(\neg P_1)} \end{cases} \tag{11}$$

このように，どの評価関数も分割前後の各属性値の個数の関数として表現することができる．つまり，最適な分割を探索する際にはカウント演算結果を定数として扱うことができる．そこで，以下では目的属性が 2 値および 3 値以上の決定木に必要なカウント演算結果を摂動されたテーブルから再構築する手順についてのみ説明する．なお，目的属性が 3 値以上の決定木を構築する際の最適な分割を探索する手法は，文献 [7], [10], [11] 等で述べられている．

3.2 2 値属性分類のためのカウント演算

まず，2.2.2 で説明した手法を用いて，目的属性が 2 値 C_0，C_1 の決定木を構築するために必要なカウント演算結果を摂動されたテーブルから再構築する手順を説明する．

目的属性が 2 値の場合は，$P_0^1 = \neg P_0^0$ である．したがって，条件 P_1 により分割する場合，目的属性が 2 値の決定木を構築するために必要なカウント演算は表 3 となる．

表 3　2 値属性分類のためのカウント演算
Table 3　Count aggregate queries for a binary target attribute.

目的属性	条件 P_1 で分割する場合のカウント演算
C_0	$COUNT(P_0^1 \wedge \neg P_1)$
	$COUNT(P_0^1 \wedge P_1)$
C_1	$COUNT(\neg P_0^1 \wedge \neg P_1)$
	$COUNT(\neg P_0^1 \wedge P_1)$

このように，目的属性が 2 値の決定木を構築するために必要なカウント演算結果は，2.2.2 の再構築処理を一回行うことにより得ることができる．

3.3 多値属性分類のためのカウント演算

続いて，2.2.2 で説明した手法を用いて，目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を摂動されたテーブルから再構築する手順を説明する．目的属性が 3 値以上の場合は，属性値ごとのカウント演算結果が必要となる．例えば，目的属性が 3 値 C_0, C_1, C_2 である集合を条件 P_1 により分割する場合，表 4 のカウント演算が必要となる．

表 4　3 値属性分類のためのカウント演算
Table 4　Count aggregate queries for a 3-values target attribute.

目的属性	条件 P_1 で分割する場合のカウント演算
C_0	$COUNT(P_0^0 \wedge \neg P_1)$
	$COUNT(P_0^0 \wedge P_1)$
C_1	$COUNT(P_0^1 \wedge \neg P_1)$
	$COUNT(P_0^1 \wedge P_1)$
C_2	$COUNT(P_0^2 \wedge \neg P_1)$
	$COUNT(P_0^2 \wedge P_1)$

ここで，従来手法により一回の処理で再構築できるカウント演算を表 5 に示す．なお，$\boldsymbol{x}^i$ を，目的属性の属性値が C_i であるか否かの条件 P_0^i または $\neg P_0^i$ をもつカウント演算結果を表すベクトルとする．

表 5　従来手法によって再構築可能なカウント演算
Table 5　Reconstructable count aggregate queries by the conventional technique.

目的属性	条件 P_1 で分割する場合のカウント演算	$\boldsymbol{x}^i$
C_i	$COUNT(\neg P_0^i \wedge \neg P_1)$	x_{00}^i
	$COUNT(\neg P_0^i \wedge P_1)$	x_{01}^i
	$COUNT(P_0^i \wedge \neg P_1)$	x_{10}^i
	$COUNT(P_0^i \wedge P_1)$	x_{11}^i

そのため，従来手法では，目的属性の各値のカウント演算結果 $\boldsymbol{x}^1$，$\boldsymbol{x}^2$，$\boldsymbol{x}^3$ をそれぞれ独立に再構築しなければ，表 4 に示すすべてのカウント演算，すなわち，$COUNT(P_0^i \wedge \cdots)$ の結果を得ることができない．

4. 提 案 手 法

前述の通り，摂動されたテーブルから，目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を従来手法を用いて再構築する場合，目的属性の各値においてそれぞれ独立に再構築処理を行わなくてはならない．そこで，本節では，目的属性の各値のカウント演算結果を一括して再構築する手法を提案する．なお，摂動手法については，従来手法と同様，維持 - 置換摂動を用いる．したがって，摂動されたテーブルは，文献 [4] と同等のプライバシー保護が保証される．

以下では，2.2.2 の説明を基にして，本研究で提案する再構築手法について説明する．

4.1 レコードの状態の再定義

まず，2.2.2 で定義したレコードの状態を再定義する．従来手法では，各属性の状態を示すビットにより，条件 P_i，$\neg P_i$ を満たすかどうかをそれぞれ 1，0 として表現していた．目的属性が 2 値の場合には，あるレコードの目的属性が「一方の属性値をもつ／もたない」として，目的属性の状態を 1 ビットで表現することができた．しかしながら，目的属性が 3 値以上の場合には，目的属性の状態を 1 ビットで表現することはできない．そこで，本研究では目的属性の状態をビット列として表現する．

目的属性の属性値の個数を c とすれば，目的属性の状態を表すために必要となるビット数は，$\lceil \log_2 c \rceil$ により計算できる．したがって，レコードの状態は，目的属性の状態を表す長さ $\lceil \log_2 c \rceil$ のビット列と，m 個の説明属性の状態を表す長さ m のビット列を連結したものとなる．

例として，3 値 C_0, C_1, C_2 を属性値としてもつ目的属性と，1 つの説明属性からなるレコードが取り得る状態を，その状態に対する条件式と対応付けて表 6 に示す．ここで，P_0^i は目的属性の属性値が C_i であることを表す条件式，そして P_1 は説明属性に対する条件式である．

表 6　提案手法におけるレコードの状態
Table 6　Status of a record in our proposal.

状態	条件式	目的属性の値
000	$P_0^0 \wedge \neg P_1$	C_0
001	$P_0^0 \wedge P_1$	
010	$P_0^1 \wedge \neg P_1$	C_1
011	$P_0^1 \wedge P_1$	
100	$P_0^2 \wedge \neg P_1$	C_2
101	$P_0^2 \wedge P_1$	

ここでは，目的属性が 3 値であるため，目的属性の状態は $\lceil \log_2 3 \rceil = 2$ ビットで表現することができる．また，説明属性の状態は 1 ビットで表現することができるため，レコードの状態は $2+1=3$ ビットで表現することができる．

4.2 カウント演算の定式化

次に，4.1 で定義した各状態に対するカウント演算を定式化する．2.2.2 と同様に，元のテーブルにおけるカウント演算結果を表すベクトル $\boldsymbol{x} = (x_0, x_1, ..., x_t)$ と，摂動されたテーブルにおけるカウント演算結果を表すベクトル $\boldsymbol{y} = (y_0, y_1, ..., y_t)$ を以下のように定義する．

$$\begin{cases} x_i = COUNT(P_0^{left(i,\lceil \log_2 c \rceil)} \wedge_{r=1}^{k} Q(r, bit(i,r))) \\ \qquad\qquad \text{in } T, \text{ for } 0 \leqq i \leqq t' \\ y_i = COUNT(P_0^{left(i,\lceil \log_2 c \rceil)} \wedge_{r=1}^{k} Q(r, bit(i,r))) \\ \qquad\qquad \text{in } T', \text{ for } 0 \leqq i \leqq t' \end{cases} \tag{12}$$

ここで $t' = c \cdot 2^k - 1$ である．また，$left(i,r)$ を i の左から r ビットの整数表現とする．したがって，$left(i, \lceil \log_2 c \rceil)$ は，値 i の左から $\lceil \log_2 c \rceil$ ビット，すなわち目的属性の属性値を表しているため，式 $P_0^{left(i,\lceil \log_2 c \rceil)}$ は $left(i, \lceil \log_2 c \rceil)$ によって表される属性値を選択するための条件式となる．例えば，目的属性が 3 値，$i = 010_{(2)}$ の場合，$P_0^{left(010_{(2)},2)} = P_0^{01_{(2)}} = P_0^1$ となる．なお，$Q(r,i)$ および $bit(i,j)$ の定義は 2.2.2 と同様である．

例として，3 値の目的属性と 1 つの説明属性からなるテーブルを考えた場合，$\boldsymbol{x}$ および $\boldsymbol{y}$ に対応するカウント演算を表 7 に示す．

表 7　提案手法によって再構築可能なカウント演算
Table 7　Reconstructable count aggregate queries by the proposed technique

カウント演算	$\boldsymbol{x}$	$\boldsymbol{y}$	目的属性
$COUNT(P_0^0 \wedge \neg P_1)$	x_{000}	y_{000}	C_0
$COUNT(P_0^0 \wedge P_1)$	x_{001}	y_{001}	
$COUNT(P_0^1 \wedge \neg P_1)$	x_{010}	y_{010}	C_1
$COUNT(P_0^1 \wedge P_1)$	x_{011}	y_{011}	
$COUNT(P_0^2 \wedge \neg P_1)$	x_{100}	y_{100}	C_2
$COUNT(P_0^2 \wedge P_1)$	x_{101}	y_{101}	

ここでは，目的属性の状態は，$left(i, \lceil \log_2 3 \rceil) = 2$ ビットにより表すことができるため，i の左から 2 ビットは目的属性，そして残りの 1 ビットは説明属性の状態として表される．

4.3 遷移確率の再定義

続いて，状態 p のレコードが状態 q となる遷移確率 a'_{pq} を以下に示す．

$$a'_{pq} = \left\{(1-rp_0)\cdot b_0 + rp_0 \cdot \delta_{(left(p,\lceil \log_2 c \rceil), left(q,\lceil \log_2 c \rceil))}\right\} \cdot \prod_{r=1}^{k} \left\{(1-rp_r)\cdot R_{r,bit(q,r)} + rp_r \cdot \delta_{(bit(p,r),bit(q,r))}\right\} \tag{13}$$

説明属性においては，条件 P_r と $\neg P_r$ の範囲が一致するとは限らないので，置換された値が元の状態ではなくなる確率 $1-b_r$ と，元の状態と同じになる確率 b_r が一致するわけではない．一方，目的属性においては，条件 P_0^i の範囲はすべて等しくなるため，置換された値が各属性値になる確率は均一に $b_0 = 1/c$ である．また，説明属性の状態の変化は 1 ビットを比較することで判別できるが，目的属性の状態の変化は目的属性の状態を表す長さ $left(i, \lceil \log_2 c \rceil)$ のビット列を比較する必要がある．

4.4 再構築の計算式

再構築の計算式は，従来手法の式 (10) の遷移確率 a_{pq} を式 (13) の遷移確率 a'_{pq} に置き換える．すなわち，以下の式により表される．

$$x_p^{T+1} = \sum_{q=0}^{t'} y_q \frac{a'_{pq} x_p^T}{\sum_{r=0}^{t'} a'_{rq} x_r^T} \tag{14}$$

そして，2.2.2 と同様に，$\boldsymbol{x}$ を繰返し計算すればよい．

5. 実　　験

本節では提案手法の効果を評価するために従来手法との比較実験の結果を示す．ここでは，各再構築手法の再構築エラーを，元のテーブルに対するカウント演算結果と各手法により再構築したカウント演算結果との L1 距離を元のレコードの総数で割った値とする．また，再構築手法の効果を示すため，元のテーブルに対するカウント演算結果と摂動されたテーブルに対するカウント演算結果との L1 距離についても同様に計算する．なお，再構築エラーには，各実験を 10 回繰り返して得られた結果の平均値を用いている．

実験に用いるテーブルの属性は，パラメータ 1 の Zipf 分布に従う．目的属性は 2 値〜10 値とし，実験により値を変化させる．説明属性の属性値の個数は 1,000 個とし，説明属性の上位 400 個をカウント演算の条件とする．テーブルを構成するレコードの個数は 10^4 個または 10^5 個とする．維持確率は各属性で同じ値とする．以下では簡略化のため，目的属性の属性値の個数を c，説明属性の個数を m，テーブルを構成するレコードの個数を n，維持確率を rp として説明する．

5.1 維持確率を変化させた場合の再構築エラー

まず，目的属性が 5 値（$c = 5$）と 10 値（$c = 10$）で，1 個の説明属性をもつ 10^4 個のレコードからなるテーブルを用いた場合の実験結果を，それぞれ図 2 および図 3 に示す．

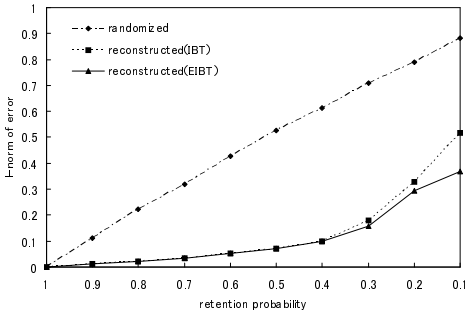

図 2　$c = 5, m = 1, n = 10^4$ で異なる rp の再構築エラー
Fig. 2　Errors for $c = 5, m = 1, n = 10^4$ with varying rp.

図 2 および図 3 の縦軸は，摂動されたテーブルに対するカウント演算結果（randomized），反復ベイズ手法によって再構築したカウント演算結果（IBT），そして提案手法によって再構築したカウント演算結果（EIBT）と，元のテーブルに対するカウント演算結果との再構築エラー（L1 距離）であり，横軸は維持確率（rp）を表している．

維持確率が高い場合，提案手法と従来手法との差はほとんどない．しかしながら，維持確率が低い場合においては，従来手法に比べて提案手法の再構築エラーが小さくなっている．これは，維持確率が高い場合には，目的属性の各属性値を独立に処理することによる再構築エラーが小さいためと考えられる．一方，維持確率が低い場合においては，テーブル内での各属性値のカウント演算結果の依存関係を考慮した分，提案手法の方が再構築エラーが小さいと考えられる．

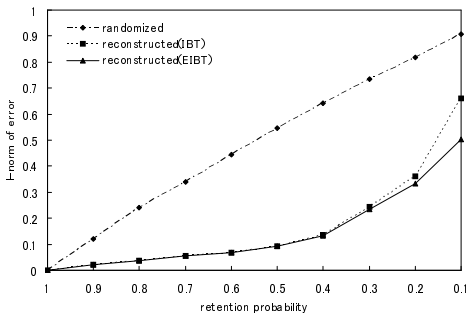

図 3　$c = 10, m = 1, n = 10^4$ で異なる rp の再構築エラー
Fig. 3　Errors for $c = 10, m = 1, n = 10^4$ with varying rp.

5.2 目的属性の属性値の個数を変化させた場合の再構築エラー

この実験では，目的属性を 2 値から 10 値とするテーブルを用いた場合の再構築エラーをそれぞれ確認する．維持確率を 0.2 および 0.1 とした場合の結果をそれぞれ図 4 および図 5 に示す．

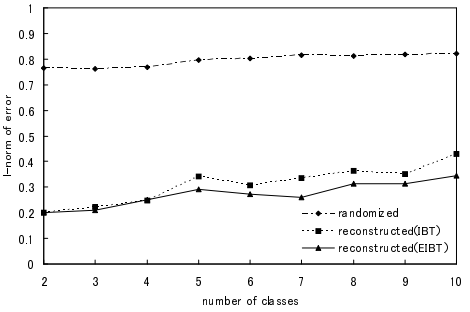

図 4　$m = 1, n = 10^4, rp = 0.2$ で異なる c の再構築エラー
Fig. 4　Errors for $m = 1, n = 10^4, rp = 0.2$ with varying c.

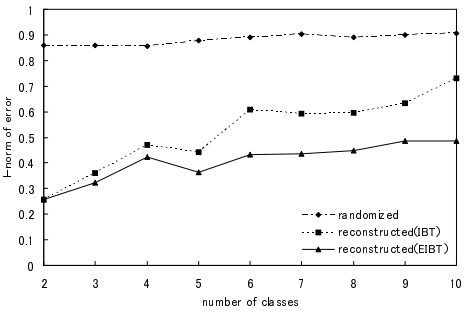

図 5　$m = 1, n = 10^4, rp = 0.1$ で異なる c の再構築エラー
Fig. 5　Errors for $m = 1, n = 10^4, rp = 0.1$ with varying c.

目的属性が 2 値の場合は，従来手法と提案手法の手順は等しいため再構築エラーも等しくなる．その他の場合（$3 \leqq c \leqq 10$）では，提案手法の再構築エラーは従来手法に比べて小さくなっ

ている．

5.3 説明属性の個数を変化させた場合の再構築エラー

続いて，5.1 の環境において，説明属性の個数を 2 個にした場合の結果を図 6 および図 7 に示す．追加する説明属性は，5.1 の説明属性と同じ性質をもつものとする．同様に，説明属性の個数を 3 個にした場合の結果を図 8 および図 9 に示す．

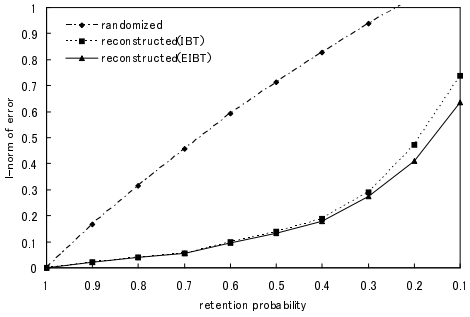

図 6　$c = 5, m = 2, n = 10^4$ で異なる rp の再構築エラー
Fig. 6　Errors for $c = 5, m = 2, n = 10^4$ with varying rp.

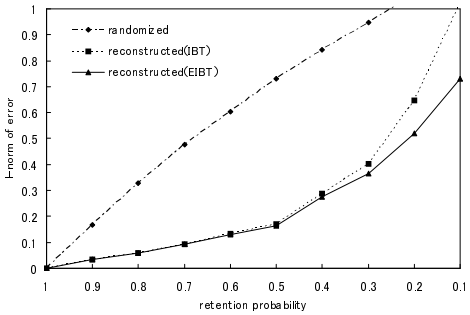

図 7　$c = 10, m = 2, n = 10^4$ で異なる rp の再構築エラー
Fig. 7　Errors for $c = 10, m = 2, n = 10^4$ with varying rp.

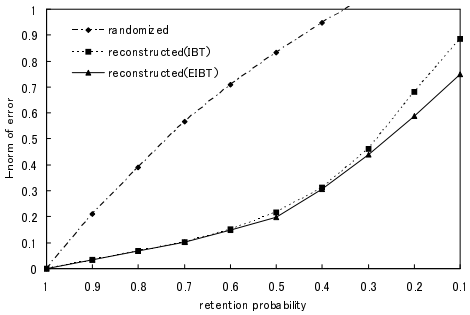

図 8　$c = 5, m = 3, n = 10^4$ で異なる rp の再構築エラー
Fig. 8　Errors for $c = 5, m = 3, n = 10^4$ with varying rp.

これらの結果と図 2，図 3 とを比較してみると全体的に再構築エラーが大きくなっているが，維持確率が低い場合の再構築エラーは従来手法よりも提案手法の方が低くなっている．

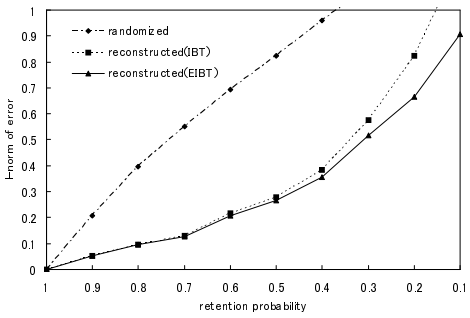

図 9　$c = 10, m = 3, n = 10^4$ で異なる rp の再構築エラー
Fig. 9　Errors for $c = 10, m = 3, n = 10^4$ with varying rp.

5.4 レコード数を増加させた場合の再構築エラー

次に，テーブルのレコード数を増加させた場合の再構築エラーを確認する．まず，レコード数を 10^5 個にして 5.1 と同様の実験を行った結果を図 10 および図 11 に示す．同様に 5.2 の環境でレコード数を増加させた場合の結果を図 12 および図 13 に示す．

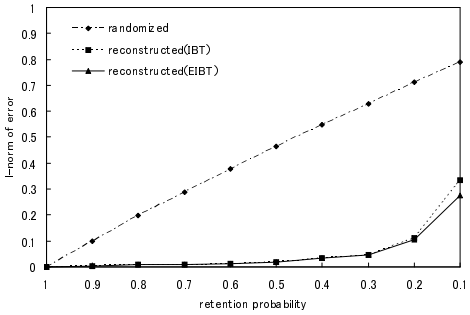

図 10　$c = 5, m = 1, n = 10^5$ で異なる rp の再構築エラー
Fig. 10　Errors for $c = 5, m = 1, n = 10^5$ with varying rp.

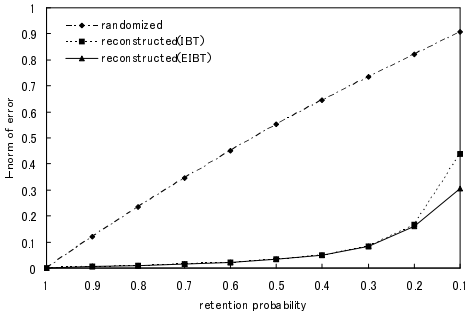

図 11　$c = 10, m = 1, n = 10^5$ で異なる rp の再構築エラー
Fig. 11　Errors for $c = 10, m = 1, n = 10^5$ with varying rp.

レコード数が 10^4 個の場合の結果と比較して，再構築エラーは全体的に低くなっている．さらに，従来手法と提案手法との再構築エラーの差も小さくなっている．図 12 では，二つの手法による再構築エラーはほとんど同じである．これは，レコー

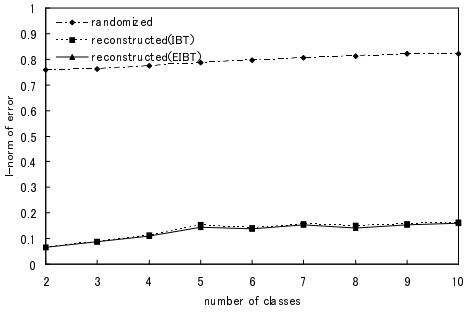

図 12　$m = 1, n = 10^5, rp = 0.2$ で異なる c の再構築エラー
Fig. 12　Errors for $m = 1, n = 10^5, rp = 0.2$ with varying c.

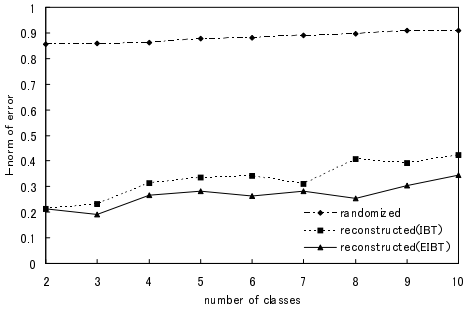

図 13　$m = 1, n = 10^5, rp = 0.1$ で異なる c の再構築エラー
Fig. 13　Errors for $m = 1, n = 10^5, rp = 0.1$ with varying c.

ド数の増加に伴い，目的属性の各属性値を独立に処理することの影響が小さくなったと考えることができる．一方，維持確率が 0.1 の結果である図 13 では，従来手法よりも提案手法の再構築エラーの方が小さい．

図 14 および図 15 では，5.3 の環境において目的属性が 10 値のテーブルのレコード数を増加させた場合の結果を示している． 図 10〜図 13 と同様，レコード数が 10^4 個の場合と比較して，再構築エラーは全体的に小さくなっている．そして，図 7, 図 9 と比べて，従来手法と提案手法の再構築エラーの差は小さくなってはいるが，提案手法の再構築エラーの方が小さい．

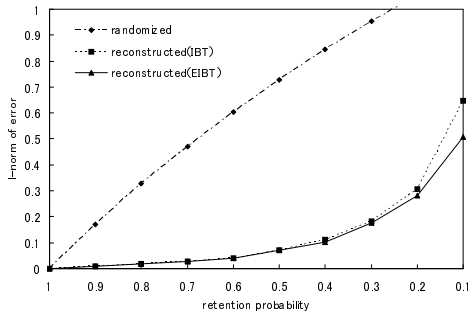

図 14　$c = 10, m = 2, n = 10^5$ で異なる rp の再構築エラー
Fig. 14　Errors for $c = 10, m = 2, n = 10^5$ with varying rp.

6. おわりに

本稿では，維持 - 置換摂動によって摂動されたテーブルから，目的属性が 3 値以上の決定木を構築するために必要なカウント演算結果を再構築する手法を提案した．従来手法を用いて目的属性が 3 値以上のカウント演算結果を摂動されたテーブルから再構築する場合，目的属性の各属性値で独立に再構築処理を行う必要があった．そこで，本稿では目的属性の各属性値を一括して再構築する手法を提案した．そして，提案手法がテーブルを摂動する割合が高い場合の再構築エラーを従来手法よりも低減させていることを実験により示した．

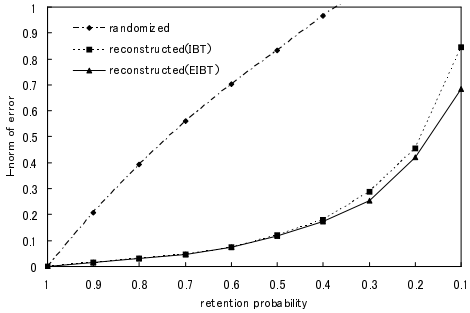

図 15　$c = 10, m = 3, n = 10^5$ で異なる rp の再構築エラー
Fig. 15　Errors for $c = 10, m = 3, n = 10^5$ with varying rp.

謝　辞

本研究の一部は，科学研究費補助金基盤研究 (C)(18500073) により行なわれた．

文　献

[1] R. Agrawal and R. Srikant: “Privacy-Preserving Data Mining.”, SIGMOD Conference (Eds. by W. Chen, J. F. Naughton and P. A. Bernstein), ACM, pp. 439–450 (2000).
[2] Y. Lindell and B. Pinkas: “Privacy Preserving Data Mining.”, CRYPTO (Ed. by M. Bellare), Vol. 1880 of Lecture Notes in Computer Science, Springer, pp. 36–54 (2000).
[3] A. V. Evfimievski, J. Gehrke and R. Srikant: “Limiting privacy breaches in privacy preserving data mining.”, PODS, ACM, pp. 211–222 (2003).
[4] R. Agrawal, R. Srikant and D. Thomas: “Privacy Preserving OLAP.”, SIGMOD Conference (Ed. by F. Özcan), ACM, pp. 251–262 (2005).
[5] A. J. Broder: “Data Mining, the Internet, and Privacy.”, WEBKDD (Eds. by B. M. Masand and M. Spiliopoulou), Vol. 1836 of Lecture Notes in Computer Science, Springer, pp. 56–73 (1999).
[6] N. R. Adam and J. C. Wortmann: “Security-Control Methods for Statistical Databases: A Comparative Study.”, ACM Comput. Surv., **21**, 4, pp. 515–556 (1989).
[7] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone: “Classification and Regression Trees.”, Wadsworth (1984).
[8] J. Han and M. Kamber: “Data Mining: Concepts and Techniques”, Morgan Kaufmann (2000).
[9] 福田剛志, 森本康彦, 徳山豪 : “データサイエンスシリーズ 3 データマイニング”, 共立出版 (2001).
[10] 福田剛志, 森本康彦, 徳山豪 : “多値属性を用いた最適なデータセグメンテーションを生成するアルゴリズム”, 電子情報通信学会技術研究報告, **98**, 316, pp. 83–91 (1998).
[11] Y. Morimoto, T. Fukuda, H. Matsuzawa, T. Tokuyama and K. Yoda: “Algorithms for Mining Association Rules for Binary Segmentations of Huge Categorical Databases.”, VLDB, pp. 380–391 (1998).