DEWS2005 6C-o3

ノイズを含むデータに対するFP木を用いた相関ルールマイニング

成田　和世† 　北川　博之††

† 筑波大学理工学研究科　〒 303–8573 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　〒 303–8573 茨城県つくば市天王台 1–1–1
†† 筑波大学計算科学研究センター　〒 303–8573 茨城県つくば市天王台 1–1–1
E-mail: †narita@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

あらまし　ディジタル情報の巨大化，多様化が進むにつれて，巨大なデータから特徴や法則を発見するデータマイニングはますます重要視され，様々な研究がなされている．アイテム間の相関性を発見する相関ルールマイニングにおいては，近年，同値のサポートを持つ真超集合を持たない頻出アイテム集合（**頻出飽和アイテム集合**，*frequent closed itemsets*) を見つけ出すことが重要であるとして，注目されつつある [4], [5]．一方で，情報に対するプライバシーへの関心が高まっていることを背景に，プライバシーを考慮したマイニングに関する研究も進められている [2], [3]．本稿では，プライバシーを考慮に入れたデータからいかにして頻出飽和アイテム集合を求めるかを検討し，マイニングの手法を提案する．プライバシーを考慮に入れた相関ルールマイニングにおいて，頻出飽和アイテム集合を見つけ出す手法はこれまで提案されていない．本提案手法では，マイニングの効率化を図るため，コンパクトなデータ構造で知られる J.Han らの FP 木 [1] を用いる．
キーワード　相関ルールマイニング，FP 木，ノイズ入りデータ，飽和アイテム集合

FP-tree using mining of Association rules for distorted data

Kazuyo NARITA† and Hiroyuki KITAGAWA††

† Master's program of 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　Tennodai1–1–1，Tsukuba-shi，Ibaraki，305–8573 Japan
†† Center for Conputational Sciences，University of Tsukuba　Tennodai1–1–1，Tsukuba-shi，Ibaraki，305–8573 Japan
E-mail: †narita@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

Abstract　As we are facing with a huge amount of various, data mining, which discovers features or rules from large data, has become more important. In association rules mining, *frequent closed itemsets* mining is gaining much attraction recently, which mines only frequent itemsets having no proper superset. In addition, privacy preserving mining has become one of hot research issues due to increased interest in privacy. In this paper, we propose a novel algorithm to mine frequent closed itemsets under the constraint of privacy preservation. To the best of our knowledge, there are no proposal on mining frequent closed itemsets taking privacy preservation into consideration. The proposed algorithm is based on J. Han's FP-tree [1].
Key words　association rules, FP-tree, distorted data, closed itemsets

1. まえがき

ディジタル情報の巨大化，多様化が進むにつれて，巨大なデータから特徴や法則を発見するデータマイニングはますます重要視され，様々な研究がなされている．アイテム間の相関性を発見する相関ルールマイニングで近年注目されている**飽和アイテム集合**（*closed itemsets*）マイニングは，同値のサポートを持つ真超集合を持たない頻出アイテム集合を全てマイニングすることであり，これまで様々なアルゴリズムが提案されている [4], [5]．一方で，情報に対するプライバシーへの関心が高まっていることを背景に，プライバシーを考慮して故意にノイズを入れたデータに対するマイニングも研究されている [2], [3]．

本稿では，プライバシーを保護するため故意にノイズを入れたデータから，どのようにして頻出飽和アイテム集合を発見す

るかを検討する．プライバシーを考慮に入れた相関ルールマイニングにおいて，頻出飽和アイテム集合を見つけ出す手法はこれまで提案されていない．本研究の，プライバシー保護のためのノイズ入りデータの扱いは R.Agrawal らの研究 [2] に基づく．また，マイニングの効率化を図るため，コンパクトなデータ構造で知られる J.Han らの FP 木 [1] を用いる．

本稿の構成は次の通りである．まず，2. で本研究に関連するノイズ入りデータ，頻出飽和アイテム集合，FP 木とそれを用いたマイニングアルゴリズム CLOSET+についてそれぞれ説明し，3. で提案手法を述べる．4. で，今回行った実験の内容と，その結果について説明し，最後に 5. でまとめと今後の研究への課題を述べる．

2. 関 連 研 究

2.1 ノイズ入りデータとサポートの推定

R.Agrawal らは，プライバシー保護のためにデータにノイズを入れる手法として，次に述べる Randomization Operator を提案している [2].

［Definition 1］ ノイズのない本来のトランザクションデータベース DB に含まれるトランザクション t_i の大きさ m に対して，2 つのパラメタ $\rho_m \in (0,1)$，整数 $K_m > 0$ が与えられるとする．DB における全てのアイテムの集合を I としたとき，トランザクション t_i に対して以下の処理を行うことでノイズを含むトランザクション t'_i を得る操作を *cut-and-paste randomization* という．

1 $0 \leqq j \leqq K_m$ となる整数 j を一様ランダムに選択する．その際，$j > m$ ならば $j = m$ とする．

2 ノイズのないトランザクション t_i から一様ランダムに j 個のアイテムを選択し，それらのアイテムを含むトランザクション t'_i を得る．

3 選択した j 個のアイテム以外の各アイテム $a \in I$ を，各々確率 ρ_m で t'_i に加える．

定義 1 より，大きさ m のトランザクションから k 個のアイテムが選択される確率 $p_m[k]$ は次式で表される．

$$p_m[k] = \sum_{i=0}^{min(K_m,k)} {}_{m-i}C_{k-i} \cdot {\rho_m}^{k-i}(1-\rho_m)^{m-k} \cdot \cdot \begin{cases} 1-m/(K_m+1) & \text{if } i=m \text{ and } i<K_m \\ 1/(K_m+1) & \text{otherwise} \end{cases} \quad (1)$$

次に，上述の Randomization でノイズを入れたデータから，どのようにして本来のデータ上のサポートを推定するかを説明する．T を本来のデータベース DB に含まれる全てのトランザクションの集合，T' をノイズ入りデータベース DB' に含まれる全てのトランザクションの集合，T，T' の大きさを共に N とする．また，DB における全てのアイテムの集合を I，$A \subset I$ を濃度 $card(A)$ が k であるアイテム集合とする．さらに，簡単化のため，ここでは全ての $t \in T$ は濃度 m であるとする．このとき, [2] は A のサポート値を推定する方法について次のように説明している．

［Definition 2］ アイテム集合 A との共通集合の濃度が l，すなわち $card(A \cap t) = l$ であるトランザクション t の，全てのトランザクションに対する割合を A の共通集合濃度 l の**部分サポート**(*partial support*) と呼ぶ：

$$supp_l{}^T(A) = \frac{card(\{t \in T | card(A \cap t) = l\})}{N} \quad (2)$$

また，$card(A \cap t) = l$ かつ $card(t) = m$ であるトランザクション $t \in T$ が，Randomization によって A との共通集合の濃度が l'，すなわち $card(A \cap t') = l'$ であるトランザクション $t' \in T$' となる確率は,

$$p_k{}^m[l \to l'] = \mathbf{P}[card(t' \cap A) = l' | card(t \cap A) = l].$$

と表され，式 1 とパラメタで与えられる確率 ρ_m を用いて次のように表現できる．

$$p_k{}^m[l \to l'] = \sum_{j=0}^{m} p_m[j] \cdot \sum_{q=max\{0,j+l-m,l+l'-k\}}^{min=\{j,l,l'\}} \frac{{}_lC_q \cdot {}_{m-l}C_{j-q}}{{}_mC_j} \cdot \cdot {}_{k-l}C_{l'-q}{\rho_m}^{l'-q}(1-\rho_m)^{k-l-l'+q}. \quad (3)$$

［Remark 1］ cut-and-paste randomization は個々のトランザクションに各々独立に作用する Randomization である．濃度の異なるトランザクションに対しては，上式を各々の濃度ごとに計算する必要がある．

さらに，データベース DB における部分サポートを $s_l = supp_l{}^T(A)$，データベース DB' における部分サポートを $s'_l = supp_l^{T'}(A)$ と表したとき，ベクトル $\vec{s'} = (s'_0, s'_1, \cdots, s'_k)^t$ の期待値 $\mathbf{E}\vec{s'}$ はベクトル $\vec{s} = (s_0, s_1, \cdots, s_k)^t$ を用いて次式で表される．

$$\mathbf{E}\vec{s'} = P \cdot \vec{s}$$

ここで，P は各要素が $P_{l'l} = p_k{}^m[l \to l']$ の $(k+1) \times (k+1)$ 行列である．

P の逆行列を Q としたとき，

$$\vec{s} = \mathbf{E}Q \cdot \vec{s'} \quad (4)$$

であることから，不偏推定量 $\vec{s}_{est} = Q \cdot \vec{s'}$ が得られる. [2] はここで，$s_k = supp_k{}^T(A)$ であることから，$\vec{s}_{est}$ の k 番目の要素をアイテム集合 A のサポートの推定値 s_{est} として，次式で与えている：

$$s_{est} = \sum_{l'=0}^{k} s'_{l'} \cdot q[k \leftarrow l']; \quad \text{where } q[l \leftarrow l'] = Q_{ll'}$$

これを我々は**推定サポート**と呼ぶ.

以上から，ノイズ入りデータから任意のアイテム集合の部分サポートを全て求め，かつ確率 $P_{l'l}$ を計算することで，本来のデータにおけるサポートの推定値が得られる．

2.2 飽和アイテム集合マイニング

[4] は，データ内に存在する全ての頻出飽和アイテム集合を求めるため，**FP 木**(*FP-tree*) [1] を用いたマイニング手法

TID	itemsets	ordered list
101	A, C, D, F, H	A, D, F
102	D, E, F, J	D, E, F
103	A, B, E, G, I	A, B, E, G
104	A, B, D, G	A, B, D, G

図１　データベースとアイテムリスト

Fig. 1　Database and ordered frequent item list

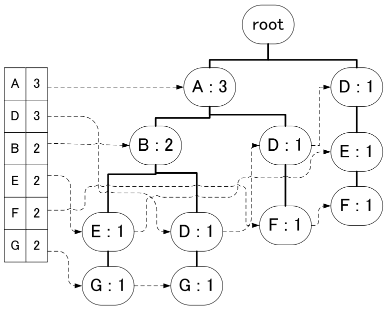

図２　FP　　木

Fig. 2　FP-tree

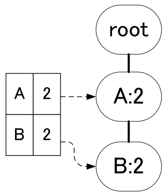

図３　G : 2 の FP 木

Fig. 3　G:2’s FP-tree

CLOSET+を提案している．本研究ではノイズを含むデータから頻出飽和アイテム集合をマイニングするのに，CLOSET+ライクな手法を提案する．本小節では CLOSET+について説明する．まず頻出飽和アイテム集合，FP 木について順番に述べたあと，CLOSET ＋の説明に入る．

［Definition 3］　サポートが $supp(A)$ であるアイテム集合 A が最小サポートを満たす頻出アイテム集合であり，かつ $supp(A) = supp(A')$ を満たす真超集合 $A' \supset A$ が存在しないとき，頻出アイテム集合 A を頻出飽和アイテム集合という．

Definition3 より，頻出飽和アイテム集合の全体集合は，全ての頻出アイテム集合の情報を完全に保持していながら頻出アイテム集合の全体集合よりもコンパクトであることは明らかであり，飽和アイテム集合マイニングの高い有効性が分かる．

次に FP 木について簡単に述べる．FP 木とは，データベース中の似たパターンをもつトランザクションをひとつのパスとしてマージすることでデータをコンパクトに圧縮するデータ木構造である [1].

図１の１，２列目はトランザクションデータベースである．３列目は最小サポートを 2 と設定した場合に，各トランザクションの最小サポート条件を満たすアイテムを，サポートの降順に左から並べ直したリストであり，これから FP 木を構築すると図２のようになる．

FP 木の構築手順を具体的に述べる．出現頻度の高い順に左からアイテムを並べ替え，最小サポート条件を満たさないアイテムを刈り取った各トランザクションを，左から右へスキャンする．最初のアイテムが FP 木の根ノードの子のラベルと一致すれば，その子ノードのカウントを１増やし，さらに下の子ノードと，トランザクションの次のアイテムを見る．子ノードのラベルとトランザクションのアイテムが初めて一致しないとき，トランザクションのアイテムをラベルとして持ち，かつ count の値が１であるノードを作り，新たな子ノードとして木に加える．また，FP 木は木構造と同時にヘッダテーブルと呼ばれる表を持つ．ヘッダテーブルの各行には，最初サポート条件を満たすアイテムと，それをラベルに持つ最初のノードへのポインタが入っており，そのノードは同じラベルを持つ次のノードを指すポインタを持っている．次のノードが存在しない場合は null を指す．

FP 木を用いた飽和アイテムマイニングのアルゴリズム CLOSET+ [4] は FP-growth [1] ライクな手法で，入力データベースが密であるか，疎であるかでふるまいを変えるハイブリッド型のアプローチである．本提案手法は，密データに対するふるまいに準じているので，ここでは密データに対するふるまいのみ説明する．密データに対する CLOSET+は，ヘッダテーブルから単一アイテムに着目して FP 木を辿り，そのアイテムについて再帰的に FP 木を作りながらより大きなアイテム集合をマイニングしていくという FP-growth ライクのアルゴリズムとなる．頻出アイテム集合マイニングとの相違点二点を明らかにしながら動作を述べる．

(1) まず，アイテム集合の刈り取りの技術が異なる．CLOSET+では次の３つの方法を採用している．(a)**item margin**：サポートが $supp(X)$ である頻出アイテム集合 X を含むトランザクションが必ずアイテム集合 Y も含み，かつ Y に $supp(X)$ と等しいサポートを持つ真超集合が存在しないなら，$X \cup Y$ を頻出飽和アイテム集合として採用する．(b)**sub-itemset pruning**：頻出アイテム集合 X が，既に発見された飽和アイテム集合 Y の部分集合で，かつ X と Y のサポートが等しいなら X を部分集合に持つ飽和アイテム集合は探す必要がない．(c)**item skipping**：再帰的に FP 木を作るとき，ヘッダテーブルのある単一アイテムのサポートが，異なるレベルの FP 木におけるヘッダテーブルの同一アイテムのサポートと等しいならば，グローバルな FP 木により近いレベルにある木のヘッダテーブルから，その単一アイテムの情報を刈り取ることができる．(c) は CLOSET+に固有の刈り取り方法である．

(2) 飽和アイテム集合マイニングでは，頻出アイテム集合を発見したらその飽和性を調べる必要がある．飽和性の有無は同値のサポートを持つすでに発見された頻出飽和アイテム集合との包含関係を調べることで分かる．CLOSET+はアプローチの性質から，新しく発見したアイテム集合が既に発見済みの頻出飽和アイテム集合の部分集合であるかどうかのみを調べるだけで飽和性の有無の確認が可能である．

例として，一連の流れを図 1 のデータベースを用いて説明する．以降から，任意のアイテム集合 $\{x_1, x_2, \cdots, x_n\}$ を，$x_1x_2\cdots x_n$ と略記し，集合 $x_1x_2\cdots x_n$ が k 個のトランザクションに出現している場合は，明示的に $x_1x_2\cdots x_n : k$ と表す．図 1 のデータベースから構築される FP 木は図 2 である．今，ヘッダテーブルの G のエントリから，G のノードを持つパスを辿ると，アイテム G は，アイテム A, B, E を一緒に持つトランザク

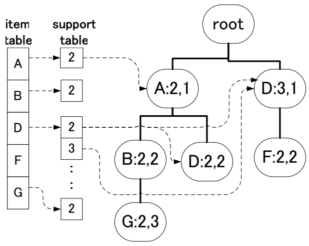

図 4　結　果　木
Fig. 4　result tree

ションと，アイテム A, B, D を一緒に持つトランザクションの２つに現れていることが分かる．よって，G と共に出現し，かつ最小サポート２を満たすアイテムは A, B である．ここで，アイテム集合 $G:2$ と共起するアイテムから FP 木を構築すると，図 3 となる．これを，データベースから最初に構築したグローバルな FP 木（グローバル FP 木）に対して，集合 $G:2$ の FP 木と言う．$G:2$ の FP 木のヘッダテーブルを辿ることで，再帰的なマイニングを行うが，ここで，アイテム A, B のサポートは G と同値であるため，item margin が適用され，頻出飽和アイテム集合 $ABG:2$ が発見される．また，sub-itemset pruning により，$AG:2$ および $BG:2$ を部分集合にもつ飽和アイテム集合を探す必要はない．さらに，$G:2$ の FP 木と，グローバル FP 木において，アイテム B のサポートは等しい．item skipping から，B についての飽和アイテム集合はこれ以上探す必要がないので，グローバル FP 木のヘッダテーブルから B のエントリを削除する．マイニングの流れは以上である．

密データに対する CLOSET+では，頻出飽和アイテム集合を発見すると結果木 (*result tree*) と呼ばれる FP 木に挿入していく．新たな頻出アイテム集合 X が発見されると，飽和性の有無を調べるために結果木をスキャンする．結果木のどのアイテム集合とも包含関係になければ X を木に加える．図 4 は，図 2 から頻出飽和アイテム集合 $ABG:2, DF:2, AD:2, D:3$ を，この順で発見した時の結果木である．結果木は頻出飽和アイテム集合の情報を保存する FP 木と，アイテムテーブルとサポートテーブルで構成される２次のテーブルからなる．FP 木の根以外のノードには，アイテム名と count の他に，FP 木におけるノードの深さの情報が入る（根の子供が深さ１である）．複数のアイテム集合がひとつのパスもしくは部分パスを共有する場合，count には，共有するアイテム集合の数ではなく，複数のアイテム集合の中で，最大のサポートの値が入る．１次元目のアイテムテーブルは頻出飽和アイテム集合の末尾のアイテムをエントリとして持ち，そのエントリは頻出飽和アイテム集合のサポートをエントリに持つサポートテーブルを指す．今，新たな頻出飽和アイテム集合の候補 $BD:2$ を発見し，飽和性を判定するために結果木を調べる．$BD:2$ の末尾にあるアイテム D から，アイテムテーブルを辿り，サポート２からサポートテーブルのエントリを辿る．エントリが指す２つのノードと、各先祖ノードを調べても，BD を部分集合とするパスは見つからないので，$BD:2$ は頻出飽和アイテム集合として結果木に加えられる．

密データに対する CLOSET+のふるまいは以上である．

3. 提 案 手 法

3.1　ノイズ入りデータにおける刈り取り

2.1 節で，ノイズが含まれたデータから本来のデータにおけるアイテム集合のサポートをどのように推定するかを説明した．しかし，このようにして得られた推定サポートは相関ルールマイニングにおいて重要なアプリオリ特性を満たさない．アプリオリ特性は任意のアイテム集合のサポートは，その部分集合のサポートを上回らないというアイテム集合の性質のことであり，現在までに提案されている主要な相関ルールマイニングはこの特性を利用して，頻出アイテム集合の候補集合を効率的に刈り取っている．データから得られるアイテム集合は膨大な数となり，それら全てについて推定サポートを計算するのは高コストであることから，推定サポートがアプリオリ特性を満たさないこの場合も，アイテム集合のなんらかの刈り取りは欠かせない．ノイズを含むデータに対して刈り取りを導入するため，まずは式 5 を考える．

$$\sigma^2 = \frac{1}{N}\sum_{l'=0}^{k} {s'}_{l'} \left(q[k \leftarrow l']^2 - q[k \leftarrow l']\right). \qquad (5)$$

これは, [2] が推定サポート s_{est} を求める式と同時に与えている，s_{est} の不偏分散を表す式である．

このとき，領域 $[s_{est} - \sigma, s_{est} + \sigma]$ が最小サポート条件を満たさないアイテム集合は，本来のデータにおいても最小サポート条件を満たさないとして，我々は [2] の手法に基づき，推定サポートを σ だけ修正した値 s_{modify} が最小サポートを満たさないアイテム集合があれば，それを含むどんな超集合も最小サポートを満たさないと仮定して，そのアイテム集合を刈り取ることとした．推定による頻出アイテム集合マイニングにおいては，本来なら最小サポート条件を満たさない集合が頻出であるとして検出される場合と，本来なら最小サポート条件を満たす集合が検出されない場合の２つのエラーが起こる．前者のエラーのように，検出されるべきではないのに検出されるアイテム集合を *false positive*，後者のように検出されるべきが検出されないアイテム集合を *false negative* と呼ぶ. [2] は false negative が発生する後者のエラーがより問題であるとして，s_{modify} を $s_{est} - \sigma$ ではなく $s_{est} + \sigma$ と設定している．しかし，頻出飽和アイテム集合を扱う本研究では，本来なら頻出飽和アイテム集合と検出されるはずの集合が，false positive とサポート値が一致し，かつ false positive の部分集合であった場合，検出されずに false negative となる２重のエラーが起こる可能性がある．そのため，false positive が多い場合は前者のエラーも深刻になるかも知れない．本実験では時間の関係上，$s_{est} + \sigma$ の判定条件のみを行ったが，$s_{est} - \sigma$ については今後検証する予定である．

以上をまとめると，推定サポートの修正値 $s_{modify} = s_{est} + \sigma$ が最小サポート条件を満たさないアイテム集合があれば，それを部分集合に持ついかなる超集合も最小サポート条件を満たさないと仮定して，刈り取りを行う．

3.2　飽和性の判定

ある集合が飽和アイテム集合であるかどうかは，同値のサ

ポートを持つ超集合が存在しないことを調べればよい．本提案手法は CLOSET+ライクの深さ優先探索型であるため，新しく発見した頻出アイテム集合が既に発見済みの頻出飽和アイテム集合の部分集合であり，かつサポートが等しいか否か確かめれば十分である．しかし，ノイズ入りデータから求めた推定サポートは，本来のデータから求められる真のサポートから，ある程度の差が生じていると考えられる．そこで我々は，２アイテム集合のサポートに差があるかどうかを，推定サポートから次のように判定する．

既に発見済みの頻出飽和アイテム集合を X，新しく発見した頻出アイテム集合を Y とし，これら２つの集合間には $X \supset Y$ が成り立つと仮定する．我々は，Y の真のサポート ${s_y}^T$ と X の真のサポート ${s_x}^T$ の大小関係の判定を，次のように考えた．X と Y の２つの集合のサポートの推定値に関する確率変数を，それぞれ互いに独立な s_x，s_y としたとき，２つの確率変数の差の平均 $E(|s_x - s_y|)$ が著しく大きい場合は，２つの確率変数に差があると考えることとした.「著しく大きい」と判断する基準には，２確率変数の差の標準偏差 $\sqrt{V_{xy}}$ を用いる．すなわち，差の平均 $E(|s_x - s_y|)$ が差の標準偏差 $\sqrt{V_{xy}}$ を上回るとき，真のサポート ${s_x}^T$，${s_y}^T$ が ${s_x}^T < {s_y}^T$ を満たすと判定することを考えた．ここで，$E(|s_x - s_y|) = |E(s_x) - E(s_y)|$ であり，$E(s_x)$，$E(s_y)$ は式 4 よりそれぞれ集合 X，Y の推定サポートである．以降，X，Y の推定サポート，すなわち $E(s_x)$，$E(s_y)$ を，各々$\overline{s_x}$，$\overline{s_y}$ と記述する．

さらに，判定条件をより具体化するため，差の分散 V_{xy} を以下で導く．

$$\begin{aligned} V_{xy} &= \mathbf{E}(((s_x - s_y) - (\overline{s_x} - \overline{s_y}))^2) \\ &= {\sigma_x}^2 + {\sigma_y}^2 \end{aligned}$$

上式で与えられる分散 V_{xy} を用いた飽和性の判定条件をまとめると以下となる．

$$|\sqrt{V_{xy}}| < |\overline{s_x} - \overline{s_y}| \quad \Rightarrow \quad {s_x}^T < {s_y}^T$$

この式は，$\overline{s_x} < \overline{s_y} + \sqrt{V_{xy}}$ と，$\overline{s_x} < \overline{s_y} - \sqrt{V_{xy}}$ のケースが考えられる．

前者は集合 X と集合 Y には飽和関係がないと判定されやすく，厳しい条件である．頻出飽和アイテム集合の利点であるコンパクト性は薄れるが，3.1 節で説明したような２重のエラーが起こりにくくなると考えられる．本稿では，この判定条件 $\overline{s_x} < \overline{s_y} + \sqrt{V_{xy}}$ を用いて実験を行った．後者の判定条件 $\overline{s_x} < \overline{s_y} - \sqrt{V_{xy}}$ は，前者とは逆に飽和関係があると判定されやすい，緩い条件である．頻出飽和アイテム集合のコンパクト性が期待できるが，２重のエラーは起こりやすくなると考えられる．今後，この条件に対する検証実験も行う予定でいる．

以上の記述から我々が提案するマイニングアルゴリズムを，次の小節で述べる．

3.3 提案アルゴリズム

アルゴリズムをまとめる前に，図 2 をノイズ入りデータから構築した FP 木として，マイニングの流れを具体例を用いて説明する．簡単化のため，ノイズのない本来のデータは，全てのトランザクションが濃度 m であるとする．ヘッダテーブルの G のエントリから，G のノードを持つパスを辿ると，アイテム G を含むトランザクションは，アイテム A, B, E を一緒に持つものと，アイテム A, B, D を一緒に持つものの２つである．つまり，ノイズ入りデータベース上で集合 $AG:2, BG:2, DG:1, EG:1$ が発見された．ここで，各々の推定サポートを計算する．各集合の部分サポートはグローバル FP 木を走査することで容易に発見できる．推定サポートを計算し，3.1 節に従い絞込みを行い，仮に，AG, BG が最小サポート条件を満すとする．アイテム D, E が刈り取られたノイズ入りデータのトランザクションの情報から，$\{G\}$ の FP 木，図 3 を構築し，$\{G\}$ の FP 木で再帰的なマイニングを行う．ここで，アイテム A, B のノイズ入りデータ上のサポートは G と同値であるが，本来のデータにおける真のサポートが等しいかどうかは分からないので，アイテム集合 ABG, AG, BG の推定サポートを調べる必要がある．AG, BG については既に計算済みなので，ABG の推定サポートのみを計算し，3.2 節に従い飽和性の判定を行い，推定サポートに差がないと判断された集合に対しては item margin を適用する．もし ABG が AG, BG と同値のサポートを持つと判定されれば，sub-itemset pruning により，集合 AG および BG を部分集合にもつ飽和アイテム集合を探す必要はない．また，item skipping により，グローバル FP 木のヘッダテーブルから B のエントリを削除する．

［Remark 2］ 本提案手法では推定サポートに加え，分散 σ^2 が重要であるので，結果木のサポートテーブルには，推定サポートに加えて σ^2 の情報を保存する．

また，CLOSET+の結果木のノードの count には，パスを共有するアイテム集合の中で最大のサポートを保存するが，推定サポートがアプリオリ特性を満たさないため，短いパス，すなわち小さな集合ほど推定サポートが大きいとは限らないので，本提案手法では count に，各アイテム集合のサポートの和を保存する．

［Remark 3］ cut-and-paste randomization はノイズのない本来のデータのトランザクションの濃度に依存したパラメタを用いた randomization であり，このパラメタは推定サポートの計算時に必要である．すなわち，推定サポートを計算するためには，本来のデータのトランザクションの濃度の情報が必要不可欠である．本提案手法では，ノイズ入りデータの各トランザクションには本来のデータの対応するトランザクションの濃度が予め所与であるとして，グローバル FP 木を構築する．グローバル FP 木の各パスの末尾となるノードに，対応するトランザクションの濃度を保存する．これにより，推定サポートの計算は可能となる．

以上を踏まえて，マイニングアルゴリズムを次にまとめる．

Algolithm: 飽和アイテム集合マイニング

Input: ノイズ入りデータベース DB'，トランザクションの各サイズ m に対応するパラメタ ρ_m，K_m，最小サポート min_sup

Output: 頻出飽和アイテム集合の完全集合 CF

Method:

1. DB' をスキャンし，各単一アイテムのサポートをカウントする．min_sup を下回るサポートのアイテムは刈り取り，残ったアイテムをサポートの降順に並べたリスト f_list を作る．
2. DB' をスキャンし，f_list を用いてグローバルな FP 木 fpt を構築する．その後，DB' が疎であるか密であるかに合わせたヘッダテーブルを構築する．
3. ヘッダテーブルから fpt を辿ることで，全ての頻出アイテムに対してマイニングを行う．item marging，sub-itemset pruning を行いながら再帰的に新たな FP 木を構築し，ヘッダテーブルに対して item skipping を行う．サポートを推定するときに必要な部分サポートは，ヘッダテーブルを辿ることで全て求める．推定サポートと σ^2 を求め，修正値 s_{modify} を得る．$s_{modify} \geqq min\_sup$ ならば，結果木 $result$ をスキャンし，目的のアイテム集合が飽和アイテム集合であるかを調べ，条件を満たすなら $result$ に加える．
4. ヘッダテーブルの全てのアイテムについて fpt をマイニングしたら，$result$ から CF を作って出力する．

4. 実　　験

本提案手法でマイニングした結果の評価実験を行った．

提案手法の実装には Java で行い，１ G メモリの Linux デスクトップ PC を使用した．刈り取りにおける判定条件は $s_{est} + \sigma$，飽和性の判定条件は $\overline{s_x} < \overline{s_y} + \sqrt{V_{xy}}$ に設定した．

4.1 データセット

最初に, [2] の cut-and-paste randomization とサポートの復元方法には，限界があることを明らかにしておく．プライバシーを保護するためには，トランザクションの濃度が大きくなればなるほど，多くの「真の」情報を削り,「偽の」情報を加えなければならず，そのため推定サポートの精度が格段に落ちる．トランザクションの濃度が大きいデータに対しては，この Randomization は有効ではない．また，アイテム集合の大きさについても，大きいアイテム集合ほど，推定サポートの精度が落ちる傾向がある．本実験では上記を踏まえて，3-アイテム集合までをマイニングすることとした．またトランザクションの濃度は全て３である．cut-and-paste randomization は各トランザクションに独立に作用する Randomization であるので，トランザクションの濃度が一様であることは，推定サポートの精度に影響を与えない．大きなトランザクションやアイテム集合への対処は，本稿では Left open とする．

実験に使用したデータは全て IBM Almaden によるデータ生成器 [6] で生成した．いくつか得られた結果の中から，例として，次のデータについて説明する．トランザクション数 31K，総アイテム数 60 で，Randomization に必要なパラメタは $K = 7, \rho = 0.24$ とした．

Itemset Size	True F.C.I	True positive	False negative	False positive
1	47	47	0	9
2	275	212	63	94
3	56	41	15	26

表１　Data1 の結果

Table 1　Result of Data1

実験ではこのデータに cut-and-paste randomization を用いてノイズを入れ，最小サポートを 0.6% として提案手法によりマイニングを行った．そこから得られた結果を，本来のデータを飽和アイテム集合マイニングした結果と比較し，検証した結果を 4.2 節で述べる．

4.2 結果と考察

表 1 がマイニング結果の内訳である．１列目はアイテム集合の各濃度を示している．２列目には，Randomization を掛ける前の本来のデータの頻出飽和アイテム集合 (true F.C.I) の数が，濃度別に表されている．３列目は提案手法のマイニングしたもののうち，True F.C.I と一致する集合 (true poritive) の，濃度別の数である．４列目，５列目はそれぞれ，false negative，false positive の数が濃度別に入っている．

これらの結果がどれだけ正しいかを測るために，以下の尺度を導入する．

a）Recall (r)

true positive の数 tp の，true F.C.I の数 $tfci$ に対する割合で表される．提案手法から推定された頻出飽和アイテム集合のコレクションが，本来のデータ上の頻出飽和アイテム集合のコレクションを，どれだけ検出できているかを測る．

$$r = tc/tfci * 100 \quad (\%)$$

b）Precistion (p)

true positive の数 tp の，ノイズ入りデータをマイニングして得た頻出飽和アイテム集合の数 rc に対する割合で表される．ノイズ入りデータからマイニングされたアイテム集合に，どれだけ true positive が入っているかを示す．

$$p = tp/rc * 100 \quad (\%)$$

c）F-measure (f)

Recall と Precision の評価を総合的に評価する．

$$f = \frac{2}{\frac{1}{r} + \frac{1}{p}} * 100 \quad (\%)$$

以上の尺度を用いて，表 1 の Recall, Precision, F-measure を，表 2 に示す.

結果の評価

1-アイテム集合の Recall は 100%で，Precision は 8 割程度となった．2-アイテム集合については，Recall がおよそ 8 割，Precision がおよそ 7 割で，3-アイテム集合は，Recall が 7 割強,

Itemset Size	r	p	f
1	100.0	83.9	91.2
2	77.1	69.3	73.9
3	73.2	61.2	66.6

表 2　Recall，Precision，F-measure
Table 2　Recal, Precision, and F-measure

Precision が 6 割強となった．

2-アイテム集合，3-アイテム集合の評価が下がった要因として，現時点ではトランザクションの数の差が考えられる．式 5 から分かるように，推定サポートの分散はトランザクションの数に依存している．部分サポートの値にもよるが，一般に，トランザクションの数が小さいほど分散は大きくなる．σ により大きく見積もられた推定サポートの修正値により，false positive が増え，そのため飽和性の判定時にエラーが生じやすくなり，false negative も増えたのではないかと考える．今後，より大きなデータを用いた実験や，そのほかのマイニングの結果へ影響を与える要因の調査をする予定である．

5. おわりに

本稿では，プライバシーを保護する目的で，ノイズを入れたデータに対して，頻出飽和アイテム集合をどのように推定するかを検討し，FP 木を用いた CLOSET+ライクのマイニング手法を提案した．

今後の課題は，より多様なデータを用いて実験を続行すること，その際，刈り取り判定と飽和性の判定に使用する条件を変えて試みること，マイニングの評価結果がデータのどの部分に影響されるのか明らかにすることなどが上げられる．

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)，ならびに CREST「自律連合型基盤システムの構築」の助成による．

文　　献

[1] J. Han, J. Pei and Y. Yin. “Mining Frequent Patterns without Candidate Generation,” SIGMOD Conference, 2000
[2] A. V. Evfimievski, R. Srikant, R. Agrawal and J. Gehrke. “Privacy preserving mining of association rules,” Inf. Syst, 2004
[3] S. Rizvi and J. R. Haritsa. “Maintaining Data Privacy in Association Rule Mining,” VLDB, 2002
[4] J. Wang, J. Han, and J. Pei. “CLOSET+: Searching for the best strategies for mining frequent closed itemsets,” KDD, 2003
[5] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. “Discovering frequent closed itemsets for association rules,” ICDT, 1999
[6] R. Agrawal and R. Srikant. “Fast algorithms for mining association rules” VLDB, 1994