人工知能学会研究会資料
SIG-FPAI-B404-02

# 負相関ルールマイニングの高速化と関連性尺度の導入

# Incorporating Relevance Measures into Negative Association Rule Mining and its Acceleration

黒岩健歩 [1*]　岩沼宏治 [2]　山本泰生 [2]
Yasuho Kuroiwa[1]　Koji Iwanuma[2]　Yositaka Yamamoto[1,2]

[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 Science and Media Engineering, Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi
[2] 山梨大学大学院医学工学総合研究部
[2] Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

**Abstract:**
The purpose of this study is to add new effective evaluation methods of negative rules with statistical measures. Also, we propose a branch and bound mining of top-k rules, and give a filtering method using weak relevance to negative rules mining. Besides, we propose a data structure for achieving high speed computation. Thus, we can realize efficient and effective negative rules mining. We also show some good results of experiments for evaluating our proposed method.

## 1　はじめに

本論文では, 先行研究 [井出他 14] のトップダウン型の負の相関ルール抽出アルゴリズムをもとに, 関連性尺度に基づく有効な負相関ルールの抽出および解の効果的な絞り込み手法を提案する. また, 実装に際し高速処理を可能とする実装方式について述べる.

相関ルール発見問題は, データマイニングや知識発見の代表的な問題として知られている [TSK06]. 相関ルールとは, トランザクションデータベース中で同時に発生することの多い事象同士の強い共起関係を記述したものである. データベース中でアイテム集合 $X$ が出現するトランザクションに同時にアイテム集合 $Y$ が出現することが多いことを, $X \Rightarrow Y$ と記述する. これを**正の相関ルール**と呼ぶ. 本研究で扱う**負の相関ルール**は $X \Rightarrow \neg Y$, $\neg X \Rightarrow Y$, $\neg X \Rightarrow \neg Y$ と表記され, アイテム集合の出現と非出現の関係を表す規則である. 負の相関ルールは近年研究が盛んになった分野 [WZZ04, WZC08] であり, 正の相関ルールでは発見されない知識を提供し, 有益な情報を与える. しかし, 正の相関ルールに比べて, 負の相関ルールは非出現のアイテム集合を含むためにその数は膨大となる. そのため, 負の相関ルール抽出問題は困難であることが知られている.

先行研究 [井出他 14] ではトップダウン型の負の相関ルール抽出アルゴリズムが提案された. これは, 負の相関ルールを完全かつ効率的に抽出する手法であり, 著者の知る限り最も効率的な手法であるが, 負の相関ルールの評価尺度として支持度, 確信度のみを使用している. 本研究では, ルールの評価尺度に関連性尺度を追加する. これによりルールをさらに絞り込み, より有効な負の相関ルールの抽出を行う. 更に関連研究 [亀谷他 11] の頻出パターン発見法を参考にし, 分枝限定法による探索空間の枝刈り, 弱関連性を適用した上位 k ルール抽出法を提案する. 最後に提案手法の性能評価を行った結果を示す. 本研究は, 負の相関ルールにおける評価尺度として *cosine* のみを考慮するが, これは他の評価尺度を導入する上で基盤となるものである.

*連絡先：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
〒 400-8511 山梨県甲府市武田 4-3-11
E-maill: g14mk004@yamanashi.ac.jp

## 2　準備

### 2.1　正の相関ルール

$I = \{a_1, a_2, ..., a_n\}$ を**アイテム**の全体集合とするとき, **トランザクション** $t$ をアイテムの集合 $t \subseteq I$ と定める. **トランザクションデータベース** $D$ 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 $D$ 中のトランザクション $t$ を X の**出現**と呼び, その集合を $D(X)$ と略記する. 集合 $A$ の大きさを $|A|$ と表記するとき, $X$ の $D$ 中の**支持度** $\sup(X)$ を, $\sup(X) = \frac{|D(X)|}{|D|}$ と定義する.

正の相関ルール (以下, 適宜 “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 を以下のように定義する.

$$\sup(X \Rightarrow Y) = \sup(X \cup Y), \quad \mathrm{conf}(X \Rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)}$$

**最小支持度** $ms$ と**最小確信度** $mc$ とは, ユーザが支持度と確信度に関して与える閾値である. $\sup(X) \geq ms$ を満たす $X$ を**頻出アイテム集合** と呼び, 満たさない場合は**非頻出アイテム集合**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) な正の相関ルールと呼ぶ.

### 2.2　負の相関ルールの定義

負の相関ルールについて先行研究 [井出他 14, CYZC06, SON98, WZZ04, WZC08, YBYZ02] にならい, 定義を

示す．**負の相関ルール** (negative association rule: 以下では適宜 “負ルール” と略記) は, $X$ と $Y$ を $X \cap Y = \emptyset$ であるアイテム集合とするとき, 以下のいずれかの表現である．

$X \Rightarrow \neg Y$ (**右否定形**もしくは**後件負形**),
$\neg X \Rightarrow Y$ (**左否定形**もしくは**前件負形**),
$\neg X \Rightarrow \neg Y$ (**両否定形**)

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．負アイテム集合内のアイテムは論理積で関係づけられているとする．つまり, $X \Rightarrow \neg\{a, b\}$ は $X \Rightarrow \neg(a \wedge b)$ と解釈し, 「$X$ が出現する場合, $a, b$ のどちらか一方は出現しないことが多い」を表していると考える．$X \Rightarrow (\neg a \vee \neg b)$ と変形できるので, **否定和形**と呼ぶ．否定和形の負ルールの支持度は, 下記に示すように正のアイテム集合の支持度を基に計算でき, 正の相関ルールマイニングで開発された技術を比較的容易に転用できる．以下では $C_X$ は, アイテム集合 $X$ または負アイテム集合 $\neg X$ のどちらかを表すものとする．

**定義 1 (負ルールの支持度, 確信度)** 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．

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

先に示した両否定形 $\neg X \Rightarrow \neg Y$ は, 一般に非常に数が多い．そのため, 両否定形の効率的な抽出は困難である．また, ルールとしての有用性も低いことが通常である．そのため本論文では, 右否定形 $X \Rightarrow \neg Y$ および左否定形 $\neg X \Rightarrow Y$ に焦点を絞って考察を進める．

### 2.3 有効な負の相関ルールの定義

ルール $C_X \Rightarrow C_Y$ に対して関連性尺度の値を $R(C_X \Rightarrow C_Y)$ とし, その閾値を $mr$ とする．先行研究 [WZZ04, WZC08, 井出他 14] の有効な負の相関ルールの条件に関連性尺度の検査を加える．

**定義 2 関連性尺度に基づく有効な負の相関ルール** $C_X \Rightarrow C_Y$ とは, 以下の 6 つの条件を満たすルールである．

(1) 重複性条件 $X \cap Y = \emptyset$
(2) 頻出条件 $\sup(X) \geq ms$ かつ $\sup(Y) \geq ms$
(3) 無矛盾性条件 $\sup(X \Rightarrow Y) < ms$
(4) 支持度条件 $\sup(C_X \Rightarrow C_Y) \geq ms$
(5) 確信度条件 $\mathrm{conf}(C_X \Rightarrow C_Y) \geq mc$
(6) 関連性尺度条件 $R(C_X \Rightarrow C_Y) \geq mr$

(6) は [WZZ04] でも用いられた, 関連性尺度の条件である．(3) の無矛盾性条件は正ルール $X \Rightarrow Y$ が有効である場合, 同様のアイテム集合を持つ負ルール $C_X \Rightarrow C_Y$ が同時に有効である状態を矛盾とし, 負ルールの抽出を行わない条件である．

## 3 負ルールの関連性尺度

### 3.1 関連性尺度の定義

本節では, 負ルールマイニングに導入する統計的評価尺度を定義する．以降, 統計的評価尺度を前件と後件の**関連性尺度**と呼ぶ．関連性尺度として相関ルールマイニングではよく, アイテム集合間の独立性を見る lift[TSK06] や interest[TSK06] 等の尺度がある．本研究では, 評価尺度の特性を踏まえ, *cosine*[TSK06] を使用する．*cosine* 尺度は, 正の相関ルールマイニングにおいてアイテム集合間の独立性を見る尺度であり, 高い値ほど強い正の相関を示す．ここで負ルールの評価尺度における *cosine* 尺度を考え以下のように定義する．

**定義 3 (関連性尺度: *cosine*)**

$$cosine(C_X \Rightarrow C_Y) = \frac{\sup(C_X \cup C_Y)}{\sqrt{\sup(C_X)\sup(C_Y)}}$$

この *cosine* の定義は, 正のアイテム集合と負のアイテム集合の独立性を見る．例えば, $X \Rightarrow \neg Y$ を評価する．このとき, アイテム集合 $X$ と $\neg Y$ が高い評価値を示す場合, $X$ と $Y$ には強い負の相関がある．この表現は正の相関ルールにおける表現の自然な拡張となっていることから, 負ルールにおける関連性尺度の定義として妥当なものと考える．

### 3.2 評価尺度の特性

本節では, 負相関ルールマイニングにおける評価尺度が満たすべき特性について述べる．評価尺度はそれぞれ固有の特性をもち, 正相関ルールにおいて, 次のような分割表を用いていくつかの尺度特性 [TSK06] が述べられている．まず, 相関ルールにおいて表 1 のような分割表を考える．

表 1: 分割表

| | $Y$ | $\neg Y$ | |
|---|---|---|---|
| $X$ | $f_{11}$ | $f_{10}$ | $f_{1+}$ |
| $\neg X$ | $f_{01}$ | $f_{00}$ | $f_{0+}$ |
| | $f_{+1}$ | $f_{+0}$ | $N$ |

これはアイテム集合 $X, Y$ の出現と非出現の関係を表しており, $f_{ij}$ はそれぞれの出現トランザクション数であり, $N$ はトランザクションの総数を示す．正ルールにおける尺度特性の中でも特に重要と考えられているのが, $Null$-**不変性**であり, 次のように定義される．

**定義 4 (尺度特性: $Null$-不変性)** 正ルール $X \Rightarrow Y$ について尺度 $M$ で評価する場合, $M(X \Rightarrow Y)$ が $f_{00}$ の変化に対して不変であるならば, 尺度 $M$ は $Null$-不変性を満たすという．

これは, 注目するアイテム集合以外の要因 ($f_{00}$) に対し, 不変であるという特性である．この特性を満たさない場合, $f_{00}$ の変化に対して相関性の評価が正から負に逆転してしまうことも稀ではない．正相関ルールマイニングにおいて, $Null$-不変性は相関性の評価に大きく影響するために重要な特性である．

本研究では負ルールにおいても, 同様に重要な尺度特性であると考える．ここで $Null$-不変性を拡張し, 正負ルールにおける評価尺度の特性 $f_{ij}$-**不変性**を次のように定義する．

**定義 5 (尺度特性: $f_{ij}$-不変性)** 相関ルール $C_X \Rightarrow C_Y$ について尺度 $M$ で評価する場合, $M(C_X \Rightarrow C_Y)$ が $f_{11}$ の変化に対して不変ならば $XY$-不変性, $f_{10}$ の変化に

対して不変ならば $X\overline{Y}$-不変性, $f_{01}$ の変化に対して不変ならば $\overline{X}Y$-不変性, $f_{00}$ の変化に対して不変ならば $\overline{XY}$-不変性, を満たすという.

$\overline{XY}$-不変性は上記の $Null$-不変性と同義である.

右否定形ルール $X \Rightarrow \neg Y$ は, $\overline{XY}$-不変性を満たすことが望ましい. なぜならば, $X \Rightarrow \neg Y$ は $X$ と $\neg Y$ の関係に着目している. そのため, $f_{11}$, $f_{10}$, $f_{00}$ の関係が重要であるため, 考慮していない $f_{01}$ の影響により評価値が変化すべきではないからである. 左否定形ルール $\neg X \Rightarrow Y$ についても同様な理由で, $X\overline{Y}$-不変性を満たすことが望ましい. 本研究で使用する $cosine$ 尺度はこれを共に満たす.

# 4 負ルール抽出手法

本章では, 先行研究 [井出他 14] の手法および本論文での提案手法について示す. 負相関ルールマイニングはアイテム集合の組合せ問題であるためにその解は指数的である. そこで, 負ルールの候補数を削減することで組合せ計算の効率化を図る.

## 4.1 先行研究の提案手法

本手法は, 頻出アイテム集合を節点とする図 1 のような**接尾木** [亀谷他 11] の組合せ探索により, 負ルールを抽出する. アイテムの間には適当な順序 $\prec$ を仮定し,

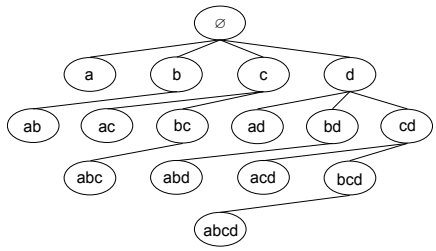


図 1: 接尾木の例

アイテム集合をアイテムの列として取り扱う. 図 1 ではアルファベット順 $a \prec b \prec c \prec d$ を仮定している. 各節点 $N_c$ の親は, 長さが 1 つだけ短い接尾辞 (suffix) をもつ節点 $N_p$ である. 子 $N_c$ と親 $N_p$ の差分アイテムは, $\prec$ 上において $N_p$ 中のアイテムより前にあるものである. 兄弟関係にある節点は $\prec$ に基づく辞書順で左から右へ並ぶ. 接尾木上で左優先深さ優先探索を行うと, 節点 $N$ を訪問する時点で $N$ の部分集合は全て訪問が完了している. これは後で示す弱関連性において包含関係の検査をする上で都合が良い. また, 負ルールは接尾木を左優先深さ優先で探索する. 先行研究 [井出他 14] で利用しているトップダウン型のアルゴリズムは, 包含関係のあるルール間の関係性の検査が容易であるために, 効率よく探索空間を削減することを可能にする. 本手法も同様にトップダウン型のアルゴリズムを採用する.

本手法では, 重複性検査と次節で示す分枝限定法による枝刈り操作を使用する. 重複性検査は定義 3 の条件（1）を保証するものであり前件, 後件のアイテム集合の独立性条件を用いた枝刈り手法である.

**命題 1 (重複性の単調性)** $X \cap Y \neq \emptyset$ ならば, $Y \subset Y'$ である $Y'$ に対して $X \cap Y' \neq \emptyset$ が必ず成り立つ.

$X$ と $Y$ の重複性を検査し, 重複部分があれば子節点を枝刈りする. この枝刈は命題 1 よりその安全性が保証される.

## 4.2 分枝限定法

$cosine$ を含め, 関連性尺度は逆単調性を満たさないものが多い. そのような場合, 不用意に探索空間の枝刈りを行うと, 有用なルールを見落としてしまう恐れがある. そこで分枝限定法を用いた枝刈りを使用する. [井出他 14] では, 確信度について下記の上界関数を定義して分枝限定操作を実現している.

**定義 6 (上界関数: 確信度)**

$$\overline{\mathrm{conf}}(\neg X \Rightarrow Y) = \frac{\sup(Y)}{1 - \sup(X)}$$

同様に, 本研究で定義した $cosine$ についても, 負ルールの包含関係に関して, 逆単調性を満たす上界関数を右否定形, 左否定形についてそれぞれ以下のように定義できる.

**定義 7 (上界関数: 関連性尺度)**

$$\overline{cosine_R}(X \Rightarrow \neg Y) = \sqrt{\frac{\sup(X)}{1 - \sup(Y)}}$$

$$\overline{cosine_L}(\neg X \Rightarrow Y) = \sqrt{\frac{\sup(Y)}{1 - \sup(X)}}$$

このとき, 以下が成り立つ.

**命題 2 (上界関数の逆単調性)** $X$, $X'$, $Y$, $Y'$ をアイテム集合とし, $X \subset X'$, $Y \subset Y'$ と仮定する. $R$ を関連性尺度の一般表記とするとき, 以下が成り立つ.

1. $\overline{R}(C_X \Rightarrow C_Y) \geq R(C_X \Rightarrow C_Y)$
2. $\overline{R}(C_X \Rightarrow C_Y) \geq \overline{R}(C_{X'} \Rightarrow C_{Y'})$

命題 2 より, 上界関数は評価尺度の上界を成し, 逆単調性が成り立つことが保証されるため, 接尾木上での枝刈り操作に用いることができる. 即ち, 関連尺度に注目すると, 閾値 $mr$ に対して $\overline{R}(C_X \Rightarrow C_Y) < mr$ ならば, $C_X \Rightarrow C_Y$ 自身および $X, Y$ のアイテム集合を拡張したルール $C_{X'} \Rightarrow C_{Y'}$ は, 全て閾値を満たさないことが保証される. そのため, 即座に枝刈りを行うことができる.

## 4.3 上位 k 負ルール抽出

相関ルールを抽出する際, 適当な閾値を設定されないと有益でないルールが大量に抽出される. それに伴い計算コストも増加する. この問題を解決する手法として, **top-k 手法** [HWLT02] がある. これは, ユーザがルール数 k を指定し抽出する手法である. この手法の利点は, 評価尺度の閾値が, データベースに依存し自動調整されることにある. また, [亀谷他 11] は関連性尺度に基づく top- k 手法を示した. この手法をもとに負ルールマイニングにおいて関連性尺度に基づく top-k 負ルール抽出を適用する.

本手法では左否定形, 右否定形に対し, それぞれ top-k 個抽出する. ここで右否定形に注目し, その閾値を $mr_R$ とする. 上位 k 個の候補リストのうち, k 番目のルールの関連度を $R^k$ とする. このとき $mr_R$ は, 常に $R^k$ の値で更新できる. なぜなら, 関連度が $R^k$ 未満である場合, 最終的に有効な上位 k 個のルールとして抽出されることがない. そのため, $R^k$ を閾値として更新できる (閾値上昇法 [亀谷他 11]）.

### 4.4 負ルールにおける弱関連性

相関ルールマイニングでは, 一般に評価尺度が高いルールにつられて同じアイテム集合を含むルールの評価値も高くなる. そのため, top-k 手法では, 類似したルールが抽出の多くを占めることが経験的に知られている. そこで, 似たルールを冗長と判断し, 非冗長なルールのみを抽出する手法を適用する. 以下では [亀谷他 11] によって提案された, パターン間の「より弱い (weaker)」という関係の自然な拡張として, 負ルールにおける弱関連性を以下のように定義する. ただし, $C_X \Rightarrow C_Y$ に対して $X' \subseteq X$, $Y' \subseteq Y$ なる $C_{X'} \Rightarrow C_{Y'}$ を**部分ルール**と呼ぶ.

**定義 8 (負ルールの弱関連性)** 右否定形のルール *1:* $X1 \Rightarrow \neg Y1$ と, その部分ルールであるルール *2:* $X2 \Rightarrow \neg Y2$ において, $R$(ルール 2) $\geq R$(ルール 1) ならば, ルール 1 は ルール 2 より**弱い**.

左否定形のルール *3:* $\neg X3 \Rightarrow Y3$ と, その部分ルールであるルール *4:* $\neg X4 \Rightarrow Y4$ において, $R$(ルール 3) $\geq R$(ルール 4) ならば, ルール 4 は ルール 3 より**弱い**.

弱関連性の定義から, あるルールについて弱い関係が成り立つ場合は, 抽出しない. よって右否定形は負ルールの弱関連性に関して極小なルール, 左否定形は極大なルールを有効なルールとして残す.

接尾木上でルールを探索する場合, 右否定形ルール $C_X \Rightarrow C_Y$ について探索する場合, その部分ルールに対して全て探索済みである. そのため上位 k ルール抽出において弱関連性の検査は, 候補リストのルールのみと比較するだけでよい.

### 4.5 負ルール抽出アルゴリズム

本節では, 提案するアルゴリズムの概要を示す. 頻出アイテム集合の抽出には LCMver.2[宇野] を使用する. 以下では要素数 $k$ の頻出アイテム集合の集合を $\text{FISS}^k$ と表記する. $mr_R$, $mr_L$ をそれぞれ左否定形, 右否定形の関連尺度の閾値とし, 負ルール抽出アルゴリズム [1] の概要を以下に示す.

**Input:** データセット $D$, $ms$, $mc$, 抽出ルール数 k
**Output:** 上位 k ルールの右否定形 $RL$, 左否定形 $LL$

上界関数条件の真偽をみる $F_L$, $F_R$;
$D$ から LCM によって, $\text{FISS}^1$, ..., $\text{FISS}^N$ を抽出し, FISS を要素とする接尾木を構築 (1, ...,N は接尾木を左優先深さ優先で探索した順序);
**for** $i = 1$ to $N$ **do**
  $X := \text{FISS}^i$;
  **for** $j = 1$ to $N$ **do**
    $F_L := False$, $F_R := False$;
    $Y := \text{FISS}^j$;
    **if** $X \cap Y \neq \emptyset$ **then**
      重複性条件により, $Y$ の子孫節点を枝刈り;
    **else if** $\sup(X \Rightarrow Y) < ms$ and $\sup(X \Rightarrow \neg Y) \geq ms$ **then**
      **if** $\overline{\text{conf}}(\neg Y \Rightarrow X) \geq mc$ and $\overline{R}(\neg Y \Rightarrow X) \geq mr_L$ **then**
        $F_L := True$;
        $\neg Y \Rightarrow X$ について, 確信度, 関連尺度, 弱関連性を検査し, 合格したら左否定形の候補リスト $LL$ に追加;
      **end if**
      **if** $\overline{R}(X \Rightarrow \neg Y) \geq mr_R$ **then**
        $F_R := True$;
        $X \Rightarrow \neg Y$ について, 確信度, 関連尺度, 弱関連性を検査し, 合格したら右否定形の候補リスト $RL$ に追加;
      **end if**
      **if** $F_L == False$ and $F_R == False$ **then**
        上界関数により, $Y$ の子孫節点を枝刈り;
      **end if**
    **end if**
  **end for**
**end for**

| TID | アイテム集合 |
|---|---|
| 1 | A, B, E, F, G, I |
| 2 | B, C, D, E |
| 3 | A, B, G, H, I |
| 4 | A, G, I |
| 5 | B, C, G, I |
| 6 | B, G, I |

図 2: トランザクションデータベース

## 5 実装の高速化における工夫

本章では提案アルゴリズムの実装にあたり, 計算手法と親和性の高いデータ構造と計算を効率化する工夫について述べる.

### 5.1 データ構造

本手法を実装するにあたってダウンプロジェクション [宇野他 08] 等の計算手法と親和性の高いデータ構造を実装した.

図 2 のデータベースで $ms = 0.5$ のとき, 図 3 のような頻出アイテム集合のデータ構造を作成する. FIS は頻出アイテム集合のアイテム ID を格納し, TID はアイテム集合の出現するトランザクションの ID を格納する. このような TID を垂直配置と呼び, 頻度計算にあたってデータベースの再スキャンを行わずに頻度を計算することが可能となる.

### 5.2 非頻出アイテム集合の頻度の記憶

図 3 のようなデータ構造を利用し, 頻度計算を行う. たとえば $X1 \Rightarrow \neg Y1$ についてルールの有効性検査を行う場合, $X$, $Y$, $X \cap Y$ の頻度が必要である. $X$, $Y$ は頻出アイテム集合であるため, データ構造を参照することで頻度を得ることができる. $X \cap Y$ の頻度は, 非頻出アイテム集合であるため $X$ と $Y$ の和集合をとることで計算される. また, アイテム集合 $\{A, B, C\}$ について, $\{A, B\} \cap \{C\}$, $\{A, C\} \cap \{B\}$, $\{B, C\} \cap \{A\}$ の 3 パターンがあるように, 同一の非頻出アイテム集合について複数回計算している. この計算は, ルールの候補について有効性検査する度に行い, 全体の計算に占める計算時間の割合も非常に高い.

そこで非頻出アイテム集合の情報を蓄積し, 再計算によるオーバーヘッドを削減することを考える. 非頻出アイテム集合の数は非常に多いが, 記憶すべきデータは頻度のスカラー値のみであるため, 頻出アイテム集合に比べれば少量である. 非頻出アイテム集合のデータ構造を図 4 に示す.

INFIS は, 非頻出アイテム集合のアイテム ID が格納される. $freq$ はアイテム集合の頻度を示す. また, 提

[1] 支持度について $X \Rightarrow \neg Y$ のみを検査している. これは $X \Rightarrow \neg Y$ と $\neg Y \Rightarrow X$ には支持度の同値性 [井出他 14] が成り立つために, $X \Rightarrow \neg Y$ の支持度の有効性のみを検査するだけでよい.

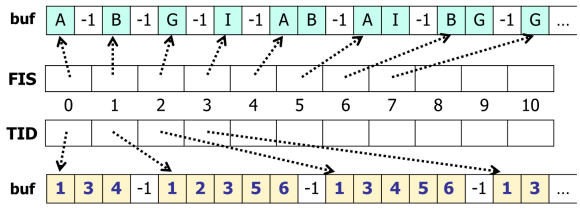


図 3: 頻出アイテム集合のデータ構造

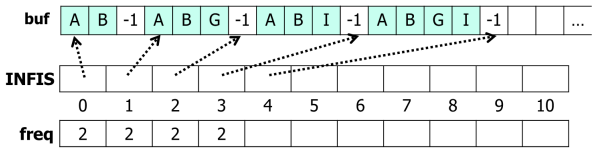


図 4: 非頻出アイテム集合のデータ構造

案手法の枝刈りにより探索空間を削減することでその数も抑え込めることが予想される.

# 6 評価実験

本研究の提案手法のアルゴリズムを実装し, 提案手法の効果を測定した実験の結果および考察を示す. 実験には Frequent Itemset Mining Dataset Repository [FIMI] から 4 種のデータセットを使用した. 各データセットの詳細を表 2 に示す.

表 2: 実験に使用したデータ

| データセット | #(item) | #(trans.) | ave(item) |
|---|---|---|---|
| T10I4D100K | 870 | 100,000 | 10.1 |
| retail | 16,470 | 88,162 | 10.3 |
| mushroom | 119 | 8,124 | 23 |
| connect | 130 | 67,557 | 43 |

そのうち T10I4D100K, retail は疎 (sparse) なデータセット, mushroom, connect は調密 (dence) 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.) はデータセット中のトランザクションの総数, ave(item) は 1 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. #(FIS) は頻出アイテム集合の総数である.

## 6.1 トップダウン探索における探索効率

相関ルールの候補に対して支持度の検査を行った頻出アイテム集合の対 $\langle X,\ Y\rangle$ を, 以下では sup 検査対と呼ぶ. また, 重複性検査, 分枝限定法の 2 つの枝刈り手法により探索空間をどの程度削減したかを示す**削減率**を以下のような式から算出される.

$$\text{削減率} \quad = \quad 1 - \frac{\text{sup 検査対の総数}}{\text{直積 FISS}^2\text{の要素数}} \quad (\%)$$

評価尺度として *cosine* を加え, 最小確信度 mc を 0.5, 抽出ルール数を 100 に固定し, 最小支持度 ms の値を変化させて負ルールを抽出した実験結果を表 3 に示す. 実験結果の $\#(FI)$ は頻出アイテムの数であり, 和集合計算は, 頻出アイテム集合の頻度を計算した回数を表す.

表 3 より, 削減率は調密なデータセットの方が高い結果となった. 特に connect では, 90 %以上の探索空間に対して枝刈りされている. 実行時間についても connect は最も速い結果となった. これは, 調密なデータセットと疎なデータセットの頻出アイテム集合に注目した時, 調密なデータセットは頻出アイテム集合に対し, 頻出アイテムの割合が小さいことが関係していると考える. 本手法はトップダウン探索であり, アイテム集合の包含関係により探索空間を削減する. そのため, 1 つのアイテム集合を包含するアイテム集合の多い調密なデータセットの方が削減率が高くなったと考える. 本手法は特に調密なデータセットに対して, 効果的な手法であると考える.

## 6.2 top-k ルール抽出の速度比

top-k 手法はしきい値を自動調節する手法である. top-k 手法で抽出した k 番目の負ルールの閾値は, k 個の負ルールを抽出する上での最適な閾値となる. top-k 手法は最適値を設定し抽出する場合に比べると当然遅くなってしまうが, 実際に k 番目の負ルールの閾値を得ることは難しい. [亀谷他 11] ではその有効性が示されているが, 負ルールの抽出において有効であるか, 実験を行う. top-k 手法と k 番目の負ルールの閾値を入力した場合を比較し, その結果を表 4 に示す.

表 4: top-k 抽出の効果比較

| データセット | 閾値の設定 | sup 検査対 | 実行時間 (sec) |
|---|---|---|---|
| retail[1] | 最適値 | 2,830,442 | 6.29 |
| | $top-k$ 手法 | 2,830,442 | 6.33 |
| mushroom[1] | 最適値 | 1,246,828 | 2.69 |
| | $top-k$ 手法 | 1,257,488 | 2.73 |

1 retail: ms=0.002, mushroom: ms=0.3

実験結果より, top-k 手法を用いた場合も検査するルールの数もほとんど変わらず, 実行時間に関しても少しの増加で抑えられている. このことから, ルール探索の早い段階で閾値が最適値に収束していることが考えられ, 負ルールマイニングにおいても有効な手法である.

## 6.3 弱関連性の適用による効果

弱関連性は冗長なルールを削減し, 有効な負ルールのみを抽出する. 弱関連性を使用する場合, 計算コストは余計にかかってしまう. そこで, 弱関連性を使用する場合としない場合の処理について比較し, その抽出結果を表 5 に示す.

表 5: 弱関連性の効果比較

| データセット | 弱関連性 | sup 検査対 | 実行時間 (sec) |
|---|---|---|---|
| retail[1] | 無 | 2,830,442 | 6.29 |
| | 有 | 2,830,442 | 6.33 |
| mushroom[1] | 無 | 1,226,468 | 2.68 |
| | 有 | 1,257,488 | 2.73 |

1 retail: ms=0.002, mushroom: ms=0.3

弱関連性の検査を行った場合と行わない場合と比較して, ほとんど変わらない実行時間で計算されていることがわかる. 少量の時間増加で, 有効な負ルールに絞り込めるため, 有用な手法であると考える.

## 6.4 実装における高速化の工夫

第 5 章で述べたように, 頻度の計算が全体の処理中で負荷が高い処理である. そこでアイテム集合の頻度

表 3: *cosine* に基づく上位 k 負ルール抽出の実験結果

| データセット | ms | #(FIS) | #(FI) | sup 検査対 | 和集合計算 | 削減率 (%) | 負ルール探索 (sec) | 実行時間 (sec) |
|---|---|---|---|---|---|---|---|---|
| T10I4D100K | 0.005 | 1,073 | 569 | 635,497 | 453,783 | 44.8 | 6.53 | 6.75 |
| | 0.010 | 385 | 375 | 144,188 | 72,861 | 2.7 | 1.88 | 2.09 |
| | 0.015 | 237 | 237 | 55,932 | 27,752 | 0.4 | 0.93 | 1.11 |
| retail | 0.002 | 2,715 | 963 | 2,830,442 | 1,093,783 | 61.6 | 6.31 | 6.53 |
| | 0.003 | 1,409 | 521 | 810,742 | 415,185 | 59.1 | 2.81 | 2.96 |
| | 0.004 | 837 | 320 | 296677 | 174,171 | 57.6 | 1.43 | 1.61 |
| mushroom | 0.30 | 2,735 | 28 | 1,257,488 | 82,403 | 83.1 | 2.76 | 2.78 |
| | 0.35 | 1,189 | 24 | 345,441 | 27,035 | 75.5 | 0.99 | 1.01 |
| | 0.40 | 565 | 21 | 94,018 | 9,827 | 70.5 | 0.39 | 0.41 |
| connect | 0.94 | 4,227 | 17 | 740,820 | 22,304 | 95.8 | 0.36 | 0.56 |
| | 0.95 | 2,205 | 17 | 306,273 | 14,987 | 93.7 | 0.15 | 0.28 |
| | 0.96 | 1,031 | 15 | 95,483 | 6,102 | 91.0 | 0.05 | 0.17 |

つまり和集合の計算について, 非頻出アイテム集合の計算結果を記憶した場合の効果を比較する. 実験結果を表 6 に示す.

表 6: 非頻出アイテム集合を記憶する場合の抽出結果

| データセット | 記憶データ | 和集合計算 | 実行時間 (sec) |
|---|---|---|---|
| retail[1] | FIS | 2,823,759 | 12.53 |
| | FIS+INFIS | 1,093,783 | 6.33 |
| mushroom[1] | FIS | 1,263,611 | 26.95 |
| | FIS+INFIS | 82,403 | 2.73 |

1 retail: ms=0.002, mushroom: ms=0.3

図 6 の和集合計算回数とは, 非頻出アイテム集合の頻度を計算した回数である. 和集合計算回数は, 最大で sup 検査対と同じ回数だけ試行される. $FIS$ は非頻出アイテム集合は記憶しない場合であり, $FIS+INFIS$ は非頻出アイテム集合の頻度を記憶する場合である. 実験結果より, retail, mushroom 共に和集合の計算回数は減少し, 実行時間に関しても減少している. 和集合の計算は全体の処理の中でも特に負荷が高い. そのため計算結果を記憶することは, 大量な非頻出アイテム集合の情報を記憶しなくてはならないが, 有効であることがわかる.

## 7 おわりに

先行研究 [井出他 14] で提案された負の相関ルール抽出アルゴリズムをもとに, 関連性尺度の導入および負の相関ルールを効率的に抽出する手法を提案した. 有効な負ルール抽出では, 関連性尺度を用いることで負ルールを絞り込み, 抽出を行う. また, 効率的な抽出を実現するため, 分枝限定法や top-k 抽出法を用い, 検査する負ルールの候補の削減を行った. 実装に伴い, 有効なデータ構造の考案や, 計算の効率化による高速に負相関ルールマイニングの実装方式の提案をおこなった.

## 謝辞

本研究は一部, ISPS 科学研究費補助金 (25330256) および JST さきがけの支援を受けている.

## 参考文献

[井出他 14] 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, 人工知能学会研究会資料, SIG-FPAI, B303, pp.7-12, (2014).

[CYZC06] Cornelis, C., Yan, P., Zhang, X. and Chen, G.: Mining Positive and Negative Association Rules from Large Databases. *Proc. CIS* 2006. LNCS, Vol.4456, pp.613–618, (2006).

[SON98] Savasere, A., Omiecinski, E. and Navathe, S.: Mining for Strong Negative Associations in a Large Database of Customer Transactions. *Proc. Intl. Conf. on Data Engineering*, pp.494–502, (1998).

[WZC08] Wang, H., Zhang, X. and Chen, G.:Mining a Complete Set of Both Positive and Negative Association Rules from Large Databases. *Proc. the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining (PAKDD'08)*, pp.777–784, (2008).

[WZZ04] Wu, X., Zhang, C. and Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules. *ACM Trans. on Information Systems*, Vol.22(3), pp.381–405, (2004).

[YBYZ02] Yuan, X., Buckles, B. P. and Yuan, Z. and Zhang, J.: Mining Negative Association Rules.Proc. 7th Intl. Symp. on Computers and Communication, pp.623–629, (2002).

[亀谷他 11] 亀谷由隆, 佐藤泰介: 最小サポート上昇法に基づく上位 k 関連パターン発見, SIG-DOCMAS, B101, pp.(2-24)–(2-32) (2011).

[HWLT02] Han, J., Wang, J., Lu, Y. and Tzvetkov, P.:Mining top- K frequent closed patterns without minimum support, In Proc. of the 2002 IEEE Int'l Conf. on Data Mining(ICDM-02), pp. 211–218, (2002).

[TSK06] Tan, P., Steinbach, M. and Kumar, V.: Introduction to Data Mining, Addison Wesley (2006).

[宇野] 宇野毅明: 宇野毅明と有村博紀による公開プログラム（コード）, <http://research.nii.ac.jp/ uno/codes-j.htm> (2014-3-10).

[宇野他 08] 宇野毅明, 有村博紀: 頻出パターン発見アルゴリズム入門-アイテム集合からグラフまで-, 人工知能学会全国大会論文集, 22nd, 3M1-01 (2008).

[FIMI] Frequent Itemset Mining Dataset Repository, URL: http://fimi.ua.ac.be/ (2014-2-26).