負の相関ルールマイニング効率化のための極小生成子の抽出計算

Extraction Calculation of Minimal Generators for Improve Efficiency of Negative Association Rules

佐生 隼一 [1*]　岩沼 宏治 [2]　山本 泰生 [2]　黒岩 健歩 [1]
Sasho Shunichi [1]　Koji Iwanuma [2]　Yositaka Yamamoto [2]　Yasuho Kuroiwa [1]

[1] 山梨大学大学院医学工学総合教育部コンピュータ・メディア工学専攻
[1] Computer Seience and Media Engineering, Interdisciplinaly Graduate School of Medieine and Engineering, University of Yamanashi
[2] 山梨大学大学院医学工学総合研究部
[2] Interdiseiplinary Graduate School of Medicine and Engineering, University of Yamanashi

Abstract:
This paper is consider extraction calculation of minimal generator that is compressed expression of itemset to realize efficient Negative Association Rules Mining. Closed itemset is used for extracting Minimal Generator. we propose bottumup and topdown Algorithm to extract minimal generetor. We also show some results of experiments for evaluating our proposed method.

1　はじめに

相関ルールとはデータベース中で頻繁に共起する事象の関係を記述したものである．この相関ルールの発見はデータマイニングにおける代表的な問題である．

X と Y をアイテム集合とするとき，トランザクションデータベース中で X が出現するトランザクションの多くに Y も出現することを，$X \Rightarrow Y$ と記述し，これを正の相関ルールと呼んでいる．これに対して，本研究で考察する負の相関ルールは，ある事象が発生した際に別の事象が生起しない現象を記述するものであり，$\neg X \Rightarrow Y$, $X \Rightarrow \neg Y$ などの形のルールとして記述される．

負の相関ルール抽出問題は近年研究が盛んになった分野 [1,2,3] である．正の相関ルールでは表現が困難な共起関係を記述でき，データベースから有益な情報を抽出することを可能にする．ただ，負の相関ルールは非頻出なアイテム集合を扱う必要があり，正の相関ルールと比べて探索空間が格段に大きい．探索の高速化と効果的化は重要な課題であった．これに対して，井出らの先行研究 [4] では高速なトップダウン型アルゴリズムが提案されている．黒岩らの研究 [5] ではルールの統計的評価尺度を併用して負ルールの抽出の高速化を行っている．先行研究では膨大に存在する頻出アイテム集合を用いてルールを抽出していた．本研究では，新しく極小生成子 (minimal generators)[6] を用いた負の相関ルールマイニングのため，極小生成子の抽出手法を提案する．頻出アイテム集合の圧縮法は飽和集合がよく知られているが，この飽和集合を圧縮に用いた場合，本来抽出すべき負ルールが抽出できなくなるなどの現象が生じることが知られている [7]．そのため圧縮率は低くなるが完全な負ルールの抽出が可能となる極小生成子を用いたルール圧縮を行う [7]．本論文ではでは極小生成子を求める際に膨大に存在する頻出アイテム集合ではなく，飽和アイテム集合から求めることでルール探索の効率化を検討を行う．本論文の構成は以下の通りである．第 2 章は準備である．第 3 章では極小生成子に基づく負ルールの抽出の枠組みを示す．第 4 章では，今回の提案手法である極小生成子の抽出アルゴリズムについて述べる．第 5 章では本論文で提案する手法についての実験結果及び考察を示す．第 6 章はまとめである．

*連絡先：山梨大学医学工学総合教育部
コンピュータ・メディア工学専攻
〒 400-8511 山梨県甲府市武田 4-3-11
E-mail: g15mk008@yamanashi.ac.jp

2　準備

$I = \{a_1, a_2, \cdots, a_n\}$ を**アイテム**の全体集合とし，**トランザクション** t をアイテム集合 $t \subseteq I$ と定める．**トランザクションデータベース** $\mathcal{D}$ をトランザクションの多重集合とする．X をアイテム集合とするとき，$X \subseteq t$ となる $\mathcal{D}$ のトランザクション t を X の**出現**と呼び，その多重集合を $\mathcal{D}(X)$ と略記する．多重集合 A の大きさを $|A|$ と表記するとき，X の $\mathcal{D}$ 中の**支持度** $\sup(X)$ を $\sup(X) = \frac{|\mathcal{D}(X)|}{|\mathcal{D}|}$ と定義する．**正の相関ルール** (以下，“正ルール”と略記) を $X \cap Y = \emptyset$ であるアイテム集合 X，Y からなる表現 $X \Rightarrow Y$ と定める．X と Y をそれぞれルールの**前件**，**後件**と呼び，$X \cup Y$ を**台集合**

(underlying set) と呼ぶ．正ルールに対する支持度 sup と確信度 conf は以下のように定義する．

定義 1

$$\begin{aligned} sup(X \Rightarrow Y) &= sup(X \cup Y) \\ conf(X \Rightarrow Y) &= \frac{sup(X \cup Y)}{sup(X)} \end{aligned}$$

最小支持度 ms と**最小確信度** mc とはユーザが与える支持度と確信度に関する閾値である．$\sup(X) \geq ms$ を満たす X を**頻出アイテム集合**と呼ぶ．また $\sup(X \Rightarrow Y) \geq ms$ と $\mathrm{conf}(X \Rightarrow Y) \geq mc$ の両方を満たす $X \Rightarrow Y$ を**有効** (valid) な正ルールと呼ぶ．

本研究では，**負の相関ルール** (negative association rule: 以下では“負ルール”と略記) とは，アイテム集合 X と Y を $X \cap Y = \emptyset$ とする時，以下のいずれかの表現のこととする．

$$\neg X \Rightarrow Y \text{ (左否定形)}, \quad Y \Rightarrow \neg X \text{ (右否定形)}$$

上記の $\neg X$ はアイテム集合の否定表現であり，**負アイテム集合**と呼ぶ．以下では C_X は，正と負のアイテム集合 X または $\neg X$ のどちらかを表すものとする．

定義 2 ([?, ?, ?]) 負アイテム集合および負ルールの**支持度** sup と**確信度** $conf$ を以下のように定める．

$$\begin{aligned} sup(\neg X) &= 1 - sup(X) \\ sup(X \Rightarrow \neg Y) &= sup(X) - sup(X \cup Y) \\ sup(\neg X \Rightarrow Y) &= sup(Y) - sup(X \cup Y) \\ conf(C_X \Rightarrow C_Y) &= \frac{sup(C_X \Rightarrow C_Y)}{sup(C_X)} \end{aligned}$$

先行研究 [1,2,3] では，負ルールの抽出は Apriori 流の幅優先型アルゴリズムで行われていた． このため，負ルール間の関係性の検査は困難であり，効率的な枝刈りを行うことができなかった．井出ら [4] は接尾辞木を用いた深さ優先型アルゴリズムを新しく提案し，負ルール間の包含関係を効率的に検査して，探索空間を削減して高速化を達成している．

先行研究 [4, 5] では以下のような条件を満たすルール $C_X \Rightarrow C_Y$ を**有効な** (valid) 負ルールと定めていた．

定義 3 最小支持度 ms，最小確信度 mc，関連尺度の閾値 mr としたとき，有効な負ルール $C_X \Rightarrow C_Y$ とは以下の条件を満たすものである．

1. $X \cap Y = \emptyset$
2. $sup(X) \geq ms$ かつ $sup(Y) \geq ms$
3. $sup(X \Rightarrow Y) < ms$
4. $sup(C_X \Rightarrow C_Y) \geq ms$
5. $conf(C_X \Rightarrow C_Y) \geq mc$
6. $lift(C_X \Rightarrow C_Y) \geq mr$

Algorithm 1 負の相関ルール探索の大枠

トランザクションデータベースから頻出アイテム集合 (FIS) の集合を抽出し，N を抽出した頻出アイテム集合の総和とする;
FIS の集合から接尾辞木を生成;
接尾辞木上で左優先深さ優先探索で頻出アイテム集合を FIS^1，…，FIS^N の順に並べる;
for $i = 1$ to N **do**
  X = FIS^i ;
  for $j = 1$ to N **do**
    $Y = FIS^j$;
    Check_Rule(X，Y);
  end for
end for

先行研究 [4, 5] における，有効な負の相関ルールは以下の Algorithm 1 で抽出を行う．疑似コード中のアイテム集合 X，Y は接尾辞木のある節点のアイテム集合である．$Check\_Rule(X,Y)$ で $X \Rightarrow \neg Y$ と $\neg Y \Rightarrow X$ が有効な負ルールであるかをチェックしている．また，同時に適宜枝刈りを行い，探索の効率化を行っているが本論文ではその詳細は省略する．

3 極小生成子に基づく負ルール抽出

妥当な負ルールは正ルールに比べ, 本質的に大量に存在 [1] する．先行研究 [7] では，極小生成子を用いて頻出アイテム集合を圧縮し，アイテム集合の組み合わせを減少させ，有効な負ルールの抽出の効率化を行っている．

頻出アイテム集合の圧縮表現としては以下の飽和アイテム集合と閉包集合が良く知られている．**極小生成子** (minimal generator) は閉包の対になる表現である．

定義 4 ([?]) アイテム集合 X が**飽和**しているとは，以下の条件を満たす X' が存在しない場合を言う．

$$X \subset X',\ X \neq X' \text{ かつ } sup(X) = sup(X')$$

アイテム集合 Y の**閉包** $clo(X)$ とは，$Y \subset X$ かつ $sup(Y) = sup(X)$ を満たす飽和アイテム集合 X のことである．Y の**生成子**とは以下の条件を満たす Z を言う．

$$Z \subset Y \quad \text{かつ} \quad sup(Z) = sup(Y)$$

生成子 Z が**極小**とは，$Z' \subset Z$ かつ $Z' \neq Z$ となる生成子 Z' が存在しない場合を言う．

X' を X の閉包または極小生成子とすれば，X と X' はデータベース中で必ず同じトランザクションに出現する．このとき，X が出現する相関ルール $\mathcal{R}[X]$ に対して，X を X' に置き換えた相関ルール $\mathcal{R}[X']$ を考えれば，$\mathcal{R}[X]$ と $\mathcal{R}[X']$ の支持度，確信度および関連尺度は全く同じになる．よって，この 2 つのルール両方

TID	アイテム集合
1	ABC
2	AB
3	AB
4	BC
5	BC

図 1: トランザクションデータベース D

を生成することは冗長と考えられ，どちらかで代表させることが適切である．

以下の例 **??**に示すように，飽和集合を用いると本来抽出すべき妥当なルールが抽出できなくなる場合がある．極小生成子を用いることにより妥当な負ルール全てが抽出可能になる [7].

例 1 図 1 のトランザクションデータベース $\mathcal{D}$ を考え，最小支持度 $ms = 0.4$, 最少確信度 $mc = 0.4$ とする．本例では，議論の簡単化のために，関連尺度は考慮しない．また，アイテム集合はその要素の列で表記し，出現頻度を適宜付記する．例えば，出現頻度 3 のアイテム集合 $\{A, B\}$ を $AB : 3$ のように表記する．

データベース D の頻出アイテム集合は以下の通りである．

$$A : 3,\ \ B : 5,\ \ C : 3,\ \ AB : 3,\ \ BC : 3$$

この中で，頻出飽和アイテム集合は B, AB，BC の 3 つである．このとき以下の負ルール $\mathcal{R}$ を考える．

$$\mathcal{R} = (AB \Rightarrow \neg C)$$

$\sup(\mathcal{R}) = 0.4 \geq ms$, $conf(\mathcal{R}) = \frac{2}{3} \geq mc$ であり，後件の否定を外した正ルールに対しても $\sup(AB \Rightarrow C) = \frac{1}{5} < ms$ なので，$\mathcal{R}$ は妥当な負ルールである．一方で頻出アイテム集合 C の閉包は BC であることから，$\mathcal{R}$ の飽和集合による表現は

$$AB \Rightarrow \neg BC$$

となる．しかしこれは定義 2 の (1) に示した前件と後件の**独立性**条件に違反し，妥当なルールにはならない．これに対して，D 上の頻出アイテム集合に対する極小生成子は A, B, C なので，極小生成子を用いた $\mathcal{R}$ の表現は

$$A \Rightarrow \neg C$$

となる．独立性の条件を満足するので妥当な負ルールとして生成できる．

極小生成子 X から頻出アイテム集合を復元するためには，どこまで頻度が同じままアイテム集合を拡大できるかという情報が必要となる．これは閉包 $clo(X)$ を用いることで解決できる．$X \subseteq Y \subseteq clo(X)$ である Y は全て X と同じ出現頻度を持つ．これにより極小生成子から頻出アイテム集合を復元する．

4 提案手法

極小生成子を用いた負ルール抽出では飽和アイテム集合とその閉包となる極小生成子のペアを作ることが必要である．そのため本論文では飽和アイテム集合から極小生成子を抽出する手法を提案する．このため飽和アイテム集合を用いることで，大量に存在する頻出アイテム集合を生成せずに，負ルールの抽出が行うことができる．また，ルール探索の効率化及びメモリ使用量の削減などの効果があると考えられる．

4.1 極小生成子の性質

極小生成子について，以下のような性質があげられる．

補題 1 ある飽和アイテム集合 X に対して，Y がその極小生成子である場合，Y を含むような X の部分集合 X' は全て X の極小生成子になり得ない．

補題 1 から，飽和アイテム集合 X の極小生成子 Y を含むような部分集合 X' については極小生成子になりえないことがわかる．これを利用し，極小生成子抽出の枝刈りに利用することができる．

補題 2 飽和アイテム集合 X の極小生成子 Y の部分集合 Y' は X の極小生成子とならない．

補題 2 より，$|\ Y\ | \geq 2$ の極小生成子の部分集合については極小生成子となりえないことがわかる．上記の 2 つの性質から極小生成子抽出の効率化を行う．

4.2 ボトムアップ型アルゴリズム

極小生成子を飽和アイテム集合からボトムアップ型で求める手法について考える．ボトムアップ型アルゴリズムでは，飽和アイテム集合 Y' の $|\ Y\ | = 1$ となる部分集合から検査する．そのため補題 1 による極小生成子についての枝刈りを行うことができるが，部分集合についてはすでに探索済みのため，補題 2 による枝刈りは行えない．

補題 1 の枝刈りについて $|\ Y\ | \geq 2$ 極小生成子 Y を用いる場合，Y を含むようなアイテム集合の情報を保持しなければならない．そのため本論文では飽和アイテム集合 X から極小生成子 Y の差分をとることで Y を含むようなアイテム集合の生成を抑える．しかし，この手法では $|\ Y\ | \geq 2$ の極小生成子を用いることができない．以下に例を示す．

例 2 図 2 より求められる飽和アイテム集合は，B, C, D, BCD, $ABCD$ となり，極小生成子は A, B, C, D, BC, BD, CD となる．この時，飽和アイテム集合 BCD から極小生成子 BC の差分をとると D となり，BD, CD が極小生成子として抽出できなくなる．

TID	アイテム集合
1	ABCD
2	ABCD
3	BCD
4	B
5	C
6	D

図 2: トランザクションデータベース D

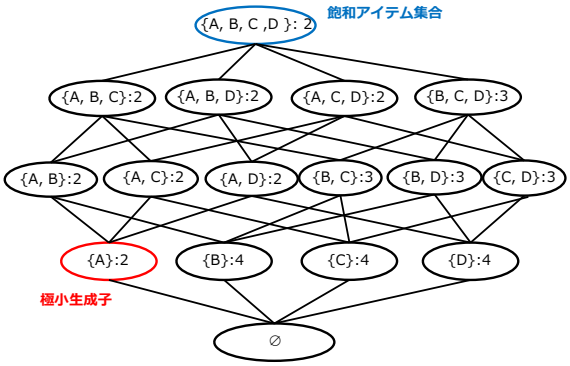

図 3: 探索空間

例 2 より，$|Y| \geq 2$ の極小生成子では差分を用いた効率的な枝刈りは行えない．そのため差分を用いた効率化では補題 1 について，$|Y| = 1$ の極小生成子のみに用いる．

ボトムアップ型のアルゴリズムを Algorithm2 に示す．

Algorithm2 では，まず要素数 1 の部分集合を生成し，飽和アイテム集合に対して極小生成子であるかを検査する．極小生成子であれば，極小生成子として追加し，飽和アイテム集合から差分をとる．差分をとった飽和アイテム集合と同じ要素数になるまで部分集合を検査し，すべての飽和アイテム集合について検査を行う．要素数 1 のアイテム集合は必ず極小生成子となるため，ボトムアップ型アルゴリズムで効率よく抽出できるのではないかと考えられる．図 2 にトランザクションデータベース例，図 3 に図 2 の飽和アイテム集合 $ABCD$ に対しての探索空間，図 4 にボトムアップ型による $ABCD$ に対しての探索空間を示す．

図 4 のような飽和アイテム集合 $ABCD$ に対して，要素数 1 の極小生成子 A が存在し，A を含むような部分集合について枝刈りを行い，探索空間を削減することができる．

4.3 深さ優先トップダウン型アルゴリズム

深さ優先トップダウン型アルゴリズムは飽和アイテム集合 X の要素数を減らしながら，左深さ優先で極小生成子の検査を行うアルゴリズムである．そのため補題 1, 2 の極小生成子についての枝刈りを行うことができる．深さ優先トップダウン型アルゴリズムを Algorithm3 に

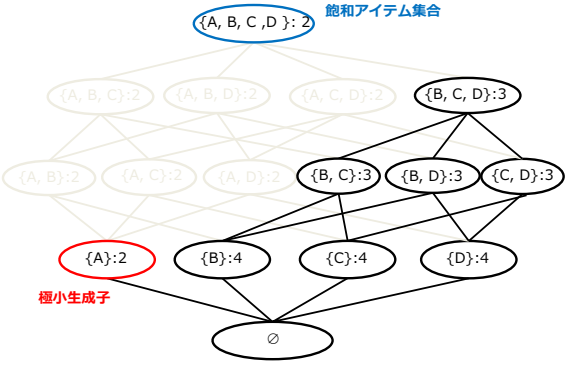

図 4: 探索空間 (ボトムアップ型)

Algorithm 2 ボトムアップ型極小生成子の抽出アルゴリズム

Input: 3 項組 $< ai, X_i, f_i >$ の族 $\mathcal{CS} = \{< 1, X_1, f_1 > , \ldots, < n, X_n, f_n >\}$．但し，各 i は識別番号，X_i は飽和アイテム集合，f_i は X_i の出現頻度である．

Output: $\mathcal{CS}$ の各飽和集合（識別番号 i）に対する極小生成子 M_j との対の集合

$\mathcal{MG} = \{< M_j, i, f_i > \mid \exists < i, X_i, f_i > \in \mathcal{CS},\ M_j \in \mathcal{MG}(X_i)\}$

$\mathcal{MG} \leftarrow \emptyset$ { $\mathcal{MG}$ は，極小生成子の族を格納する大域変数 }

$Y \leftarrow \emptyset$ { Y は，飽和アイテム集合を格納する変数 }

$\mathcal{D}$ { 垂直配置で飽和アイテム集合 $\mathcal{CS}$ を保持する }

for each $< i, X_i, f_i > \in CS$ **do**

  $Y \leftarrow X_i$ {Y は検査する飽和アイテム集合を格納する変数 }

  for each $e \in Y$ **do**

    {Y 各要素 e について検査 }

    if $f_i =$ SUPPORT(e) **then**

      {e が生成子であるかの検査 }

      $\mathcal{MG} \leftarrow \mathcal{MG} \cup < e, i, f_i >$ {$\mathcal{MG}$ に極小生成子 e を併合 }

      $Y \leftarrow Y - e$ { 飽和アイテム集合から極小生成子の差分 }

    end if

  end for

  $k \geq 2$ {k は極小生成子を検査するアイテム集合の要素数 }

  for each $|Y| \subseteq k$ **do**

    {Y の部分集合について検査を行う }

    for each $Y' s.t.\ Y' \subset Y\ and\ |Y'| = k$ **do**

      {Y の要素数 k の部分集合について検査 }

      if $f_i =$ SUPPORT(Y') **then**

        {Y' が生成子であるかの検査 }

        $\mathcal{MG} \leftarrow \mathcal{MG} \cup < Y', i, f_i >$ {$\mathcal{MG}$ に極小生成子 Y' を併合 }

      end if

    end for

    $k \leftarrow k + 1$

  end for

end for

return $(\mathcal{MG})$

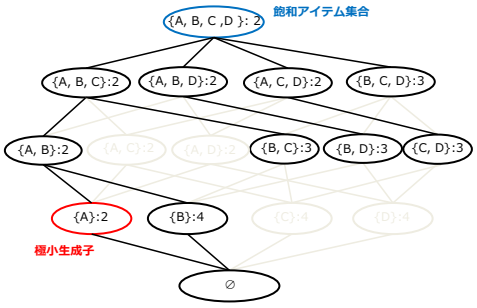

図 5: 探索空間 (トップダウン型)

TID	アイテム集合	頻度
1	ABCD	2
2	BCD	3
3	DE	4
4	D	5

図 6: 飽和アイテム集合

TID	アイテム集合
A	1
B	1, 2
C	1, 2
D	1, 2, 3, 4
E	3

図 7: 飽和アイテム集合 (垂直配置)

示す.

Algorithm3 ではまず飽和アイテム集合 Y をスタックに追加する．スタックから取り出したアイテム集合 X の $\mid X' \mid = \mid X \mid -1$ となる部分集合 X' が生成子であるかを検査し，生成子が存在しなければ取り出したアイテム集合 X を極小生成子として登録する．要素数 1 の極小生成子を見つける効率はボトムアップ型に比べ非効率であるため，効率的ではないと考えられる．図 5 に探索空間を示す.

図 5 のような飽和アイテム集合 $ABCD$ に対して，要素数 1 の極小生成子 A が存在し，まだ検査を行っていない A を含むような部分集合について枝刈りを行うことができる．また要素数 2 以上の極小生成子 BC, BD, CD が存在し，それぞれの部分集合について枝刈りを行うことができる.

4.4 頻度計算手法

極小生成子の検査には飽和アイテム集合の部分集合について頻度の計算を行う必要がある．この頻度計算処理を効率的に行うため，飽和アイテム集合から頻度を求める．アイテム集合 X の頻度を求めるには X を含む飽和集合について検査し，その中で最も高い頻度が X の頻度となる．また候補となる飽和アイテム集合を効率よく見つけるためにアイテム集合の出現する ID

Algorithm 3 深さ優先トップダウン型極小生成子の抽出アルゴリズム

Input: 3 項組 $< i, X_i, f_i >$ の族 $\mathcal{CS} = \{< 1, X_1, f_1 >, \ldots, < n, X_n, f_n >\}$．但し，各 i は識別番号，X_i は飽和アイテム集合，f_i は X_i の出現頻度である．
Output: $\mathcal{CS}$ の各飽和集合（識別番号 i）に対する極小生成子 M_j との対の集合 $\mathcal{MG} = \{< M_j, i, f_i > \mid \exists < i, X_i, f_i > \in \mathcal{CS},\ M_j \in \mathcal{MG}(X_i)\}$
$\mathcal{MG} \leftarrow \emptyset$ { $\mathcal{MG}$ は，極小生成子の族を格納する大域変数 }
$\mathcal{S} \leftarrow \emptyset$ { $\mathcal{S}$ は，stack 構造で生成子の候補となるアイテム集合の族を格納する大域変数 }
$Y \leftarrow \emptyset$ { $\mathcal{Y}$ は，S から取り出すアイテム集合を格納する }
$Z \leftarrow \emptyset$ { $\mathcal{Z}$ は，要素数 1 の極小生成子を格納する変数 }
$\mathcal{D}$ { 垂直配置で飽和アイテム集合 $\mathcal{CS}$ を保持する }
for each $< i, X_i, f_i > \in \mathcal{CS}$ **do**
  X_i を S に追加
  while each $\mathcal{S} \neq \emptyset$ **do**
    { 飽和アイテム集合 X_i の部分集合について検査 }
    $Flag \leftarrow ture$ {Y が極小生成子であるかを判別するフラグの初期化 }
    Y に S を取り出す
    if $Z \neq \emptyset$ **then**
      { 要素数 1 の極小生成子が存在するかの検査 }
      $Y \leftarrow Y - \mathcal{Z}$ {Y から要素数 1 の差分をとる }
    end if
    for each $Y's.t.\ Y' \subset Y\ and\ |Y'| = |Y| - 1$ **do**
      {Y の要素数 $|Y|$ -1 の部分集合について検査 }
      if $f_i =$ sup(Y') **then**
        {Y' が生成子であるかの検査 }
        Y' を S に追加
        $Flag \leftarrow false$ {Y' が生成子であれば $false$}
      end if
    end for
    if $Flag = ture$ **then**
      { 全ての Y' が生成子でなければ Y を登録 }
      $\mathcal{MG} \leftarrow \mathcal{MG} \cup < Y, i, f_i >$
      if Y の要素数が 1 ならば **then**
        $Z \leftarrow Z \cup Y$ {Z に要素数 1 の Y を追加 }
      end if
    end if
  end while
  $Z \leftarrow \emptyset$ {Z の初期化 }
end for
return $(\mathcal{MG})$

を格納する垂直配置と呼ばれるデータセットの表現を用いる．図 6 に飽和アイテム集合を示す．図 7 には垂直配置で表現した飽和アイテム集合を示す.

垂直配置された飽和アイテム集合から頻度を求めるためには検査したい部分集合の各アイテムが共通に出現している飽和アイテム集合を探す必要がある．例として CD というアイテム集合の頻度を求めると，それぞれのアイテムが ID1, 2 の飽和アイテム集合に含まれていることがわかる．この飽和アイテム集合の頻度を比較し，最も高いものが頻度となり，CD の頻度は 3 となる.

5 実験

実験には，FIMD Repository[?] から実データである connect, retail を用いた．データセットの詳細を表

1 に示す．また，connect は稠密 (dense) なデータセット, retail は疎 (sparse) なデータセットである．#(item) はデータセット中に含まれるアイテムの種類数を示し，#(trans.) はトランザクションの総数，ave(item) は各トランザクション中に出現するアイテムの平均数である．

表 1: 実験に使用したデータベース

データベース	#(item)	#(trans.)	ave(item)
connect	130	67,557	43.0
retail	16,470	88,162	10.3

最小支持度 ms を変化させて飽和アイテム集合からボトムアップ型，トップダウン型アルゴリズムで極小生成子を抽出した実験結果を表 2, 3 に示す．

FI，CFI，MG はそれぞれ，頻出アイテム集合，頻出飽和アイテム集合，頻出な極小生成子の総数である．なお，頻出飽和アイテム集合の抽出には，宇野らの頻出集合発見プログラム LCM ver.3 [9] を使用した．

表 2: 実験結果 (アイテム集合数)

データセット	ms	FI	CFI	MG
connect	0.85	142,271	8,255	8,460
	0.9	27,127	3,486	3,546
retail	0.0005	19,836	19,698	19,748
	0.001	7,712	7,695	7,704

表 3: 実験結果 (計測時間)

データセット	ms	ボトムアップ [s]	トップダウン [s]
connect	0.85	193.42	11.52
	0.9	53.02	3.94
retail	0.0005	65.36	6.53
	0.001	24.83	3.91

表 2 より，密なデータセットである connect については頻出アイテム集合に比べ，飽和アイテム集合が大幅に減少していることがわかる．しかし，疎なデータセットである retail についてはほとんど変化がない．密なデータセットについては，飽和アイテム集合を用いることで効率化を行うことが出来る．
表 3 より，ボトムアップ型とトップダウン型ではトップダウン型の極小生成子抽出手法の方が効率よく抽出できていることがわかる．connect, retail については飽和アイテム集合と極小生成子の数に差がないため効果的に枝刈りを行えていたと考えられる．

6 まとめ

本研究では負の相関ルールマイニングの効率化のために極小生成子を抽出する手法を検討した．極小生成子の抽出には頻出アイテム集合を圧縮した飽和アイテム集合を用いた．これにより膨大な数が存在する頻出アイテム集合を用いることなく効率的に抽出を行うことができた．

提案したボトムアップ型，深さ優先トップダウン型について，実験結果より深さ優先トップダウン型アルゴリズムがボトムアップ型より効率的であることを示した．今回利用したデータセットである connect. retail は飽和アイテム集合と極小生成子の数に差がないため，$|Y| \geq 2$ となる極小生成子の部分集合についての検査を枝刈りすることができ，効率的に抽出できたと考えられる．今後の課題として，飽和アイテム集合に対して極小生成子が多いデータセットなどを用いた実験を行い性能の評価を行うことが挙げられる．

謝辞: 本研究の一部は JSPS 科学研究費補助金 (16K00298) の援助を受けている．

参考文献

[1] C. Cornelis, P. Yan, X. Zhang, and G. Chen: Mining Positive and Negative Association Rules from Large Databases. *Proc. CIS* 2006. LNCS, Vol.4456, pp.613–618, 2006.

[2] H. Wang, X. Zhang and G. Chen: Mining a Complete Set of Both Positive and Negative Association Rules from Large Databases. *Proc. the PAKDD'08*, pp.777–784, 2008.

[3] X. Wu, C. Zhang, and S. Zhang: Efficient Mining of Both Positive and Negative Association Rules. *ACM Trans. on Information Systems*, Vol.22(3), pp.381–405, 2004.

[4] 井出典子, 岩沼宏治, 山本泰生: 負の相関ルールを抽出する高速トップダウン型アルゴリズム, 人工知能学会論文誌 29 巻 4 号, pp. 406-415 (2014).

[5] 黒岩健歩, 岩沼宏治, 山本泰生: 関連尺度に基づいた負の相関ルール抽出手法の高機能化, 第 28 回人工知能学会全国大会, 3J3-3in, (2014).

[6] M. J. Zaki: Mining Non-Redundant Association Rules. Data Mining and Knowledge Discovery, Vol.9, pp.223-248 (2004)

[7] 岩沼宏治, 佐生隼一, 山本泰生, 黒岩健歩 : 負の相関ルール集合の極小生成子に基づく圧縮表現, 情報処理学会論文誌 第 57 巻 第 8 号, pp.1,845-1,849(2016).

[8] Frequent Itemset Mining Dataset Repository, <http://fimi.ua.ac.be/>(2017-2-13).

[9] 宇野毅明, 有村博紀: LCM 公開プログラム, <http://research.nii.ac.jp/uno/dodes-j.htm>(2017-2-13).