ノイズ入りデータからの頻出アイテム集合の推定

成田　和世†　　北川　博之††

† 筑波大学理工学研究科 〒 305-8573 茨城県つくば市天王台 1-1-1
†† 筑波大学システム情報工学研究科
†† 筑波大学計算科学研究センター
E-mail: †narita@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

あらまし　情報技術の発達に伴い，ディジタル情報は増加，多様化の一途を辿っている．それに伴い，巨大なデータから隠れた特徴やパターンを体系的に発見するデータマイニングは，ますます重要となっている [3], [4], [6], [7]．しかし，実世界にあるデータは欠損値や誤った値などのノイズを含み，ダーティなものも多い．このような**ノイズ入りデータ**からマイニングされる情報は不正確なものとなってしまう．そこで，本研究ではデータに確率的にノイズが入るケースを想定して，ノイズ入りデータを確率的なモデルで表現する．同時に，そこからノイズのない真の状態のデータにおける**頻出アイテム集合** [2] を確率的な計算によって発見する手法を提案する．さらに，マイニングの高速性を上げるため **FP 木** [7], [12] を用いたアルゴリズムを提示する．

キーワード　データマイニング，知識発見，知識処理，ノイズ入りデータモデル，FP 木

Eextraction of Frequent Itemsets from Noisy Data

Kazuyo NARITA† and Hiroyuki KITAGAWA††

† Science and Engineering，University of Tsukuba
Tennodai1-1-1，Tsukuba-shi，Ibaraki，305-8573 Japan
†† Graduate School of Systems and Information Engineering，University of Tsukuba
†† Center for Conputational Sciences，University of Tsukuba
E-mail: †narita@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

Abstract　As we face huge amounts of varied information, data mining, which helps us discover hidden features or rules from voluminous data systematically, has become more important [3], [4], [6], [7]. However, much data in real world is dirty, including noises such as missing values or irrelevant values. The information mined from such *noisy data* becomes incorrect. Whereat, assuming on the case where noises are mixed to data statistically, we represent noisy data as a probabilitic model. We also propose the way to estimate *frequent itemsets* [2] on the noiseless data, by probabilistic calculation using the noisy one. Besides, an algorithm using *FP-tree* [7], [12] is proposed in order to mine them efficiently.

Key words　Data Mining, Knowledge Discovery, Knowledge Management, Noisy Data, FP-tree

1. はじめに

情報技術の発達に伴い，ディジタル情報は増加，多様化の一途を辿っている．それに伴い，巨大なデータから隠れた特徴やパターンを体系的に発見するデータマイニングは，ますます重要となっている [3], [4], [6], [7]．しかし，実世界にあるデータは様々な要因から，欠損値や誤った値などのノイズを含むものも多い．このような**ノイズ入りデータ**（*noisy data*）からマイニングされる情報は不正確なものとなってしまう．そこで，本研究ではデータに確率的にノイズが入るケースを想定して，ノイズ入りデータを確率的なモデルで表現し，そこからノイズのない真の状態のデータにおける**頻出アイテム集合**（*frequent itemsets*）[2] を，確率的な計算によって推定する手法を提案する．同時に，推定を行って頻出アイテム集合をマイニングするアルゴリズムを提示する．本研究の特徴およびポイントは以下の通りである．

(1) 柔軟なノイズ入りトランザクションのモデル化

我々はノイズが確率的に混入したトランザクションデータか

ら，ノイズがない本来の状態のデータベースにおける頻出アイテム集合を体系的にマイニングすることを目的として，ノイズ入りトランザクションのモデル化を行う．本データモデルは [10] のアイデアに基づいている．[10] はプライバシーを考慮に入れたデータマイニング（*Privacy Preserving Data Mining, PPDM*）[5], [11] の一つである．個人情報を隠すためにデータに故意にノイズを入れ，そこからノイズを入れる前のデータにおけるアイテム集合のサポートを推定する手法を提案している．故意にデータへノイズを入れるオペレータを，ここでは *Randomization* と言う．

[10] では，トランザクションデータベース DB と，DB に出現する全アイテムの集合 I が与えられたとき，DB 中の各トランザクション t_i に対応するノイズ入りトランザクション t'_i を用意する．t_i と t'_i はアイテムの集合であり，t'_i は最初，空集合である．各アイテム $a \in I$ に対して，$a \in t_i$ ならば確率 p で a を t'_i に加え，$a \notin t_i$ ならば確率 $1-p$ で a を t'_i に加える．この Randomization に使用されるパラメタは p のみである．言い換えれば，t'_i には，t_i が持っていたアイテムが出現しない場合と t_i が持っていなかったアイテムが出現する場合があるが，どちらも同一の確率 p にのみ依存する．これは，ノイズ入りデータの表現としては極めて制約が強い．そこで我々は [10] の Randomization を，ノイズ入りデータの特徴に則して 2 つのパラメタを用いることで拡張し，データモデルを考案する．すなわち，t_i に存在するアイテムが失われる確率と，t_i にないアイテムが出現する確率を別のパラメタとして定式化する．これにより，より柔軟にノイズ入りトランザクションをモデル化することが可能である．

（2）ノイズ入りトランザクションに対する FP 木を用いた幅優先アルゴリズム

従来のノイズを考慮しない頻出アイテム集合マイニングと異なり，本研究では頻出アイテム集合を発見するために推定を行う．推定の際にはその部分集合のノイズ入りデータにおける出現回数が必要となる．そのため，推定によって頻出アイテム集合をマイニングするには，深さ優先のマイニングアプローチよりも幅優先アプローチのほうが適している．そこで，本稿ではある候補アイテム集合に対して，その部分集合の出現回数を用いて候補アイテム集合が頻出アイテム集合であるかどうか推定する幅優先のマイニングアルゴリズムを提案する．提案アルゴリズムでは，より高速なマイニング処理を実現するために，ノイズ入りデータのスキャニングにコンパクトなデータ木構造で知られる **FP 木**（*FP-tree*）[7], [12] を用いる．頻出アイテム集合を推定するために FP 木をどのように探索するか，部分集合の出現回数をどのように扱うかが，本提案アルゴリズムのポイントとなる．

表 1　トランザクションデータベース

TID	トランザクション
101	a, c, d, e, f
102	a, b, c, e
103	b, d, f
104	a, b, c, f
105	a, c, f

（3）実験的有用性の検証

本稿では，本研究の有効性を，実験によって検証する．ノイズ入りデータから推定によって得られる頻出アイテム集合がどのくらい正確であるか，また，FP 木を用いた提案アルゴリズムの処理が，FP 木を用いない場合に比べてどの程度速いかを実験によって検証する．

以降，本稿の構成は次のようである．2. で我々が提案するノイズ入りデータモデルについて説明する．3. では，そこから頻出アイテム集合をいかにして推定するのかを述べる．4. でマイニングアルゴリズムを提示し，5. で提案手法による推定の精度や，性能評価に関する実験結果を述べる．6. で本研究に関連する諸研究について言及し，7. でまとめる．

2. ノイズ入りデータモデル

この節では我々が提案するノイズ入りデータのモデルについて説明する．

表 1 はトランザクションデータベースの例である．TID はトランザクション一個一個に一意に割り当てられた ID であり，トランザクションはアイテムの集合である．先述のとおり，我々が提案するデータモデルは表 1 のようなカテゴリ属性のみで構成されたトランザクションデータベースを想定しており，[10] のアイデアに基づいている．我々は [10] の Randomization を，2 つのパラメタ p，q を用いることで拡張し，ノイズ入りデータの特徴に則してデータモデルを考案する．

2.1 定　　義

ここで，本稿で用いる表現や記法を定義する．ノイズ入りデータベース DB^{noisy} に対して，ノイズのない真の状態のデータベースを DB^{true} と表す．どちらも各々トランザクションの集合を意味し，$|DB^{noisy}| = |DB^{true}| = N$ である．DB^{noisy}，DB^{true} における全アイテムの集合を I で表す．アイテム集合 $A = \{a_1, a_2, \ldots, a_n\}$ を略記して $a_1 a_2 \ldots a_n$ と書くこともある．A の出現回数は，A を含むトランザクションの数であり，$cnt(A)$ と書く．サポートは $sup(A) = cnt(A)/N$ である．表 1 では，集合 abc を含むトランザクションは TID が 102 と 104 のものであり，$cnt(abc) = 2$，$sup(abc) = 2/5$ である．

2.2 確率的ノイズ入りデータモデル

ノイズはデータベース中の各アイテムに独立に作用することを想定する．データにノイズが発生するケースは，次の 2 種類に大別することが可能である．

(1) あるトランザクション t に本来なら含まれるはずのアイテムが t から欠落する．
(2) 本来なら t に含まれないはずのアイテムが t に混入する．

一般に，(1)，(2) はそれぞれ別々の頻度で起こると考えられる．そこで我々のモデルでは，各アイテムに働くこのような 2 種類のノイズを 2 つのパラメタ p,q によって表す．ここで，p はトランザクション $t_i \in DB^{true}$ に出現するアイテムが，t_i の対応トランザクション $t'_i \in DB^{noisy}$ でもその状態を保つ確率であり，q は $t_i \in DB^{true}$ に出現しないアイテムが，対応トランザクション $t'_i \in DB^{noisy}$ でもその状態を保つ確率である．

［Definition 1］ **確率的ノイズ入りデータモデル**
任意のアイテム $a \in I$ は，各トランザクション $t_i \in DB^{true}$ とその対応トランザクション $t'_i \in DB^{noisy}$ に対して，

1. $a \in t_i$ ならば，確率 p で $a \in t'_i$ となる．
2. $a \notin t_i$ ならば，確率 q で $a \notin t'_i$ となる．

ただし $p \neq q \neq 0.5$ である．

[10] の Randomization は，本モデルで $p = q$ のときである．

3. 推定頻出アイテム集合の導出

3.1 推定サポートの計算

本節では，前節で述べたパラメタ p,q が与えられたとき，提案データモデルに基づくノイズ入りデータから，DB^{true} 上の頻出アイテム集合をマイニングする方法を述べる．我々は DB^{true} 上の頻出アイテム集合のサポートを DB^{noisy} 上の情報を用いて推定する．ここで，DB^{true} 上の頻出アイテム集合を**真の頻出アイテム集合**，推定によって得られる頻出アイテム集合を**推定頻出アイテム集合**と呼ぶ．以降，推定の手法を順を追って説明する．

データベース DB^{noisy} はパラメタ p, q によるノイズ入りデータである．このとき，1-アイテム集合 $\{a\}(a \in I)$ の DB^{noisy} における出現回数 $cnt'(a)$ と DB^{true} における出現回数 $cnt(a)$ の間には，確率的に次のような関係が成り立つ．

$$\begin{pmatrix} cnt(a) \\ N\text{–}cnt(a) \end{pmatrix} = \begin{pmatrix} p & 1\text{–}q \\ q & 1\text{–}p \end{pmatrix}^{-1} \cdot \begin{pmatrix} cnt'(a) \\ N\text{–}cnt'(a) \end{pmatrix}$$

この式より p,q,cnt' を用いて表現される $cnt(a)$ は $\{a\}$ の DB^{true} 上のサポートの推定値 $sup_{est}(a)$ である．$sup_{est}(a)$ を**推定サポート**，それに対して DB^{true} 上のサポートのことを**真のサポート**と呼ぶ．

以降，$N - cnt(a)$ で表されるような，アイテム集合 $\{a\}$ を含まないトランザクションの数を $cnt(\overline{a})$ と表す．$\overline{a}$ は，アイテム a が出現しないことを表す記号である．アイテム $a,b \in I$ に対して，$cnt(a\overline{b})$ は集合 $\{a,\overline{b}\}$ の出現回数，つまり a を含み b は含まないトランザクションの数である．表 1 では，集合 $\{a,\overline{b}\}$ を含むトランザクションは TID が 101 と 105 のものであるので，$cnt(a\overline{b}) = 2$ である．

次に，2-アイテム集合 $ab(a,b \in I)$ について考える．先ほどと同様に，出現回数 $cnt(ab)$, $cnt'(ab)$ の間には次式が成り立つ．

$$\begin{pmatrix} cnt(ab) \\ cnt(a\overline{b}) \\ cnt(\overline{a}b) \\ cnt(\overline{a}\overline{b}) \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} cnt'(ab) \\ cnt'(a\overline{b}) \\ cnt'(\overline{a}b) \\ cnt'(\overline{a}\overline{b}) \end{pmatrix}$$

ここで，M はパラメタ p,q から作られる**確率行列**である．

$$M = \begin{pmatrix} p^2 & p(1\text{–}q) & (1\text{–}q)p & (1\text{–}q)^2 \\ p(1\text{–}p) & pq & (1\text{–}q)(1\text{–}p) & (1\text{–}q)q \\ (1\text{–}p)p & (1\text{–}p)(1\text{–}q) & qp & q(1\text{–}q) \\ (1\text{–}p)^2 & (1\text{–}p)q & q(1\text{–}p) & q^2 \end{pmatrix}$$

一般に，n-アイテム集合 $A = \{a_1, a_2, \ldots, a_n\}$ の推定サポートを計算するには，集合 $a_1a_2 \ldots a_n$, $\overline{a_1}a_2 \ldots a_n$, $\cdots$, $\overline{a_1a_2} \ldots \overline{a_n}$ を含むトランザクションの数がそれぞれ必要である．集合 A の推定に必要なこのようなトランザクションを，以降から A の**部分一致トランザクション**と呼ぶ．集合 $a_1a_2 \ldots a_n$, $\overline{a_1}a_2 \ldots a_n$, $\cdots$, $\overline{a_1a_2} \ldots \overline{a_n}$ の出現回数を c_{2^n-1}, c_{2^n-2}, $\ldots$, c_0 とすると，DB^{true} におけるベクトル $C = (c_{2^n-1}, \ldots, c_0)^T$ は，DB^{noisy} におけるベクトル $C' = (c'_{2^n-1}, \ldots, c'_0)^T$ と対応する $2^n \times 2^n$ の確率行列 M（要素 m_{ij}）を用いて次式で表される．

$$C = M^{-1} \cdot C'$$

$$\text{ただし,} m_{ij} = p^{n_1} \cdot (1-p)^{n_2} \cdot q^{n_3} \cdot (1-q)^{n_4}$$

n_1, n_2, n_3, n_4 は，c_j に対応する A の部分一致トランザクション $trans_j$ と c'_i に対応する A の部分一致トランザクション $trans_i$ とで，

$$n_1 = |\{a \in A | a \in trans_j \in DB^{noisy}, a \in trans_i \in DB^{true}\}|$$
$$n_2 = |\{a \in A | a \notin trans_j \in DB^{noisy}, a \in trans_i \in DB^{true}\}|$$
$$n_3 = |\{a \in A | a \notin trans_j \in DB^{noisy}, a \notin trans_i \in DB^{true}\}|$$
$$n_4 = |\{a \in A | a \in trans_j \in DB^{noisy}, a \notin trans_i \in DB^{true}\}|$$

である．結局，M の逆行列を L，その各要素を l_{ij} とすると，$\{a_1, a_2, \ldots, a_n\}$ の推定サポート sup_{est} は，

$$sup_{est} = \frac{1}{N} \sum_{k=0}^{2^n-1} c'_{2^n-1-k} \cdot l_{0k} \qquad (1)$$

となる．

3.2 演算コストの削減

式 1 は，マイニングするアイテム集合のサイズが大きくなるほど推定サポートの算出コストが指数関数的に増えることを意味する．

今，集合 A のある部分一致トランザクション $trans$ に対して，

$$e(trans) = |\{a \in A | a \in trans\}|$$

となる e を，A の部分一致トランザクション $trans$ における**存**

在数と呼ぶ．確率行列 M の要素は対称的な分布を取っており，その性質は逆行列 L にも反映されるため次式が成り立つ．

$$e(trans_{2^n-i}) = e(trans_{2^n-j}) \quad \Longrightarrow \quad l_{0i} = l_{0j}$$

n-アイテム集合の存在数は 0 から n の $n+1$ 通りなので，推定サポートの計算に必要な行列要素 l_{0k} の異なる値の数は $n+1$ である．存在数が e であるような全ての部分一致トランザクションの数を cnt'_e，行列要素 l_{0k} の異なる $n+1$ 個の値のうち，cnt'_e に対応するものを $coef_e$ とすると，n-アイテム集合の推定サポート sup_{est} は次式となる．

$$sup_{est} = \frac{1}{N}\sum_{k=0}^{n} cnt'_k \cdot coef_k$$

すなわち，ある n-アイテム集合のサポートを推定する際の演算コストは，$O(2^n)$ から $O(n+1)$ に削減可能である．

3.3 幅優先 vs 深さ優先

ここでは，本手法が幅優先型と深さ優先型のどちらのマイニングアプローチが適しているかを考える．

幅優先アプローチは，アプリオリアルゴリズム [3] に代表されるように，同サイズの頻出アイテム集合を全て発見した後，その次に大きいサイズの頻出アイテム集合をマイニングするアプローチである．深さ優先アプローチは，あるアイテムに着目し，そのアイテムを含むアイテム集合を，集合サイズを大きくしていきながら見つけていく．そのアイテムに関する頻出アイテム集合が全て発見し終わったら，今度は別のアイテムに着目するといったアプローチである．サイズの大きい頻出アイテム集合を発見する場合は，深さ優先のアプローチのほうが効率的であると言われている．

今，n-アイテム集合 $A = \{a_1, a_2, \ldots, a_n\}$ の部分集合 $A_j \subseteqq A$ と，その出現回数 $cnt'(A_j)$ に対して，

$$subsets\_cnt_k = \sum_{j,|A_j|=k} cnt'(A_j)$$

であるとき，A の存在数が $e(0 < e < n)$ の部分一致トランザクションの数 cnt'_e は式 2 となる．

$$cnt'_e = subsets\_cnt_e - \sum_{i=1}^{n-e} {}_{e+i}C_e \cdot cnt'_{e+i} \qquad (2)$$

[Proof 1]　$subsets\_cnt_e$ は $|A \cap trans| \geqq e$ となるようなトランザクション $trans$ の数である．今，$trans$ が，

$$|A \cap trans| = e + i \quad (i > 0)$$

であるとき，$|A_j| = e$ であるような $A_j \subset A$ のうち，$trans$ を数え上げるような A_j は ${}_{e+i}C_e$ 種類である．また，$|trans| = cnt'_{e+i}$ であるので，$cnt'_e = subsets\_cnt_e - \sum_{i=1}^{n-e} {}_{e+i}C_e \cdot cnt'_{e+i}$ となる．∎

式 2 より，$cnt'_e(0 < e < n)$ は，A の部分集合の出現回数から計算できる．また，$cnt'_n = cnt'(A)$，$cnt'_0 = N - \sum_{k=1}^{n} cnt'_k$ である．

アプリオリアルゴリズム [3] に代表される幅優先探索のアプローチでは，サイズ $n-1$ の頻出アイテム集合がマイニングし終わった時点で，サイズが $n-1$ 以下の頻出アイテム集合の出現回数は全て数え上げられている．よって，幅優先のアプローチでは，n-アイテム集合 A のサポートを推定する際，A の部分集合の出現回数は全て既知であるので，$cnt'_e(0 \leqq e \leqq n)$ はすぐに得られる．それに対して，深さ優先のアプローチでは，サイズ n のアイテム集合 A の推定サポートを計算するとき，A の部分集合に関するマイニングが終わっているとは限らず，部分集合の出現回数が未知である場合がある．結論として，本推定手法を用いたマイニングには，深さ優先よりも幅優先のアプローチが適している．

4. FP 木を用いたアルゴリズム

本節では，推定頻出アイテム集合のマイニングアルゴリズムについて説明する．前節で述べたように，本手法はその性質上，幅優先のアプローチが適切である．そこで，我々は最もポピュラな幅優先アルゴリズムであるアプリオリアルゴリズム [3] を元に，マイニング手法を提案する．

4.1 候補集合の刈り取り

最初に問題となるのは，本研究における推定サポートが厳密には**アプリオリプロパティ**を満たさないことである．アプリオリプロパティは，アイテム集合 A のサポートは，その真部分集合 $A_j \subset A$ のサポートを上回ることがないという，アイテム集合のサポートに関する性質である．既存の主要なマイニングアルゴリズムは，アプリオリプロパティを用いて数多くある候補アイテム集合を効率的に刈り取っている．幅優先探索のアプローチで生成される候補アイテム集合の数は膨大になりやすいことから，本手法でもなんらかの刈り取りは欠かせない．そこで，本研究ではアプリオリプロパティが成り立つことを仮定して刈り取りを行う．なお，この仮定に基づく本アルゴリズムの精度については 5. 節で検討する．

別の問題として，推定サポートを用いた頻出アイテム集合か否かの判定がある．すなわち，推定サポートが最小サポート以上であれば，頻出アイテム集合と判定して良いか否かという問題である．そこで，我々は [5] で用いられる対策を本手法に適用する．[5] は [10] と同じくプライバシーを考慮に入れたデータマイニングの研究である．彼らは推定サポート sup_{est} の標準偏差 σ を導出し，最小サポート min_sup に対して下記の判定を行っている．

$$sup_{est} \pm \sigma \geqq min\_sup \Rightarrow A \text{ は頻出アイテム集合である}$$

本手法でもこの判定を用いる．標準偏差は次式で求める [5].

表 2　順序づけされたアイテムリスト

アイテムリスト
a, c, f
a, c, b
f, b
a, c, f, b
a, c, f

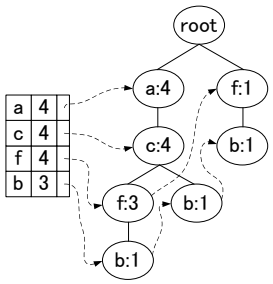

図 1　FP　　木

$$\sigma^2 = \frac{1}{N} \cdot \sum_{k=0}^{n} cnt_k \cdot (coef_k{}^2 - coef_k)$$

[5] では，真の頻出アイテム集合をマイニングし損ねることが，本当は頻出でないアイテム集合をマイニングするよりも重大なエラーと考え，$sup_{est} + \sigma$ を判定に用いている．しかし，本研究では実験的に $sup_{est} - \sigma$ のほうがわずかに高精度の結果を得るので，本稿の実験では判定に $sup_{est} - \sigma$ を用いる．

4.2　FP 木を用いたマイニング手法

本研究ではディスク上のデータベースを FP 木 [7], [12] に変換してメモリ上に置き，繰り返されるデータのスキャニングは FP 木に対して行う．FP 木は [7] で提案された木構造である．データベース中の各トランザクションに出現するアイテムの並びの中で，似たパターンを持つ接頭辞を一つのパスとして表現する．

図 1 は，最小サポートを 3 とした場合に，表 1 から構築される FP 木である．まず，表 1 を一度スキャンし，各アイテムの出現回数を得る．次に，各トランザクション中のアイテムを頻度の大きい順にソートして，表 2 のようなリストを作る．図 1 の木は表 2 から作られており，表 2 の各リストが，FP 木の各パスに対応しているのが分かる．リスト間で接頭辞が一致するものは，FP 木ではマージされて一つのパスとして表現される．FP 木のルート以外のノードは，アイテムの ID を表すラベル，カウント，同一ラベルを持つ次のノードを指すポインタを持つ．カウントの初期値は 1 で，接頭辞がマージされた分だけ増加する．同一ラベルを持つ最初のノードへのポインタは**ヘッダテーブル**と呼ばれる表に登録される．ヘッダテーブルには他に，アイテムの ID と出現回数が登録される．

このようにしてメモリ上に構築される FP 木は元のデータよりコンパクトになる．一方で，ディスクへのアクセスは時間が掛かり，処理速度のボトルネックになりがちである．そこで，本研究ではアプリオリライクな幅優先探索を，FP 木を用いて次のように行う．

今，サイズ n のアイテム集合をマイニングする．まずヘッダテーブルにアクセスし，アイテム $item_i$ をラベルに持つ最初のノード nd を得る．nd とその先祖ノードから $item_i$ を末尾に持つサイズ n の候補アイテム集合 $candidate$ を生成する．もし $candidate$ が既に作成されていたら，$candidate$ の現在までの出現回数に nd のカウントを足し合わせる．$item_i$ をラベルに持つ次のノードを nd から辿り，同じことを繰り返す．ヘッダテーブルに登録されている全てのアイテムについて処理が終了したら，各 $candidate$ の部分集合の出現回数を用いて推定サポートを計算し，$sup_{est} - \sigma \geqq min\_sup$ であるものを推定頻出アイテム集合とする．推定頻出アイテム集合となった $candidate$ の出現回数は，より大きいサイズの候補アイテム集合の推定サポートを計算するときに使うかも知れないので保存する．n-推定頻出アイテム集合を全て発見したら，マイニングの対象となるアイテム集合のサイズを $n+1$ とし，以下，同様に繰り返す．

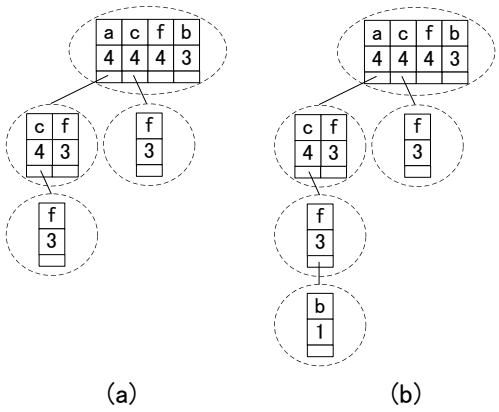

図 2　カウント木

4.3　カウント木

本手法では発見した推定頻出アイテム集合のノイズ入りデータ上での出現回数が，その真超集合の推定サポートを計算する際に必要なので，出現回数を保存するための木構造を用意する．この木を**カウント木**と呼ぶ．カウント木は 1 つのノードに 3 つの配列を持つ．1 つ目の配列は **ID 配列**で，アイテムの ID が入る．このとき，ID は一定の順序でソートされている．ここではアイテムの出現回数の大きい順にソートされるものとする．2 つ目の配列は**カウント配列**で，推定頻出アイテム集合の出現回数が入る．3 つ目の配列は**ポインタ配列**で，子ノードを指すポインタが入る．図 2 の (a) は表 1 を最小サポート 3 で，3-頻出アイテム集合までを幅優先でマイニングしたときに構築されるカウント木である．カウント木のルートノードの深さを 1 とすると，深さ n にあるノードには $n-$ アイテム集合の出現回数が保存される．図 2(a) 中，深さ 1 のノードが持つ配列の先頭には，集合 $\{a\}$ の出現回数が 4 である情報が入り，そこから辿れる深さ 2 の子ノードが持つ配列の先頭は，集合 $\{a, c\}$ の出現回数が 4 であることを示す．

今，サイズが 4 の候補アイテム集合の数え上げを FP 木から行う．ヘッダテーブルから b をラベルに持つ最初のノード nd に行き，その先祖ノードのラベルと b を組み合わせて 4-候補アイテム集合を生成する．ここでは $\{a, c, f, b\}$ のみが生成される．このとき，カウント木のルートノードの ID 配列から a を探し，その子ノードに行き，その配列から c を探し，その子ノードに

表 3　提案アルゴリズム

Input: ノイズ入りデータベース DB^{noisy},
　　パラメタ p, q, 最小サポート min_sup
Output: 頻出アイテム集合の完全集合 FI

```
1  DB^noisy から FP 木 fpt とヘッダテーブル ht を構築;
2  ht からカウント木 ct を初期化;
3  k=1;
4  while(サイズ k の集合が ct に存在) {
5    foreach(entry_i ∈ ht) {
6      node = pointer_i ∈ entry_i;
7      while(node ≠null) {
8        node を上に辿り，サイズ k + 1 の
         候補アイテム集合 candidate を生成;
9        if(ct に candidate が登録済み)
10         candidate の出現回数に
           node.count を足し合わせる;
11       else
11         node.count を初期値として ct に
           candidate を加える;
12     }
13   }
14   foreach(サイズ k の集合 set ∈ ct) {
15     set の全ての部分一致トランザクションの数 cnt を
       ct から計算;
16     cnt から推定サポート sup と標準偏差 σ を計算;
17     if(sup ± σ < min_sup)
18       set の情報を ct から削除;
19   }
20   k=k+1;
21 }
22 ct から FI を生成;
```

行く．深さ 3 の子ノードの ID 配列から f を探し，ポインタ配列の対応する位置に子ノードを作り，子ノードの ID 配列に b を登録する．カウント配列には nd のカウント 1 を入れる．nd から b をラベルに持つ次のノードを辿り，同様の処理を行う．その他のアイテム f, c, a についても処理を行い，最終的に 4-候補アイテム集合の数え上げが終わった時点のカウント木が図 2(b) に示されている．

我々が提案するマイニングアルゴリズムを，表 3 にまとめる．入力としてノイズ入りデータベース，事前に得られたパラメタ p, q（p, q を得る方法については本稿では言及しない），最小サポートを与え，推定頻出アイテム集合の集合を出力する．

5. 実　　験

実験では本手法の推定精度と処理時間の検証を行った．全ての実装は我々が行った．実験時の環境は表 4 の通りである．

推定精度

まず，IBM の人工データ生成器 [1] で T10.I4.D1M.N500 の人工データを生成した．すなわち，トランザクションの平均サイズが 10，アイテム集合の最大可能サイズの平均が 4，トランザクションの総数が 100 万，アイテムの種類数が 500 である．

表 4　実験環境

OS	Linux
メモリ	6GB
CPU	Xeon 3GHz
言語	java1.5.0

このデータをノイズのない真のデータベース Data_true とした．パラメタセット $(p, q) = (0.9, 0.999)$，$(p, q) = (0.8, 0.999)$，$(p, q) = (0.9, 0.99)$ を与えて，人為的に 3 つのノイズ入りデータを作った．

最初に，Data_true をノイズを考慮しない従来手法でマイニングし，「正解」となる真の頻出アイテム集合を得た．次に本手法で 3 つのノイズ入りデータをそれぞれマイニングし，得られた推定頻出アイテム集合を真の頻出アイテム集合と比較した．最小サポートは 0.1%である．

それぞれの比較結果が，表 5，表 6，表 7 に，集合のサイズごとに示されている．Est が本手法で発見した推定頻出アイテム集合の数，True が真の頻出アイテム集合の数を表している．R は Recall，P は Precision，F は F-measure をそれぞれ意味し，次式で求めた．

$$R = \frac{\text{Est} - \text{false positive}}{\text{True}} \times 100$$

$$P = \frac{\text{Est} - \text{false positive}}{\text{Est}} \times 100$$

$$F = \frac{2 * R * P}{R + P} \times 100$$

ここで，false positive は推定頻出アイテム集合であるが，真の頻出アイテム集合ではない集合の数を表す．

表 5，表 6，表 7 から，どの集合サイズでもかなりの高精度で推定頻出アイテム集合が得られたことが分かる．さらに，3 つのノイズ入りデータベースをそれぞれノイズを考慮しない従来手法でマイニングし，得られた集合も検証に加えた．その結果がそれぞれ表 8，表 9，表 10 に表されている．Non Est がノイズ入りデータを従来手法でマイニングして得られた集合の数である．

表 8 の再現率を見ると，パラメタ p によるアイテムが消失するノイズの影響が顕著に表れていることが分かる．適合率では，集合サイズ 5 までは高い値であるが，それより大きな集合サイズについては 0 となる．これは，パラメタ p によるノイズの影響で，大きなアイテム集合が拾いきれなくなったためである．

表 9 は表 8 に比べてパラメタ p によるノイズの程度を大きくしたノイズ入りデータに対する結果である．集合サイズが大きくなるにつれて，再現率が表 8 よりも大きく減少するのを確認した．

表 9 に対して表 10 は，表 8 に比べてパラメタ q による，アイテムが混入するノイズの程度を大きくしたノイズ入りデータに対する結果である．集合サイズ 2 の適合率の値はパラメタ q

表 5　提案手法の推定能力 ($p = 0.9, q = 0.999$)

Set	# of Itemsets				
Size	True	Est	R (%)	P (%)	F (%)
1	468	468	100.00	100.00	100.00
2	15548	15659	99.34	98.64	98.99
3	2168	2185	94.23	96.48	96.85
4	1142	1154	99.21	98.18	98.69
5	661	752	99.24	87.23	92.85
6	389	386	99.23	100.00	99.61
7	174	175	100.00	99.43	99.71
8	56	56	100.00	100.00	100.00
9	11	11	100.00	100.00	100.00
10	1	1	100.00	100.00	100.00

表 6　提案手法の推定能力 ($p = 0.8, q = 0.999$)

Set	# of Itemsets				
Size	True	Est	R (%)	P (%)	F (%)
1	468	468	99.79	99.79	99.79
2	15548	15646	99.09	98.47	98.78
3	2168	2211	95.85	93.98	94.91
4	1142	1228	97.46	90.64	93.92
5	661	746	99.39	88.07	93.39
6	389	402	98.71	95.52	97.09
7	174	176	100.00	98.86	99.43
8	56	56	100.00	100.00	100.00
9	11	11	100.00	100.00	100.00
10	1	1	100.00	100.00	100.00

表 7　提案手法の推定能力 ($p = 0.9, q = 0.99$)

Set	# of Itemsets				
Size	True	Est	R (%)	P (%)	F (%)
1	468	466	99.36	99.79	99.57
2	15548	15623	98.39	97.91	98.15
3	2168	2202	97.42	95.91	96.66
4	1142	1172	97.99	95.48	96.72
5	661	676	99.09	96.89	97.98
6	389	389	100.00	100.00	100.00
7	174	175	100.00	99.43	99.71
8	56	56	100.00	100.00	100.00
9	11	11	100.00	100.00	100.00
10	1	1	100.00	100.00	100.00

のノイズの影響で，集合サイズ 1 の適合率の値と比較して激減した．集合サイズ 3, 4, 5 において適合率が持ち直した理由として，q によるアイテムが混入するノイズは，大きなアイテム集合になるほどマイニング結果に与える影響が小さくなるためと考えられる．

総合的に見て，ノイズ入りデータに対する従来手法で得られるマイニング結果は，Recall, Precision, F-measure の全てでノイズの影響が顕著に現れており，本研究の有効性が示された．

表 8　ノイズがマイニング精度に及ぼす影響 ($p = 0.9, q = 0.999$)

Set	# of Itemsets				
Size	True	Non Est	R (%)	P (%)	F (%)
1	468	494	100.00	94.74	97.39
2	15548	12939	83.22	100.00	90.84
3	2168	454	20.94	100.00	34.63
4	1142	64	5.6	100.00	10.61
5	661	7	1.06	100.00	2.1
6	389	0	0.00	0.00	0.00
7	174	0	0.00	0.00	0.00
8	56	0	0.00	0.00	0.00
9	11	0	0.00	0.00	0.00
10	1	0	0.00	0.00	0.00

表 9　ノイズがマイニング精度に及ぼす影響 ($p = 0.8, q = 0.999$)

Set	# of Itemsets				
Size	True	Non Est	R (%)	P (%)	F (%)
1	468	494	100	94.74	97.3
2	15548	9167	58.96	100	74.18
3	2168	29	1.34	100	2.64
4	1142	1	0.09	100	0.17
5	661	0	0.00	0.00	0.00
6	389	0	0.00	0.00	0.00
7	174	0	0.00	0.00	0.00
8	56	0	0.00	0.00	0.00
9	11	0	0.00	0.00	0.00
10	1	0	0.00	0.00	0.00

表 10　ノイズがマイニング精度に及ぼす影響 ($p = 0.9, q = 0.99$)

Set	# of Itemsets				
Size	True	Non Est	R (%)	P (%)	F (%)
1	468	500	100.00	93.60	96.69
2	15548	30013	100.00	51.80	68.25
3	2168	666	30.72	100.00	47.00
4	1142	71	6.22	100.00	11.71
5	661	6	0.91	100.00	1.80
6	389	0	0.00	0.00	0.00
7	174	0	0.00	0.00	0.00
8	56	0	0.00	0.00	0.00
9	11	0	0.00	0.00	0.00
10	1	0	0.00	0.00	0.00

処 理 時 間

ディスク上のデータをアプリオリライクにマイニングする場合と，FP 木からマイニングする場合とで実行速度を比較した．ここでの実行速度は，-Xloggc オプションを使用し，得られたガーベッジコレクタ（GC）の情報に基づき，総実行時間から GC に掛かった時間を引いたものである．得られた結果を図 3 に示す．FP 木を用いたマイニングに掛かる時間には，FP 木をディスク上のデータベースから構築する時間も含む．最小サポートが小さくなるにつれ，FP 木をスキャニングする本手法の効率性が高くなっていることが分かる．

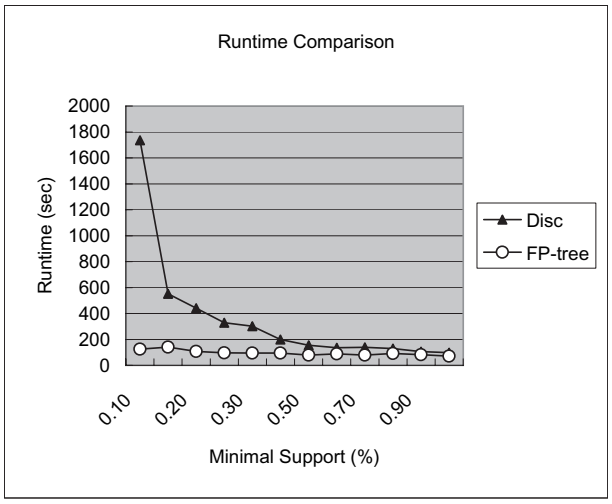

図 3　実行時間の比較

6. 関 連 研 究

本稿で提案したデータモデルはプライバシーを考慮に入れたデータマイニング手法 [10] を基にしているが, [10] の Randomization には一つのパラメタを使用しているのに対し，我々が提案するデータモデルは，ノイズ入りデータの特徴に則って二つのパラメタを用い，より柔軟にノイズの混入パターンをモデル化できる．

[5] もプライバシーを考慮に入れたデータマイニングの研究であるが，多数のパラメタを用いて [10] とは全く異なる Randomization を考案している．また，推定サポートが真のサポートに対して誤差を含むことを考慮に入れて，推定サポートの標準偏差を導出し，推定サポートを大きく見積もる方法を採っている．

[9] は，実世界情報の多くがダーティで，そのことがアイテム集合のサポートや相関ルールの確信度に少なからず影響を与え，マイニングされるほとんどの頻出アイテム集合や相関ルールが，しきい値に近い小さなサポートや確信度を持っていることに注目している．そこで，しきい値を与えて最小サポートよりある程度小さなサポートを持つアイテム集合をマイニングすることで，従来のマイニングでは見落としてしまうアイテム集合の中から興味深いパターンをマイニングしようと試みている. [8] は行列で表現されたノイズ入りデータに対して，ある程度のエラーを許容するという考え方に基づいて頻出アイテム集合のマイニングを行っている. [9] と [8] はどちらもデータにノイズが混入していることを考慮したデータマイニングの研究だが，本研究と異なり，確率的な推定計算を行わない．

7. お わ り に

本稿では，実世界に多く見られるノイズ入りデータに着目し，確率によってノイズが発生するデータに対して，直感的かつ汎用的なデータモデルを考案した．確率的な計算を行うことで，ノイズが入る前のサポート値をノイズ入りデータから計算する手法を述べた．また，高速性を狙い，FP 木を用いた幅優先のマイニングアルゴリズムを示した．実験では本手法が大きな集合に対しても高精度な結果を返すことを明らかにし，FP 木を用いることの速度面での有効性を示した．

提案手法で得られる推定頻出アイテム集合は，高精度ではあるが，本来ならマイニングされないはずの非頻出なアイテム集合もマイニングする．今後は，そのような誤りが混じる結果に対して一定の保証を与える方法等，本手法で得られる結果に対する考察を深める予定である．また，パラメタ p と q をノイズ入りデータから推定する方法や，ノイズ入りデータのより実用的なモデル化なども検討中である．

謝辞　本研究の一部は日本学術振興会科学研究費補助金基盤研究 (B)(#15300027) の助成による．

文　　献

[1] http://www.almaden.ibm.com/software/disciplines/iis/
[2] R. Agrawal, T. Imielinski and A. Swami, “Mining Association Rules between Sets of Items in Large Databases, ” Proc. ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington, D.C., USA, May, 1993.
[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, “Fast discovery of association rules, “ Advances in Knowledge Discovery and Data Mining 1996, pp. 307-328, 1996.
[4] C. Borgelt, “Recursion Pruning for the Apriori Algorithm, “ Proc. the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, vol. 126, Brighton, UK, Nov. 2004.
[5] A. V. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy preserving mining of association rules, “ Proc. 8th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 217-228, Edmonton, Canada, Jul. 2002.
[6] G. Grahane and J. Zhu, “Fast Algorithm for Frequent Itemset Mining Using FP-Trees, “ IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 10, pp. 1347-1362, Oct, 2005.
[7] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation, “ Proc. ACM SIGMOD International Conference on Management of Data, pp. 1-12, Dallas, USA, May 2000.
[8] J. Liu, S. Paulsen, W. Wand, A. Nobel, and J. Pris, “Mining approximate frequent itemsets from noisy data, “ Proc. 5th IEEE International Conference on Data Mining, pp. 721-724, Houston, USA, Nov. 2005.
[9] J. Pei, A. K. H. Thung, and J. Han, “Fault-tolerant frequent pattern mining: Problems and challenges, “ ACM SIGMOD Workshop on Research Issues in DMKD, Santa Barbara, USA, May 2001.
[10] S. Rizvi and J. R. Haritsa, “Maintaining data privacy in association rule mining, ” Proc. 28th VLDB Conference, pp. 682-693, Hong Kong, China, Aug. 2002.
[11] V. S. Verykos, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin and Y. Theodoridis, “State-of-the-art in Privacy Preserving Data Mining, ” SIGMOD Rec., vol. 33, no. 1, pp. 50-57, 2004.
[12] J. Wang, J. Han and J. Pei, “CLOSET+: searching for the best strategies for mining frequent closed itemsets, “ Proc. 9th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 236-245, Washington, D.C., USA, 2003.