核となるアイテムセットによる頻出アイテムセット抽出数削減手法

松崎 勝彦† 平手 勇宇†† 山名 早人†††, ††††

†早稲田大学理工学部 〒169-8555 東京都新宿区大久保 3-4-1
††早稲田大学メディアネットワークセンター 〒169-8050 東京都新宿区戸塚町 1-104
†††早稲田大学理工学術院 〒169-8555 東京都新宿区大久保 3-4-1
††††国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2
E-mail: {matuzaki, hirate, yamana}@yama.info.waseda.ac.jp

あらまし データマイニングにおける頻出アイテムセット抽出問題は，データベース中に頻繁に出現するアイテムセット集合を求める問題である．一般に頻出アイテムセット抽出数は膨大であり，その中からユーザが有用な知識を発見することは難しい．このため，頻出アイテムセット抽出数削減を目的として，飽和アイテムセット抽出手法や極大アイテムセット抽出手法が提案されている．しかし，飽和，極大アイテムセット抽出手法は両極端な削減を行い，それぞれ一長一短の側面を持つ．本稿では，より適切な頻出アイテムセット抽出数削減を実現するために，アイテムセットの「サポート値の減少率」を新たな指標として設け，「象徴アイテムセット」を定義し，飽和，極大アイテムセット抽出手法を一般化する手法を提案する．象徴アイテムセットは，新たな閾値である「最大減少率」の設定によって集合の大きさが変動し，「最大」で飽和アイテムセット集合と一致し，「最小」で極大アイテムセット集合と一致する．

キーワード データマイニング，頻出アイテムセット，飽和アイテムセット，極大アイテムセット

Reducing the Number of Extracted Frequent Itemsets by Core Itemset Strategy

Katsuhiko MATSUZAKI† Yu HIRATE†† Hayato YAMANA†††,††††

†Faculty of Science and Engineering, Waseda University 3−4−1 Okubo, Shinjuku-ku, Tokyo, 169−8555, Japan
††Media Network Center, Waseda University 1−104 Totsuka-cho, Shinjuku-ku, Tokyo, 169−8050, Japan
†††Science and Engineering, Waseda University 3−4−1 Okubo, Shinjuku-ku, Tokyo, 169−8555, Japan
††††National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan
E-mail: {matuzaki, hirate, yamana}@yama.info.waseda.ac.jp

Abstract In general, frequent itemset mining generates a huge number of itemsets. Major approaches against this problem are extracting “closed frequent itemset(CFI)” and “maximal frequent itemset(MFI)”, instead of all frequent itemsets. However, both closed and maximal itemset mining are too extreme approaches so that they have drawbacks and advantages. In this paper, we propose more effective method for reducing the number of frequent itemsets by generalizing both closed and maximal itemset mining. In concrete terms, we introduce “rate of support decrease measure (RSDM)” and define “landmark frequent itemset(LFI)”. LFIs are dependent on value of RSDM threshold, and they become same as CFIs in the maximum, while they become same as MFIs in the minimum.

Keyword Data Mining, Frequent itemset, Closed frequent itemset, Maximal frequent itemset

1. はじめに

近年，ネットワーク環境の整備，記憶装置の大容量化，低価格化が急速に進んでおり，企業や Web 上に大量のデータが蓄積されている．そこで，大規模なデータから有用な知識を抽出する技術としてデータマイニングが注目されている．

データマイニングの重要な問題として，頻出アイテムセット抽出がある．頻出アイテムセット抽出とは，データベース中に，最小サポート値（閾値）以上の頻度で出現するアイテムセット集合を抽出する問題である[1]．しかし，頻出アイテムセット抽出においては，以下に示す問題がある．

(1) 計算コストが高い．
(2) 抽出アイテムセット数が膨大である．

(1)の問題点を解決するために，頻出アイテムセット抽出の高速化を目的として，Apriori[2]，FP-growth[3]をはじめとするアルゴリズムが多数提案されてきた．

(2)の問題点を解決するために，頻出アイテムセット抽出数削減を目的として，頻出アイテムセット集合の可逆的削減手法[4, 5, 6]と不可逆的削減手法[4, 7, 8,

9，10，11，12]が提案されている．可逆的削減手法とは，もとの頻出アイテムセット集合を完全に復元可能な削減手法である．可逆的削減手法として飽和アイテムセット抽出手法[5]が提案されている．飽和アイテムセット，すなわち CFI(Closed Frequent Itemset)とは，頻出アイテムセット集合において，サポート値の等しいスーパーセットが存在しないアイテムセットである．

一方，不可逆的削減手法とは，もとの頻出アイテムセット集合を復元する際に，情報の欠落を伴う削減手法である．頻出アイテムセット抽出数削減に伴い，以下に示す情報の欠落が考えられる．

(a) 頻出アイテムセットのサポート値情報の欠落．
(b) もとの頻出アイテムセット集合を復元する際の頻出アイテムセットの過不足．

不可逆的削減手法として提案された極大アイテムセット抽出手法[7]は，情報の欠落を(a)のみ許容する削減手法である．極大アイテムセット，すなわち MFI(Maximal Frequent Itemset)とは，頻出アイテムセット集合において，スーパーセットが存在しないアイテムセットである．また，近年，(a)，(b)の情報の欠落をある程度許容するかわりに，頻出アイテムセット抽出数の大幅な削減を実現する手法が提案されている[10，11，12]．

本提案手法では，頻出アイテムセット抽出数の大幅な削減を目指すあまり，(b)の情報の欠落を伴うことは不適切であると考え，情報の欠落を(a)のみ許容するという立場をとる．この立場は，個々の頻出アイテムセットのサポート値情報は一部欠落するが，頻出アイテムセット集合を過不足なく復元可能を意味する．そこで，既に述べた飽和アイテムセット抽出手法と極大アイテムセット抽出手法に注目する．

飽和アイテムセット抽出手法は，可逆的削減という条件のもと，頻出アイテムセット集合を「最小」に削減する．また，極大アイテムセット抽出手法は，情報の欠落を(a)のみ許容するという条件のもと，頻出アイテムセット集合を「最小」に削減する．どちらの手法も，頻出アイテムセット集合において，スーパーセットの存在するアイテムセットの抽出を抑えることで，頻出アイテムセット抽出数削減を実現し，以下に示す包含関係が成立する．

$$\mathrm{MFIs} \subseteq \mathrm{CFIs} \quad (1)$$

しかし，飽和，極大アイテムセット抽出手法は，頻出アイテムセット集合の両極端な削減を行い，それぞれ一長一短の側面を持つ．つまり，飽和アイテムセット抽出手法は，可逆的な削減を実現するために，個々の頻出アイテムセットのサポート値情報を過度に考慮するため，一般に抽出アイテムセット数をほとんど削減しない．一方，極大アイテムセット抽出手法は，個々の頻出アイテムセットのサポート値情報の欠落を完全に許容するため，一般に過度に抽出アイテムセット数を削減する．

そこで，本稿では，より適切な頻出アイテムセット抽出数削減を実現するために，飽和，極大アイテムセット抽出手法を一般化する手法を提案する．具体的には，アイテムセットの「サポート値の減少率」と呼ぶ新たな指標を設け，「象徴アイテムセット」を定義し，飽和，極大アイテムセット抽出手法の一般化を実現する．さらに，象徴アイテムセットを抽出するアルゴリズムとして，FPLandmark を提案する．FPLandmark は，飽和アイテムセットを効率的に抽出するアルゴリズムである FPclose[4]を拡張することにより設計する．

本稿は以下の構成をとる．第 2 節では，関連研究として頻出・飽和・極大アイテムセット抽出手法，さらには頻出アイテムセット集合の近似手法について述べる．第 3 節では，提案手法について述べる．第 4 節では提案手法を実装した実験結果について述べる．最後に，第 5 節でまとめる．

2. 関連研究

本節では，まず頻出アイテムセット抽出手法について述べる．次に頻出アイテムセット抽出数の削減を目的とした，飽和アイテムセット抽出手法と極大アイテムセット抽出手法について述べる．最後に，近年の頻出アイテムセット集合の近似手法について述べる．

2.1. 頻出アイテムセット抽出手法

（頻出アイテムセット抽出問題）

アイテム集合を$I = \{i_1, i_2, \ldots, i_m\}$，トランザクションデータベースを$TDB = \{t_1, t_2, \ldots, t_n | t_i \subseteq I\}$とする．ここで，トランザクション数$|TDB| = n$である．アイテムセット X のサポート値 sup(X)は，TDB 中の X を含むトランザクション数である．頻出アイテムセット抽出とは，TDB において最小サポート値（=min_sup∈[0,|TDB|]）以上のサポート値を持つアイテムセット（頻出アイテムセット）を全て抽出することである．

（FP-growth）

頻出アイテムセット集合を効率的に抽出するアルゴリズムとして，2000 年に Han らによって FP-growth[3]が提案された．FP-growth は，TDB を圧縮した特殊なデータ構造である FP-tree により，候補アイテムセットを生成せず，2 回の TDB スキャンで全ての頻出アイテムセットを抽出する．

2.2. 飽和アイテムセット抽出手法

（飽和アイテムセット）

アイテムセット X が CFI であるとは，X が頻出であり，かつ以下の条件を同時に満たすアイテムセット X’が存在しないことである[5]．

1. X⊂X’
2. sup(X)=sup(X’)

（FPclose）

2003 年に Grahne らによって，FP-growth をベースに効率的に CFI を抽出する FPclose[4]が提案された．FPclose は，既に抽出された飽和アイテムセットを格納する CFI-tree を構築する．新たに抽出された頻出アイテムセットは，CFI-tree を参照し，サポート値の等しいスーパーセットが存在しなければ飽和アイテムセットとなる．

2.3. 極大アイテムセット抽出手法

（極大アイテムセット）

アイテムセット X が MFI であるとは，X が頻出であり，かつ以下の条件を同時に満たすアイテムセット X’が存在しないことである[7].

1. X⊂X’
2. $\sup(X') \geq \min\_sup$

（FPmax*）

2003 年に Grahne らによって，FP-growth をベースに効率的に MFI を抽出する FPmax*[4]が提案された．FPmax*は，既に抽出された極大アイテムセットを格納する MFI-tree を構築する．新たに抽出された頻出アイテムセットは，MFI-tree を参照し，スーパーセットが存在しなければ極大アイテムセットとなる．

2.4. 頻出アイテムセット集合の近似手法

近年，頻出アイテムセット集合の不可逆的削減手法として，近似手法が提案されている．

2004 年に Afrati らによって提案された手法[10]は，頻出アイテムセット集合を最もよく近似する k 個のアイテムセットを抽出する．近似の評価尺度は，k 個のアイテムセットのべき集合から構成される近似集合と，頻出アイテムセット集合の共通部分の大きさを用いる．

2005 年に Yan らによって提案された手法[11]は，個々の頻出アイテムセット間の包含関係だけでなく，サポート値も考慮した距離尺度により，頻出アイテムセット集合を k 個のクラスタに分類する．

2005 年に Xin らによって提案された手法[12]は，個々の頻出アイテムセット間の包含関係とサポート値を距離尺度とし，頻出アイテムセット集合をクラスタリングする．各クラスタの代表値として抽出されるアイテムセットは，非頻出アイテムセットも含む．

以上に挙げた手法は，いずれも頻出アイテムセット抽出数削減に伴い，第 1 節で述べた情報の欠落を(a), (b)ともにある程度許容するかわりに，頻出アイテムセット集合の大幅な削減を実現する（既に述べた極大アイテムセット抽出手法よりもさらに削減する）．

3. 提案手法

第 1 節で述べたように，提案手法は頻出アイテムセット抽出数削減に伴い，情報の欠落を(a)のみ許容する立場をとる．この立場をとる従来手法として，飽和，極大アイテムセット抽出手法が提案されている．しかし，それらの手法は，頻出アイテムセット集合の両極端な削減を行い，それぞれ一長一短の側面を持つ．

そこで，本稿では，より適切な頻出アイテムセット抽出数削減を実現するために，飽和，極大アイテムセット抽出手法を一般化する手法を提案する．具体的には，アイテムセットの「サポート値の減少率」と呼ばれる新たな指標を設け，「象徴アイテムセット」を定義する．さらに，象徴アイテムセットを抽出するアルゴリズムとして FPLandmark を提案する．

3.1. 象徴アイテムセット

3.1.1. 象徴アイテムセットの定義

象徴アイテムセットは，以下に示す定性的な特徴を同時に満たす．

1. 頻出である（最小サポート値制約を満たす）．
2. 象徴アイテムセットに任意のアイテム（アイテムセット自身に含まれるアイテムを除く）を追加したアイテムセットのサポート値は，象徴アイテムセットのサポート値に比べ極端に低い（新たな閾値である「最大減少率」を超える），または頻出でなくなる．

具体的には，象徴アイテムセットを以下のように定義する．

（象徴アイテムセット）

アイテムセット X が「象徴アイテムセット」であるとは，X が頻出であり，かつ以下の条件を同時に満たすアイテムセット X’が存在しないことである．

1. X⊂X’
2. $\sup(X') \geq \min\_sup$
3. （X から X’へのサポート値の減少率）≦（最大減少率（=max_dec，新たな閾値））

ここで，X から X’へのサポート値の減少率（=sup_dec(X, X’)）を以下のように定義する．

$$\mathrm{sup\_dec}(X, X') = 1 - \frac{\sup(X')}{\sup(X)} \qquad (2)$$

ここで，以下の補題が成り立つ．

（補題 1：サポート値の減少率の定義域）

X⊂X’を満たす頻出アイテムセット X，X’に対して，sup_dec(X, X’)∈[0,1]が成り立つ．

（補題 1 の証明）

補題を証明するには，以下を示せば十分である．

$$0 \leq \frac{\sup(X')}{\sup(X)} \leq 1 \qquad (3)$$

式(3)を証明する．アイテムセット X, X’ について，

X⊂X’であれば，性質アプリオリより，sup(X)≥sup(X’)が成立する．また，sup(X’)は，最小値として 0 をとる．以上の議論より，$0 \le \sup(X') \le \sup(X)$が成立する．よって(3)が示される．

補題 1 より，提案手法では，新たな閾値である max_dec∈[0,1]をユーザが与え，頻出アイテムセット集合において，スーパーセットの存在するアイテムセットの抽出指標とする．これにより新たな問題が定義される．すなわち，提案手法は，「TDB と min_sup に加えて，max_dec が与えられ，全ての象徴アイテムセットを抽出する」．以降，象徴アイテムセットを LFI(Landmark Frequent Itemset)，さらに max_dec を明記する場合，max_dec=t の LFIs を LFIs_t とする．

3.1.2. 象徴アイテムセットによる頻出アイテムセット抽出数削減手法の一般化

3.1.1 で定義した象徴アイテムセットは，飽和，極大アイテムセットを一般化したアイテムセットである．以下では，それを順番にそれを示す．

象徴アイテムセット集合は，新たな閾値である max_dec の設定により，集合の大きさが変動する．ここで以下の補題を与える．

（補題 2：象徴アイテムセット集合の変動）

任意の 2 つの max_dec t_1，$t_2 \in [0,1]$について，$t_1 \le t_2$ならば（min_sup=一定），

$$LFIs\_t_2 \subseteq LFIs\_t_1 \quad (4)$$

となる．つまり，象徴アイテムセット集合は max_dec の増加に対して，集合の大きさが単調減少する．

（補題 2 の証明）

任意のアイテムセット $X \in LFIs_t_2$は，X⊂X’を満たす頻出アイテムセット X’が存在しない，または存在する任意の X’に対して，$sup_dec(X, X') > t_2 \ge t_1$となる．よって，任意のアイテムセット $X \in LFIs_t_2$は，$X \in LFIs_t_1$となる．

補題 2 より，LFIs は，max_dec=0 と設定した場合「最大」，max_dec=1 と設定した場合「最小」となる．ここで，さらに以下の補題を与える．

（補題 3：象徴アイテムセットの「最大」）

max_dec=0 と設定した場合，象徴アイテムセットは，飽和アイテムセットと一致する．すなわち，

$$LFIs\_0 = CFIs \quad (5)$$

（補題 4：象徴アイテムセットの「最小」）

max_dec=1 と設定した場合，象徴アイテムセットは，極大アイテムセットと一致する．すなわち，

$$LFIs\_1 = MFIs \quad (6)$$

（補題 3 の証明）

max_dec=0 と設定した場合，象徴アイテムセットの定義におけるアイテムセット X’の条件 3 は，条件 sup(X)=sup(X’)と等しい．したがって，条件 3 を満たす X’は明らかに条件 2 を満たす．よって，条件 1，3 のみがチェックされることになり，これは既に述べた飽和アイテムセットの定義と一致する．

（補題 4 の証明）

max_dec=1 と設定した場合，象徴アイテムセットの定義におけるアイテムセット X’の条件 3 は常に真となる．よって，条件 1，2 のみがチェックされることになり，これは既に述べた極大アイテムセットの定義と一致する．

補題 2，3，4 より CFIs，MFIs，LFIs の包含関係は以下のようになる．

（CFIs，MFIs，LFIs の包含関係）

$$MFIs \subseteq LFIs \subseteq CFIs \quad (7)$$

以上で示された CFIs，MFIs，LFIs の包含関係を図 1 に示す．すなわち，LFIs は，max_dec の増加に対して集合の大きさが単調減少し，max_dec=0 とした場合，「最大」となり CFIs と一致し，max_dec=1 とした場合，「最小」となり MFIs と一致する．

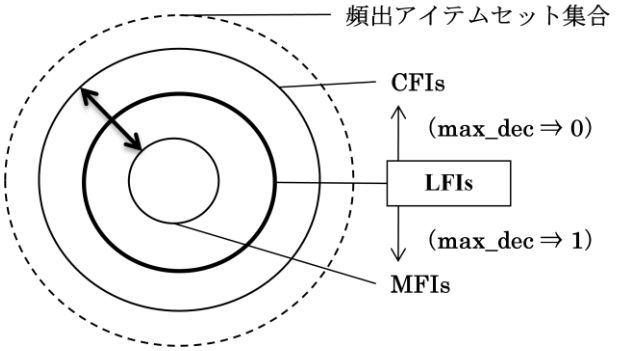

図 1 CFIs，MFIs，LFIs の包含関係

3.2. 象徴アイテムセットを抽出するアルゴリズム

3.2.1. FPLandmark

飽和，極大アイテムセット抽出手法の問題点を解決するために，3.1 で CFI，MFI を一般化する LFI を定義した．ここでは，LFI を抽出するアルゴリズムについて述べる．3.1 で示したように，LFIs は，最大で CFIs となる．したがって，CFIs から LFIs が抽出可能である．よって，提案手法である象徴アイテムセット抽出アルゴリズム（FPLandmark）は，飽和アイテムセット抽出アルゴリズムをベースにする．

FPLandmark は，飽和アイテムセット抽出アルゴリズムとして第 2 節で紹介した FPclose[4]を拡張する．FPclose は，FP-growth と同様に TDB を 2 回スキャンすることで，FP-tree を構築した後，再帰的に FP-tree を探索し，全ての CFI を抽出する．既に抽出された CFI を FP-tree と同様のデータ構造である CFI-tree に格納しておくことで，新たに抽出された頻出アイテムセットが CFI となるかどうかを CFI-tree を参照することにより判断する．CFI-tree は FP-tree と同様に，ヘッダテー

ブルと木構造で構成される．CFI-tree の各ノードは，アイテム名，レベル（root ノードから該当ノードまでにたどるエッジの数），カウント値，ノードリンクの 4 つのフィールドを持つ．ここで補題を与える．

（補題：LFI の判定）

FP-tree から新たに抽出された頻出アイテムセットが，LFI であるかどうかを調べるには，既に抽出された CFI との比較のみで十分である．

（証明）

FP-growth をベースとするアルゴリズムは，深さ優先探索により頻出アイテムセットを抽出する．さらに，FPclose 同様に，ヘッダテーブルのアイテムについてボトムアップに FP-tree を探索することで，新たに抽出された頻出アイテムセットは，既に抽出された頻出アイテムセットのスーパーセットになることはない．また，CFI は頻出アイテムセット全てのサポート値情報を保持するため，補題が成立する．

補題より，新たに抽出された頻出アイテムセット X は LFI であるかどうかによらず，CFI であれば，X は CFI-tree に挿入される．これにより，CFI の判定に利用していた CFI-tree を LFI の判定にも利用できる．

（アルゴリズムの概要）

象徴アイテムセット抽出アルゴリズムの概要を述べる．まず，ヘッダテーブルのアイテムについてボトムアップに FP-tree T を再帰的に構築していき，T が単一パス P から構成される場合，P からトップダウンにローカルな CFI X を生成し，同時に X がローカルな LFI となるかを調べる．ここで，「ローカルな」とは，「X が単一パス P から生成可能な頻出アイテムセット集合において」という意味である．X がローカルな CFI であるかは，単一パス P の各ノードのカウント値から容易に判断できる．同様にして X がローカルな LFI であるかを判断できる．具体的には，単一パスを root ノードからたどり，カウント値の変化するノードを見つける．カウント値が変化していれば，ローカルな CFI となり，さらにカウント値の変化率が max_dec を超えていれば，ローカルな LFI となる．ローカルな CFI X がローカルな LFI であれば，CFI-tree C を参照することで，X がグローバルな CFI であるか，また，X がグローバルな LFI であるかを調べる．ローカルな CFI X がローカルな LFI でなければ，C を参照し，X がグローバルな CFI であるかのみ調べる．

以上を踏まえ，具体的なアルゴリズムを示す．なお，FP-tree T は base，header，array の 3 つのエントリを持つ．base は T_X におけるアイテムセット X である．header は T に対応したヘッダテーブルである．array は T に対応した配列である．FPLandmark を図 2 に示す．

（アルゴリズム）

入力：TDB，min_sup，max_dec
出力：LFIs
メソッド：

1. TDB に対する初期 FP-tree $T_\emptyset$ を構築する．$T_\emptyset$ に対応した CFI-tree $C_\emptyset$ を ϕ で初期化する．
2. LFIs L をグローバルに定義し，ϕ で初期化し，関数 FPLandmark($T_\emptyset$, $C_\emptyset$)を呼ぶ．

関数 FPLandmark(T,C)

入力：FP-tree T，T.base に対する CFI-tree C
出力：更新された C
メソッド：

1. T が単一パス P で構成されている場合，P から生成したローカルな各 CFI X に対して，以下の処理を行う．
   i. X がローカルな LFI となる場合，以下の処理を行う．
      (a) C を参照し，X⊂Y かつ sup(X)=sup(Y)を満たすアイテムセット Y が存在するか，さらに X⊂Z かつ sup_dec(X,Z)≤max_dec を満たすアイテムセット Z が存在するか調べる．
      (b) Z が存在しなければ，X を L に挿入する．
      (c) Y が存在しなければ，X を C に挿入する．
   ii. X がローカルな LFI とならない場合，以下の処理を行う．
      (a) C を参照し，X⊂Y かつ sup(X)=sup(Y)を満たすアイテムセット Y が存在するか調べる．存在しなければ X を C に挿入する．
2. T が単一パスで構成されていない場合，T.header の各アイテム i に対して以下の処理を行う．
   i. アイテムセット T.base∪{i}を Y とする．
   ii. C を参照し，Y⊂Y'かつ sup(Y)=sup(Y')を満たすアイテムセット Y'が存在するか調べる．存在しなければ以下の処理を行う．
      (a) T.array から，i に対して全ての頻出なアイテムから構成されるアイテムセットを Tail とする．
      (b) Tail をサポート値降順にソートする．
      (c) Y's conditional FP-tree T_Y と対応した配列 A_Y を構築する．
      (d) Y's conditional CFI-tree C_Y を C から初期化する．
      (e) FPLandmark(T_Y, C_Y)を呼ぶ．
      (f) C_Y を C にマージする．

図 2 FPLandmark

（FPLandmark の正当性）

上記議論と FPclose が正確に全ての CFI を抽出すること，CFI-tree が既に抽出された頻出アイテムセットをサポート値情報も含めて完全に保持していることにより，FPLandmark は全ての LFI を抽出する．

（動作例）

表 1に示す TDB を例に FPLandmark の動作例を述べる．ここで，min_sup=2，max_dec=0.3 とする．なお，サポート値=s のアイテムセット X を{X:s}と表記する．表 1 の TDB から構築された FP-tree を図 3(a)に示す．FPLandmark は，ヘッダテーブルの末尾から順に，FP-tree を探索することで，LFI を抽出していく．ここで，ヘッダテーブルにおいてアイテム d，f，b，g を含む LFI は抽出済みであり，その過程で構築された CFI-tree を図 3(c)に示す．

アイテム a を含む LFI を抽出する．a's conditional FP-tree を図 3(b)に示す．図 3(b)より，ローカルな LFI は{c,e,a:6}となる．{c,e,a:6}は CFI-tree を参照することにより，スーパーセット{e,c,a,b:2}と{e,c,a,g:4}が存在する．しかし，両スーパーセットへのサポート値の減少率は，ともに max_dec=0.3 を超えているため，ローカルな LFI{c,e,a:6}はグローバルな LFI となる．

表 1 TDB の例

TID	Items	TID	Items
100	a, b, c, e, f, o	600	e, j
200	a, c, g	700	a, b, c, e, f, p
300	e, i	800	a, c, d
400	a, c, d, e, g	900	a, c, e, g, m
500	a, c, e, g, l	1000	a, c, e, g, n

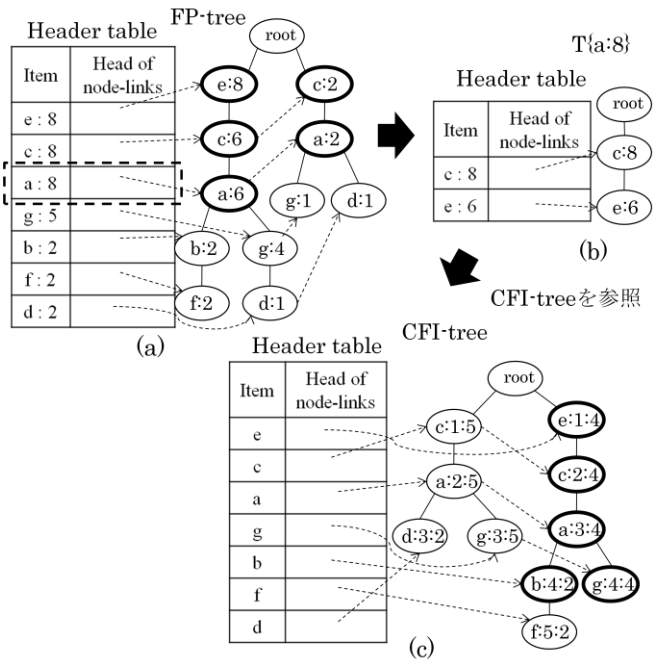

図 3 FPLandmark の動作例

3.2.2. 空間使用量，計算量の比較

FPLandmark は，FPclose 同様，FP-tree と CFI-tree を必要とするため，空間使用量は，およそ初期 FP-tree と全ての CFI を格納した CFI-tree の合計になる．これは FPclose と同等の空間使用量である．

FPLandmark は，抽出された頻出アイテムセットが CFI であるかを判定した後，LFI であるかを調べるため，結果的に全ての CFI を抽出している．よって FPLandmark は FPclose よりも計算量が多い．

4. 評価実験

第 3 節で提案した LFI を抽出するアルゴリズムである FPLandmark を実装し，抽出アイテムセット数，抽出アイテムセットにおけるアイテムセット長の分布，実行時間を評価した．

4.1. 実験環境とデータセット

第 3 節で提案した手法を，Intel(R) Core2 Duo 3.16GHz×2 プロセッサ，4GB のメモリを搭載した計算機 1 台で実行した．なお，実行毎に，マシンの再起動によりディスクキャッシュをクリアして実験を行った．

データセットとしては，[13]より 2 つの実データセット，retail データセット，pumsb*データセットを入手して評価を行った．retail データセットは，1999 年から 2000 年にかけての 5 ヶ月間に記録された,ベルギーのあるスーパーマーケットの購買履歴データである．pumsb*データセットは，アメリカの国勢調査の Public Use Microdata Sample(PUMS)から生成されたデータセットである．retail データセットは，トランザクション数 88162，アイテム総数 16470 で疎なデータセットである．pumsb*データセットは，トランザクション数 49046, アイテム総数 2088 で密なデータセットである．

4.2. 抽出アイテムセット数の評価

提案手法を 4.1 で述べた 2 つのデータセットに対して適用させた時の，最小サポート値と抽出アイテムセット数の関係を図 4，図 5 に示す．なお，飽和，極大アイテムセット抽出手法についても評価を行い同図内に示す．凡例において,「max_dec=0.25」とは，提案手法において max_dec=0.25 とした時の評価結果を表す．

図 4，図 5 より，抽出アイテムセット数という観点において, LFIs は max_dec の設定により, CFIs と MFIs の間を変化することがわかる．具体的には，max_dec を 0 に近づけていけば，飽和アイテムセット抽出数に近づき，max_dec を 1 に近づけていけば，極大アイテムセット抽出数に近づく．つまり，3.1.2 で示したように，LFIs の大きさは，max_dec の増加に対して単調減少することが確認できる．

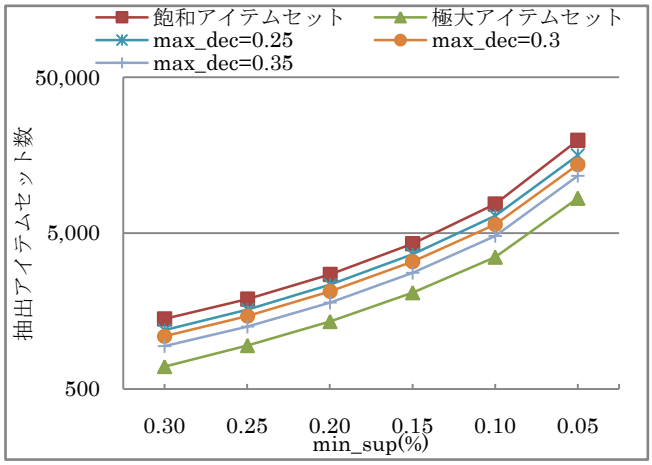

図 4 抽出アイテムセット数（retail）

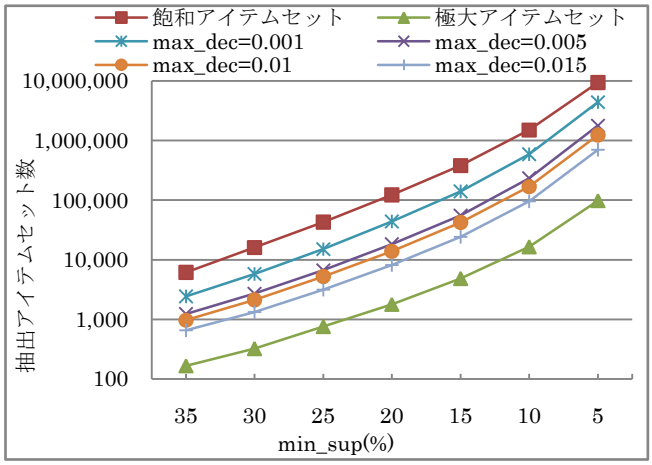

図 5 抽出アイテムセット数（pumsb*）

4.3. 抽出アイテムセット長の分布評価

次に，提案手法を 4.1 で述べた 2 つのデータセットに対して適用させた時の，抽出アイテムセットのアイテムセット長の分布評価を図 6，図 7 に示す．なお，飽和，極大アイテムセット抽出手法についても評価を行い同図内に示す．4.2 の抽出アイテムセット数という観点において，LFIs の大きさは，max_dec の増加に対して単調減少することを確認した．ここでは，抽出されるアイテムセットのアイテムセット長の分布という観点から，CFIs, MFIs, LFIs の包含関係を検証した．

図 6，図 7 より，アイテムセット長の分布評価において，LFIs は max_dec を 0 に近づけていけば，抽出されるアイテムセット集合の中身自体も，CFIs に近づいていき，max_dec を 1 に近づけていけば，MFIs に近づいていくことが確認できる．すなわち，3.1.2 で示した CFIs，MFIs，LFIs の包含関係を明確にした．

4.4. 実行時間の評価

提案手法を 4.1 で述べた 2 つのデータセットに対して適用させた時の，実行時間の評価を図 8，図 9 に示す．なお，CFI を抽出するアルゴリズムである FPclose，MFI を抽出するアルゴリズムである FPmax*についても評価を行い同図内に示す．提案手法は，max_dec の

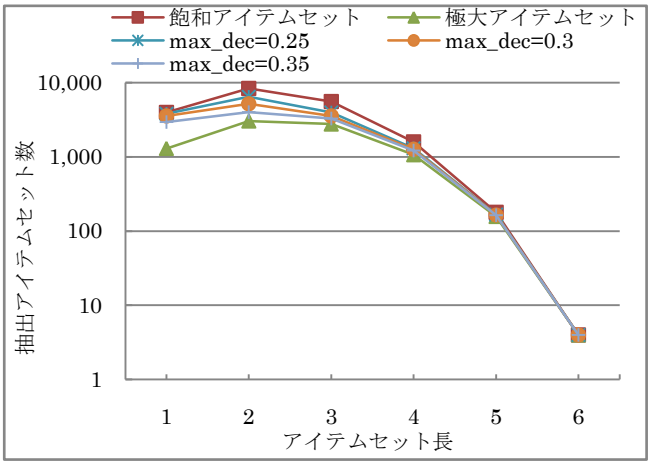

図 6 抽出アイテムセット長の分布
（retail，min_sup=0.05%）

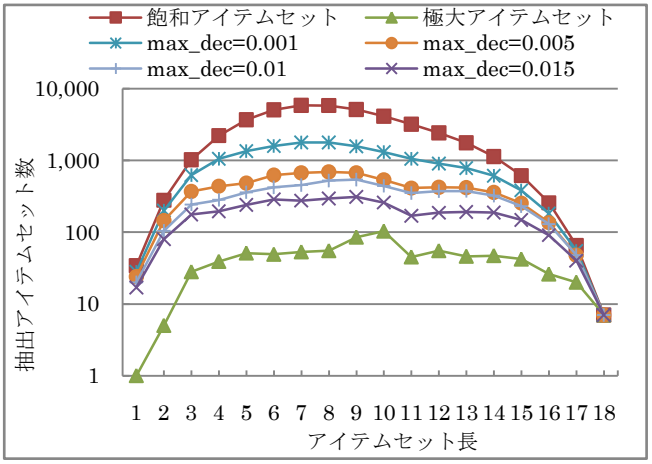

図 7 抽出アイテムセット長の分布
（pumsb*，min_sup=25%）

設定により抽出アイテムセット数は変化するが，実行時間はほとんど同等となる．これは，提案手法の大半の計算コストが CFI を抽出するコストであることに起因する．3.2 で述べたように，提案手法は max_dec の設定によらず，結果的に全ての CFI を生成するため，max_dec の設定による実行時間の変化がほとんどない．よって，提案手法の評価結果は 1 つのみ示してある．

図 8，図 9 より，提案手法は，CFI を抽出するアルゴリズムである FPclose とほぼ同等の実行時間で LFI を抽出できることがわかる．これは，LFI の判定は CFI の判定と同時に行うため，LFI を抽出する際の計算コストの大半は CFI を抽出することであるためである．

具体的な実行時間として，retail データセットにおいて min_sup を 0.3%から 0.05%まで変化させた時，FPLandmark と FPclose の実行時間は完全に一致した．また，pumsb*データセットにおいては，図 9 では FPLandmark と FPclose の実行時間の差がわかりにくいが，実際には若干の差があったため，一部を表 2 に示す．FPLandmark は，min_sup=5%の時，FPclose の実行時間に対して約 2%遅くなっていることがわかる．

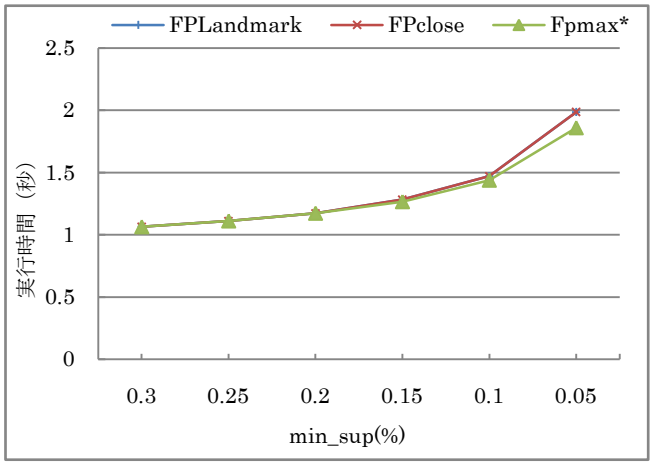

図 8 実行時間（retail）

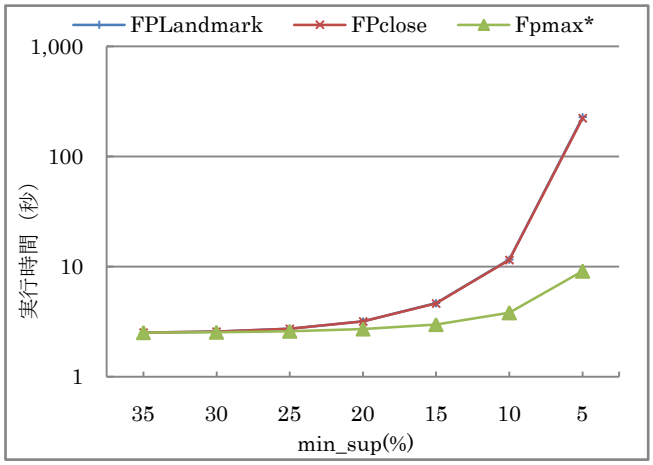

図 9 実行時間（pumsb*）

表 2 実行時間の比較（pumsb*）

	min_sup(%)		
アルゴリズム	15	10	5
FPLandmark(秒)	4.66	11.63	227.36
FPclose(秒)	4.64	11.53	222.97

5. おわりに

本稿では，頻出アイテムセット抽出数削減を目的として，アイテムセットの「サポート値の減少率」を新たな指標として設け,「象徴アイテムセット」を定義し，飽和，極大アイテムセット抽出手法を一般化する手法を提案した．さらに，象徴アイテムセットを抽出するアルゴリズムとして，FPLandmark を提案した．

提案手法を 2 つの実データセットに対して適用させた結果，新たな閾値である「最大減少率」の設定により，象徴アイテムセット集合は「最大」で飽和アイテムセット集合,「最小」で極大アイテムセット集合となり，飽和，極大アイテムセット集合の間を変動することを確認した．さらに，FPLandmark は，飽和アイテムセットを抽出するアルゴリズムである FPclose とほぼ同等の実行時間で象徴アイテムセットを抽出可能であることを確認した．

今後の課題は, FPLandmark をより洗練させることである．現状では，FPLandmark は象徴アイテムセットを抽出する過程で，結果的に全ての飽和アイテムセットを生成しているが，これは冗長なのではないかと考える．

謝　辞

本研究の一部は，科学研究費補助金「情報爆発に対応する高度にスケーラブルなモニタリングアーキテクチャ」によるものである．

参　考　文　献

[1] R.Agrawal, T.Imielinski, and A.Swami, “Mining association rules between sets of items in large databases,” In Proc. ACM SIGMOD’93, pp. 207-216, 1993.

[2] R.Agrawal and R.Srikant, “Fast algorithms for mining association rules,” In Proc. VLDB’94, pp. 487-499, 1994.

[3] J.Han, J.Pei, and Y.Yin, “Mining frequent patterns without candidate generation,” In Proc. ACM SIGMOD’00, pp. 1-12, 2000.

[4] G.Grahne and J.Zhu, “Efficiently Using Prefix-tree in Mining Frequent Itemsets,” In Proc. IEEE ICDM Workshop FIMI’03, 2003.

[5] N.Pasquier, Y.Bastide, R.Taouil, and L.Lkhal, “Discovering frequent closed itemsets for association rules,” In Proc. ICDT’99, pp398-416, 1999.

[6] T.Calders and B.Goethals, “Mining All Non-Derivable Frequent Itemsets,” In Proc. PKDD’02, pp74-85, 2002.

[7] R.J.Bayardo, “Efficiently mining long patterns from databases,” In Proc. ACM SIGMOD’98, pp.85-93, 1998.

[8] J.Han, J.Wang, Y.Lu, and P.Tzvetkov, “Mining Top-K Frequent Closed Patterns without Minimum Support,” In Proc. IEEE ICDM’02, 2002.

[9] Y.Hirate, E.Iwahashi, and H.Yamana,“TF^2P-growth: An Efficient Algorithm for mining Frequent Patterns without any Thresholds,” In Proc. IEEE ICDM’04 Workshop on Alternatives Techniques for Data Mining and Knowledge Discovery, 2004.

[10] F.Afrati, A.Gionis, and H.Mannila, “Approximating a Collection of Frequent Sets,” In Proc. ACM SIGKDD’04, pp12-17, 2004.

[11] X.Yan, H.Cheng, J.Han, and D.Xin, “Summarizing Itemset Patterns: A Profile-Based Approach,” In Proc. ACM SIGKDD’05, pp314-323, 2005.

[12] D.Xin, J.Han, X.Yan, and H.Cheng, “Mining Compressed Frequent-Pattern Sets,” In Proc. VLDB’05, pp709-720, 2005.

[13] Frequent Itemset Mining Implementations Repository, http://fimi.cs.helsinki.fi/