2C1-4

極小性を用いた負の相関ルールの効率的な抽出法

Efficient Mining of Negative Association Rules using minimality

井出典子[*1] 岩沼宏治[*2] 山本泰生[*2]
Noriko IDE Koji IWANUMA Yoshitaka YAMAMOTO

[*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
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
Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

Negative association rules represent some relationships between presence and absence of itemsets, or between absence and absence of itemsets. In this paper, to suppress the formation of explicit negative first event, do a significant reduction of the search space. Negative association rules have certain minimality. Rules that are not minimum is redundant, then can be deleted without loss of information. This paper suggests that to reduce the extraction of such redundant rules, to extract negative association rules efficiently.

1. はじめに

本研究では，トランザクションデータベース中から有用な負の相関ルールの完全な抽出を行うことを目的として, 抽出アルゴリズムの提案を行う．また，提案手法の有用性を検証するために，抽出アルゴリズムを実装し，評価実験を行ったので，その結果を報告する．

相関ルール [1] とは, トランザクションデータベース内で同時に発生することの多い事象同士を相関の強い関係として記述したものであり, マーケットバスケット分析でよく利用されている．例えばデータベース中でアイテム集合 X がトランザクション中に出現し, 同時にアイテム集合 Y もトランザクション中に出現するとき, $X \Rightarrow Y$ と記述する．このような $X \Rightarrow Y$ が正の相関ルールであり, アイテム集合の出現の関係を表している．

一方, 本研究で扱う負の相関ルールは, ある事象が発生した際に別の事象が**発生しない**現象を記述したもので, 近年研究が盛んになった分野である．負の相関ルールは $\neg X \Rightarrow Y$, $X \Rightarrow \neg Y$, $\neg X \Rightarrow \neg Y$ という形で記述される．負の相関ルールはトランザクションデータベース中で同時に出現しないアイテム集合の関係を表している．既存手法 [1][2][3] では負の相関ルール抽出の際に, 非頻出な負のアイテム集合を明示的に生成した上で探索を行っていた．しかし, 負のアイテム集合も含めて探索を行うと, 探索空間が膨大となってしまう．また先行研究 [1] の手法では, 有効な負の相関ルールの可能性があるアイテム集合まで枝刈りしてしまうため, 抽出が完全とは言えない．

そこで, 本稿では負の相関ルールが持つ極小性を利用して, 完全性を保ちつつ枝刈りの効率化を図る．

本稿の構成は以下の通りである．第 2 章で本研究で用いる相関ルールと評価尺度を説明する．第 3 章では極小性を用いる利点と提案する有効な負の相関ルールの定義について述べる．第 4 章ではその定義を利用した抽出アルゴリズムについて述べる．第 5 章で実験とその考察を示す．第 6 章はまとめである．

連絡先: 井出典子
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
g12mk001@yamanashi.ac.jp

2. 準備

2.1 相関ルールの定義

$I = \{a_1, a_2, ..., a_n\}$ をアイテムの集合とする．トランザクション T はアイテムの集合である ($T \subseteq I$)．トランザクションデータベース D はトランザクションの集合である．T とアイテム集合 X に関して $X \subseteq T$ が成り立つとき, T は X を含むという．相関ルールとは $X \subseteq I, Y \subseteq I, X \cap Y = \emptyset$ であるような任意のアイテム集合 X, Y を使って作られる $X \Rightarrow Y$ という表現のことである．相関ルール $X \Rightarrow Y$ の左辺 X を前件, 右辺 Y を後件と呼ぶ [7].

本稿で扱う負の相関ルールの定義を示す．既存研究 [2] では負の相関ルールの概念には 2 つの形式が存在すると述べられている．1 つはアイテム集合内のアイテムの関係が **or** で表されるものである．この形式の負の相関ルール $X \Rightarrow \neg(a \vee b)$ は $X \Rightarrow (\neg a \wedge \neg b)$ に置き換えることができる．これは「アイテム集合 X が出現すると, アイテム a, b の両方が現れない」ということを表す．

もう 1 つはアイテム集合内のアイテムの関係が **and** で表されるものである．この形式の負の相関ルール $X \Rightarrow \neg(a \wedge b)$ は, $X \Rightarrow (\neg a \vee \neg b)$ に置き換えることができる．これは「アイテム集合 X が出現すると, アイテム a, b のどちらか一方は出現しない」ということを表す．正の相関ルールではアイテム集合内のアイテムは **and** で関係付けられているので, この形式のほうが負の相関ルールとしては自然である．

本稿では後者の **and** で繋がった形式の相関ルールについて扱う．

2.2 評価尺度

D 中の全トランザクションに対する $X \cup Y$ の出現割合が $s\%$ であるとき, 「$X \Rightarrow Y$ は D において $s\%$ の支持度 ($support$) をもつ」といい, $supp(X \Rightarrow Y) = s$ と表す．また D 中の X を含むトランザクションのうち, Y を含むトランザクションの出現割合が $c\%$ であるとき, 「$X \Rightarrow Y$ は D において $c\%$ の確信度 ($confidence$) で成立している」といい, $conf(X \Rightarrow Y) = c$ と表す．負の相関ルールの支持度と確信度を文献 [1] に従い以

下のように定義する.

定義 1 X と Y をアイテム集合とし, アイテム集合 C_1 と C_2 をそれぞれ $C_1 \in \{X, \neg X\}, C_2 \in \{Y, \neg Y\}$ とする. このとき支持度 $supp$ と確信度 $conf$ を以下のように定める.

$$supp(\neg X) = 1 - supp(X)$$
$$supp(X \Rightarrow \neg Y) = supp(X) - supp(X \cup Y)$$
$$supp(\neg X \Rightarrow Y) = supp(Y) - supp(X \cup Y)$$
$$supp(\neg X \Rightarrow \neg Y) = 1 - supp(X) - supp(Y) + supp(X \cup Y)$$
$$conf(C_1 \Rightarrow C_2) = \frac{supp(C_1 \Rightarrow C_2)}{supp(C_1)}$$

この 2 つの尺度とユーザーから与えられた与えられた閾値を比べ, 閾値を越える相関ルールを有効として抽出することが一般的な相関ルールの抽出問題である.

3. 有効な負の相関ルールの定義

既存手法における有効な負の相関ルールの定義の問題点と, その問題点を解決する新たな定義を提案する.

3.1 既存手法の問題点

先行研究 [1] では支持度と確信度に加え, 以下で定義する興味度 ($interest$) を用いて枝刈りを行う.

$$interest(X, Y) = |supp(X \cup Y) - supp(X)supp(Y)|$$

興味度は X と Y の独立性を示す尺度で, 多くの文献でよく使われる. 先行研究 [1] で定義された**有効**な正または負の相関ルール $C_1 \Rightarrow C_2$ の定義は以下の通りである.

定義 2 ユーザーから与えられた支持度の閾値を ms, 確信度の閾値を mc, 興味度の閾値を mi とするとき, 以下の条件を満たす $C_1 \Rightarrow C_2$ を, 有効な正または負の相関ルールと定める. ただし $C_1 \in \{X, \neg X\}, C_2 \in \{Y, \neg Y\}$ である.

1. $X \cap Y = \emptyset$
2. $supp(C_1 \Rightarrow C_2) \geq ms$
3. $supp(X) \geq ms \wedge supp(Y) \geq ms$
4. $conf(C_1 \Rightarrow C_2) \geq mc$
5. $interest(C_1, C_2) \geq mi$

以上の定義を用い, 後述するアプリオリに準拠するアルゴリズムで有効な正または負の相関ルールを抽出する. ここで注意すべきは, 確信度と興味度は逆単調性を満たさないため, ルールの生成過程で枝刈りに用いると完全性が失われてしまう点である [2]. そこで先行研究 [2] では, この問題を解決するためにルール生成途中の枝刈りから確信度 (定義 2.4) と興味度 (定義 2.5) を削除し, 新たに負の相関ルールの**極小性**を以下のように定義して, 枝刈りを行っている [2].

定義 3 以下のいずれかを満たす $C_1 \Rightarrow C_2$ を極小なルールと呼ぶ.

1. $C_1 = \neg X$ のとき, $supp(\neg X' \Rightarrow C_2) \geq ms$ を満たすような $X' \subset X$ は存在しない.
2. $C_2 = \neg Y$ のとき, $supp(C_1 \Rightarrow \neg Y') \geq ms$ を満たすような $Y' \subset Y$ は存在しない.

アイテム集合 X, X' が $X \subset X'$ となるなら, $supp(\neg X') \geq supp(\neg X)$ となる. よって $supp(\neg X \Rightarrow C_2) \geq ms$ であれば $supp(\neg X' \Rightarrow C_2) \geq ms$ となる. X を拡張した X' を含む負の相関ルールは必ず頻出となる. これら $\neg X' \Rightarrow C_2$ という相関ルールは冗長と考えられ, 極小な負の相関ルールだけを抽出する. 全ての候補を抽出してから確信度等の逆単調性を満たさない評価尺度で評価を行い, 最終的に有効な負の相関ルールを抽出する.

3.2 有効な負の相関ルールの定義

本稿で提案する有効な負の相関ルールの定義は以下の通りである.

定義 4 ユーザーから与えられた支持度の閾値を ms, 確信度の閾値を mc とする. 以下の条件を満たす $C_1 \Rightarrow C_2$ を有効な負の相関ルールとして定める. ただし $C_1 \in \{X, \neg X\}, C_2 \in \{Y, \neg Y\}$ である.

1. $X \cap Y = \emptyset$
2. $supp(X \Rightarrow Y) < ms$
3. $supp(C_1 \Rightarrow C_2) \geq ms$
4. $supp(X) \geq ms \wedge supp(Y) \geq ms$
5. $conf(C_1 \Rightarrow C_2) \geq mc$
6. (a) $C_1 = \neg X$ のとき, $supp(\neg X' \Rightarrow C_2) \geq ms$ を満たすような $X' \subset X$ は存在しない.
   (b) $C_2 = \neg Y$ のとき, $supp(C_1 \Rightarrow \neg Y') \geq ms$ を満たすような $Y' \subset Y$ は存在しない.

定義 2 と比較すると, 興味度を示す条件 5 の代わりに極小性を要求する条件 6 が入っている. 更に $C_1 \Rightarrow C_2$ の骨格となる正の相関ルール $X \Rightarrow Y$ に対して条件 2 を課している. これは抽出する正負のルールの矛盾, 即ち $X \Rightarrow Y$ と $X \Rightarrow \neg Y$ の同時抽出を回避するためである.

4. 抽出アルゴリズム

既存手法における有効な負の相関ルール抽出アルゴリズムの問題点と, その問題点を解決する新たな抽出アルゴリズムを提案する.

4.1 既存手法とその問題点

既存手法 [1][2][3] ではいずれも, $supp(X) < ms$ かつ $|X| \geq 2$ となるアイテム集合 X(以下, 台集合と呼ぶ) を生成し, 台集合を $X_1 \cup X_2 = X$ となる X_1 と X_2 に分解し, 組合わせた相関ルールが有効であるかを調べていた. 先行研究 [1] のアルゴリズムを以下に示す.

入力はトランザクションデータベース D, 支持度の閾値 ms, 確信度の閾値 mc, 興味度の閾値 mi を与える. 出力は有効な正負の相関ルールとなる.

1:頻出 1-アイテム集合を生成し, L_1 に代入
2:**for**$(k = 2; (L_{k-1} \neq \emptyset); k++)$ **do**
3: Tem_k に
$\{\{x_1, ..., x_{k-2}, x_{k-1}, x_k\} \mid \{x_1, ..., x_{k-2}, x_{k-1}\} \in L_{k-1} \wedge$

$\{x_1, ..., x_{k-2}, x_k\} \in L_{k-1}\}$ を代入
4: L_k に $\{c \mid c \in Tem_k \wedge (supp(c) \geq ms)\}$ を代入
5: N_k に $Tem_k - L_k$ を代入
6: L_k から有効な正の相関ルールの条件に満たない アイテム集合を削除し, 残りを正の相関ルールの台集合の集合 PL に代入
7: N_k から有効な負の相関ルールの条件に満たない アイテム集合を削除し, 残りを負の相関ルールの台集合の集合 NL に代入
end for
8:PL と NL に含まれる各台集合 X を $X_1 \cup X_2 = X$ となる X_1 と X_2 に分解し, 確信度が閾値以上であるルールを抽出

ここで注意すべきは, 定義 4 の (2)$supp(X \Rightarrow Y) < ms$ が無い場合, $X \Rightarrow Y$, $X \Rightarrow \neg Y$ の両方のルールが抽出される可能性があるが, 先行研究ではこれをアルゴリズムの手順 5 で解消している.

しかしこの手法には問題点がある. まず負の相関ルールの台集合の数が, 正の相関ルールの台集合と比べて非常に多い.

次に手順 8 の台集合を X_1 と X_2 に分解して調べる際, 分割可能な組合わせの数が多い. 台集合に含まれるアイテムの数を n とすると, 台集合あたりの分割可能な組み合わせの数は次式で求められる.

$$3 \times (2^n - 2) \tag{1}$$

式 1 より分割可能な組み合わせの数は, 台集合に含まれるアイテムの数が増えると指数的に増加することが言える.

また, 台集合を分解して組み合わせる際に, 定義 2 の (3), あるいは定義 4 の (4) にある前件と後件の支持度がそれぞれ閾値以上であるという条件を満たさないような X_1 と X_2 の組み合わせも調べる必要があり, 効率が悪い.

そこで本稿では, 最初に $supp(X) \geq ms$ となるアイテム集合 X を全て生成し, その組み合わせだけで負の相関ルールを生成する. これにより, 組合わせの数は最大でも $O(|n|^2)$ となる. 既存手法と異なり, 負の相関ルールの台集合は明示的には生成せず, かつ常に定義 4.4 を満たす相関ルールのみを探索するため, 提案手法は既存手法より効率がいいといえる.

4.2 極小性措置と接尾木

本稿ではアイテム集合同士を組み合わせ, 有効な負の相関ルールの候補を探す際に接尾木を用い, 左優先深さ優先で探索を行う. 接尾木とは図 1 のような構造の探索木である. この探索木では各ノードは文字列でラベル付けされており, 各ノード A の親がそのノードより一つだけ短い接尾辞 B である. A と B の差分のラベルは順序 $\prec$ 上の辞書式順序において B 中のアイテムより前にある. そして兄弟は順序 $\prec$ で左から右へ並ぶ. 図 2 では $a > b > c > d$ の順序を仮定している. 接尾木では A を訪問する時点で A の部分集合は全て訪問済みであるという特徴がある [4]. この特徴は定義 4.6 を満たす相関ルールを抽出する上で大変都合がよい.

仮に, 図 2 のような接頭木を用いて同様に左優先深さ優先で探索を行うとする. 前件として a を固定し, 後件を $b, bc, bcd, bd, ...$ と順に訪問して有効性を検査する. このとき $a \Rightarrow \neg\{bcd\}$ を有効な候補として抽出したとする. もし, 次に訪問する $a \Rightarrow \neg\{bd\}$ の支持度が閾値 ms 以上であった場合, 定義 4.6 の極小性の条件より $a \Rightarrow \neg\{bcd\}$ は有効ではなくなる. しかし $a \Rightarrow \neg\{bd\}$ が有効でなければ, $a \Rightarrow \neg\{bcd\}$ は有効であるため, 先に訪問

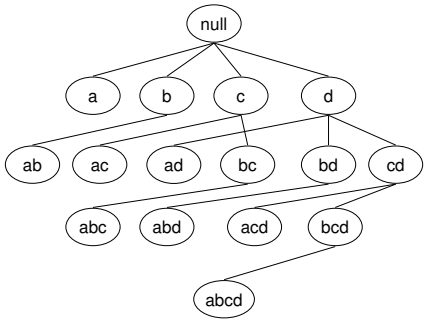

図 1: 接尾木

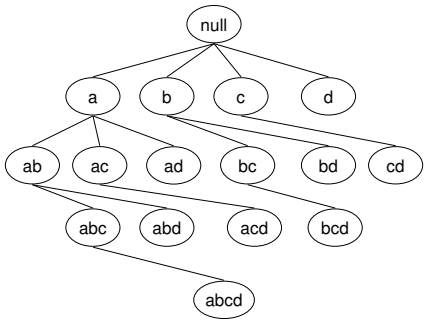

図 2: 接頭木

した $a \Rightarrow \neg\{bcd\}$ を保存しておく必要がある. このように接頭木を用いて探索を行うと, 先に訪問した冗長なアイテム集合の探索結果を保存しておく必要があり, 効率が悪い.

接尾木を用いて左優先深さ優先で探索することで, 先に全ての部分集合を探索した後に, より大きな集合の探索を行える. そのため冗長となるかもしれない候補を記憶しておく必要が無くなり, 効率が良くなる.

4.3 負の相関ルールの生成アルゴリズム

以下にアルゴリズムを示す. 例として $X \Rightarrow \neg Y$ という形式の負の相関ルールの抽出を行うアルゴリズムを示す.

入力はトランザクションデータベース D, 支持度の閾値 ms, 確信度の閾値 mc を与える. 出力は有効な負の相関ルールとなる.

1:D から $supp(X_n) \geq ms$ となるアイテム集合のリスト $FI = \langle X_1, ..., X_n \rangle$ を生成. ただし $\langle X_1, ..., X_n \rangle$ は接尾木を左優先深さ優先でたどった順序である.
2:**for**$(i = 1; i < n; i++)$ **do**
3: **for**$(j = i; j < n; j++)$ **do**
4: **if**$(X_i \cap X_j = \emptyset$ && $supp(X_i \Rightarrow X_j) < ms$ && $X_i \Rightarrow \neg X_j \geq ms)$
5: $X_i \Rightarrow \neg X_j$ を候補とする
6: X_j の子ノードを枝刈り
end if
end for
end for
7:確信度が閾値以上である候補を有効なルールとして抽出

4.1 で述べた手法 [1] では, $X \Rightarrow Y$, $X \Rightarrow \neg Y$ の両方のルー

ルが抽出される可能性をアルゴリズムで解消している．本稿で採用した先行研究 [2] の定義 4(2) は負の相関ルールの定義として妥当なものとなっている．単調性を持つ評価尺度である支持度のみを枝刈りに用いているため, このアルゴリズムは完全であると言える．なお, ここでは興味度によるルールの絞り込みは行っていない．必要に応じて負の相関ルールの候補を全て生成した手順 7 以降で絞り込めば完全性を保障できる．

5. 実験結果と考察

提案アルゴリズムを実装し実験を行った結果を以下に示す．

実験に使用したトランザクションデータベースは先行研究 [1] が実験に使用していた KDD Cup 2000 Data and Questions[6] のデータベースを用いた．正の頻出アイテム集合を抽出するツールとして Apriori[5] を使用した．最小支持度 ms と最小確信度 mc はそれぞれ先行研究と同じものを使用し, 抽出されるルールの数を求めた．結果を表 1 に示す．

表 1: 実験結果

データベース	ms	mc	ルール (個)	実行時間
Question1	0.05	0.4	37	6.1s
Question2	0.05	0.4	103,055	147m51s
Question3	0.1	0.4	53	0.12s

Question1, 3 に比べ, Question2 の実行時間と出力されたルールの数が大きくなっている．これは Question1, 3 では正の頻出アイテム集合の数がそれぞれ数十個程度であったのに対し, Question2 では約 40,000 個存在したためである．提案手法では正の頻出アイテム集合同士を組合わせ, 有効な負の相関ルールを探索するため, 正の頻出アイテム集合の数が増えれば探索量は増えると考えられる．

そこで Question1 の最小支持度を変え, 正の頻出アイテム集合の数を変更し実験を行った．結果を表 2 に示す．正の頻出アイテム集合の数が増えると実行時間が増える傾向がある．

表 2: 最小支持度の値によるルール数と実行時間の変化

ms	mc	頻出アイテム集合 (個)	ルール (個)	実行時間
0.01	0.4	1362	2283	7.8s
0.02	0.4	157	417	7.7s
0.03	0.4	32	135	7.7s
0.04	0.4	25	59	6.3s
0.05	0.4	21	37	6.1s

表 1 の実験結果をもとに先行研究と比較を行うが, 先行研究に記載されていたのは Question2 の台集合の数のみで, 出力されたルールの数は記載されていなかった．また提案手法は先行研究の手法と異なり, 興味度を有効な相関ルールの定義に用いていないため, 最終的に抽出される相関ルールが異なる．よって単純に出力結果を比較することができない．そこで, Question2 の出力結果と先行研究の台集合の数から計算量の比較を行う．

4.1 で述べたとおり, 提案手法で考えられる組合わせの数は最大で (正の頻出アイテム集合の数)^2 となる．Question2 では正の頻出アイテム集合が約 40,000 個出力されたため, 考えられる最大の組合わせの数は 16×10^8 となる．これに定義 4 の (6) に示す極小性を用いて枝刈りを行うことにより, 実際に組合わせて支持度の比較を行う回数は減少する．Qustion1 と 2 を用い, 比較回数を数えた結果を表 3 に示す．

表 3: 極小性を用いた比較回数の変化

データベース	ms	頻出アイテム集合 (個)	比較回数
Question1	0.01	1,362	42,814
	0.02	157	2,723
	0.03	32	535
	0.04	25	371
	0.05	21	271
Question2	0.05	39,401	1,523,102

極小性を用いて枝刈りを行うことにより, 比較回数は大きく減少した．Question2 では約 15×10^5 回と, 枝刈りをしない場合の 1000 分の 1 の比較回数となった．

一方, 既存手法で生成された台集合の総数は 7,382 個 [1] であった．ここから実際に組み合わせて支持度を比較した組合わせの数を調べるには, アイテムを n 個含む台集合がいくつ生成されるか求め, 4.1 の式 1 を用いて求める必要がある．また, 7,382 個という値は興味度を枝刈りに用いた上で生成した台集合の数である．そこで今後, 興味度を用いずに先行研究の手法を用いて台集合を生成し, 分割可能な組み合わせの数を求め, 比較を行う．

6. まとめ

負の相関ルールの完全かつ効率的な抽出は, 正の相関ルールの抽出と比べると困難であった．そこで本稿では, 有効な負の相関ルールの極小性を用い, 完全性を保ちつつ効率的な枝刈りを行うアルゴリズムを提案した．

提案手法では有効な負の相関ルールとして新たな定義を設けたため, 最終的に抽出される有効なルールが既存手法とは異なってしまう．そこで, 提案手法と既存手法それぞれの計算量を求め, 理論的な面からのより詳細な比較実験をこれから行う．

謝辞

本研究は一部，文科省科学研究費補助金（基盤 C：No.22500127 および No.25330256）の援助を受けている．

参考文献

[1] Wu, X., Zhang, C. and Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules. ACM Trans. on Information Systems, vol.22(3), pp381∼405, 2004.

[2] Cornelis, C., Yan, P., Zhang, X. and Chen, G.: Mining Positive and Negative Association Rules from Large Databases. CIS 2006. LNCS(LNAI), vol.4456, pp613∼618, 2006.

[3] Wang, H., Zhang, X. and Chen, G.: Mining a Complete Set of Both Positive and Negative Association Rules from Large Databases. PAKDD'08 Proceedings of the 12th Pacific-Asia conference on Advances in knowledge discovery and data mining, pp777∼784, 2008.

[4] 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. 第 1 回データ指向構成マイニングとシミュレーション研究会 SIG-DOCMAS B101-4, 2-24∼2-32, 2011.

[5] Apriori http://www.borgelt.net/apriori.html (2013).

[6] KDD Cup 2000 Data and Questions http://www.ecn.purdue.edu/KDDCUP/ .

[7] 福田剛志, 森本康彦, 徳山豪: データマイニング, 共立出版 (2001)