負相関ルールを抽出する準オンラインアルゴリズム

The Semi-online Algorithm for Mining Negative Association Rules

黒岩健歩 [1*]　岩沼宏治 [2]　山本泰生 [2]
Yasuho Kuroiwa[1]　Koji Iwanuma[2]　Yositaka Yamamoto[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 mining negative association rules in the framework of a semi-online computation for transaction stream. In order to avoid a combinatorial explosion by using closed itemsets and minimal generators. This algorithm is based on LC-CloStream which is an online ϵ-approximation owing algorithm for frequent closed itemsets. We also show some results of experiments for evaluating our proposed method.

1　はじめに

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を相関の強い共起関係として記述したものである. 本研究で扱う**負相関ルール**は, $X \Rightarrow \neg Y$, $\neg X \Rightarrow Y$ と表記され, アイテム集合 X と Y の出現と非出現の関係を表す. これは, アイテム集合間の隠れた関係を表現することから, 概念抽出やエラー検出への利用が期待される. しかし, 負相関ルールは非出現のアイテム集合を含むため, その数は膨大となる. そのため, 負相関ルール抽出問題は困難であることが知られている. そこで [2] は, アイテム集合の圧縮表現である飽和アイテム集合と極小生成子を用いた負相関ルールの抽出を試み, その有効性を示した.

一方で近年では通信技術の向上やストレージ容量の増加により, 大量のデータが高速に流れるようになった. 交通情報やネットショッピングの購買履歴などの多次元センシングデータがこれにあたり, ストリームデータと呼ばれる. ストリームデータからアイテム集合の圧縮形である頻出飽和アイテム集合を抽出するアルゴリズム LC-CloStream[1] が提案されている. これは膨大な数の候補集合に対して, 効果的に飽和アイテム集合を抽出することを可能とする. このアルゴリズムは近似解法であり, アイテム集合の頻度に対し誤差を許すが, ユーザ指定の許容誤差に基づいた誤差保証を持つ.

*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

本研究では, LC-CloStream から計算される飽和アイテム集合と極小生成子を用いて, ストリームデータから負ルールを抽出する準オンラインアルゴリズムを提案する.

2　準備

2.1　トランザクションストリーム

アイテムの全体集合を $I = \{i_1, i_2, \cdots, i_n\}$ とするとき, I の部分集合を**アイテム集合**, もしくは**トランザクション**と呼ぶ. トランザクション列 $\langle T_1, T_2, \ldots, T_N \rangle$ を**トランザクションストリーム**と呼び, N を**ストリーム長**と定める. アイテムは英子文字で表し, 簡略のためアイテム集合 $\{i_1, i_2, \cdots, i_n\}$ を $i_1 i_2 \cdots i_n$ と表記する. アイテム集合 $\{a, b, c\}$ ならば, abc と表記する.

$S = \langle T_1, T_2, \ldots, T_N \rangle$ をストリーム, X をアイテム集合とするとき, $S(X)$ を $\{T_i \in S | X \subset T_i\}$ となるトランザクション集合とする. このとき, アイテム集合 X の S 上の**絶対頻度** $s(X)$ を $s(X) = |S(X)|$ と定める. また, **相対支持度** $\sup(X)$ を $\sup(X) = \frac{s(X)}{N}$ と定める. ユーザ指定の閾値 ms（以下, **最小支持度**と呼ぶ）に対して, $\sup(X) \geq ms$ を満たす X を**頻出アイテム集合**と呼ぶ.

定義 1 (飽和アイテム集合) アイテム集合 X, X' に対して, $X \neq X'$ であり $X \subset X'$ かつ $s(X) = s(X')$ なる X' が存在しないとき, X を**飽和アイテム集合**と呼ぶ.

よく知られているように, 飽和アイテム集合はアイテム集合の可逆圧縮形式の 1 種である.

2.2 負相関ルール

本研究で取り扱う**負相関ルール** (negative association rule: 以下では適宜 “負ルール” と略記)[3, 4, 5, 6, 7, 8] は, $X \Rightarrow \neg Y$(**右否定形**), $\neg X \Rightarrow Y$ (**左否定形**) のいずれかの表現である. $\neg X$ 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. 負アイテム集合および負ルールの評価尺度は以下のように計算される. ただし, アイテム集合 C_X は X または $\neg X$ のどちらかを表す.

定義 2 ([2, 7]) 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.

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

ここで本研究で抽出すべき妥当な負ルールの定義を示す. **最小支持度** ms と**最小確信度** mc は, ユーザが支持度と確信度に関して与える閾値である.

定義 3 ([2]) **妥当** *(valid)* **な負ルール** $C_X \Rightarrow C_Y$ とは, 以下の 6 つの条件を満たすルールである.

(1) 独立性条件 $X \cap Y = \emptyset$
(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 $\mathrm{conf}(C_X \Rightarrow C_Y) \geq mc$

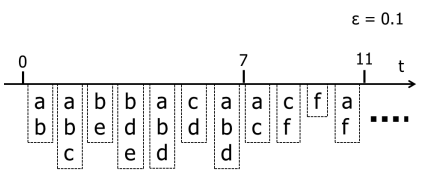
本論文では, 全ての妥当な負ルールの抽出を効果的に行うアルゴリズムの開発を研究の目的とする.

3 先行研究

本章では, 先行研究にあたる LC-CloStream[1] 及び, 極小生成子に基づく負ルール抽出法 [2] について述べる.

3.1 LC-CloStream

LC-CloStream[1] は飽和アイテム集合を計算するオンライン型近似アルゴリズムである.



Cid	CIS	頻度
01	ab	4
02	be	2
03	b	6
04	bd	3
05	abd	2
06	d	3
07	c	5
08	f	3
10	a	6
11	af	2

図 1: ストリームデータと頻度表 (t=11) の例

図 1 から, 出力となる頻度表は各飽和アイテム集合とその頻度の組からなる. ストリームを読み込むごとに頻度表を更新し, 飽和アイテム集合を計算する.

LC-CloStream はアイテム集合の頻度に誤差を許す近似解法であるため, 以下, このアルゴリズムによって計算されるアイテム集合 X の頻度を, **見積もり頻度** $c(X)$ とする. また, X 真の絶対頻度を $s(X)$ とするとき, 頻出アイテム集合の頻度に関して以下の定理が成り立つ.

定理 1 (ϵ 誤差保証 [1]) 許容誤差を ϵ とする時, ある時刻 n におけるアイテム集合 X の絶対頻度 $s(X)$ と見積もり頻度 $c(X)$ について以下が成り立つ.

$$s(X) \leq c(X) \leq s(X) + \epsilon n$$

LC-CloStream はすべての頻出アイテム集合の頻度を誤差 ϵn の範囲で求めることができる.

一方で, LC-CloStream は全ての飽和アイテム集合を抽出しない [1]. しかし, 頻出アイテム集合に対して以下の定理が成り立つことが示されている.

定理 2 (頻出アイテム集合における完全性 [1]) S をストリーム, N をストリーム長, B を S 上の頻出アイテム集合とする場合, *LC-CloStream* により計算されるある飽和アイテム集合 $C \in TS$ について $B \subset C$ なるものが必ず存在する.

これにより, 全ての頻出アイテム集合を計算することが出来るため, LC-CloStream で計算した飽和アイテム集合から負ルールを計算し, 次に頻出アイテム集合からなる負ルールを復元することが可能である.

3.2 極小生成子に基づく負ルール抽出

相関ルールの抽出にはアイテム集合の組合せ計算を行うため, 抽出されるルールの数は膨大である. そこで [2] はアイテム集合の圧縮表現である**飽和アイテム集合**と**極小生成子** [9] を利用することを提案した. 極小生成子 (minimal generator) は閉包の対になる表現であり, その定義を以下に示す.

定義 4 ([9]) アイテム集合 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.

[2] は極小生成子を前件, 後件とし, 負ルールを抽出することを提案した.

4 提案手法

ストリームデータを走査し, 一定の間隔ごとに負ルールを抽出する準オンラインアルゴリズムを提案する.

4.1 極小生成子による負ルール生成の完全性

本手法では, LC-CloStream から計算される飽和アイテム集合から極小生成子を計算し, その組合せから極小生成子からなる負ルールを抽出する.

飽和アイテム集合から効果的に極小生成子を計算するるアルゴリズムを示す.

---

calculateMinimalGenerate(TS) :
入力: TS: 飽和アイテム集合 $X_1, \ldots, X_n$ の集合;
ms: 最小支持度;
出力: MGS: 頻出飽和アイテム集合 $X \in TS$ から計算される極小生成子 mg の $< mg, X >$ なる組の集合;
変数: k: 計算する極小生成子のアイテム数;
α: 極小生成子を抽出する元 $X_1, \ldots, X_n$ の集合;
β^k: α のアイテム数 k の部分集合 $sub_1, \ldots, sub_m$ の集合;
SUB^X: $X \in \alpha$ から計算されるアイテム数 k の部分集合 $sub_1, \ldots, sub_l$ の集合;
γ: $sub \subset X \in TS$ なる飽和アイテム集合 X の集合;

$MGS \leftarrow \emptyset$
$\alpha \leftarrow TS$
$k \leftarrow 1$
while $k \leq \{|X| \mid X \in \alpha\}$ **do**
  $\beta^k \leftarrow \emptyset$
  $SUB^X \leftarrow \emptyset$
  MakeSubset(TS, α, k, β^k, SUB^X)
  registerMG(TS, β^k, MG^X, MGS)
  UpdateElement(MG^X, α)
  $k++$
end while
return MGS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MakeSubset(TS, α, k, β^k, SUB^X)
for each $X \in \alpha$ **do**
  if sup(X) $\geq ms$ **then**
    $SUB^X \leftarrow \{sub \subset X \mid |sub| = k\}$
    for each $sub \in SUB^X$ **do**
      if $sub \notin \alpha$ and ($sub \notin TS$ or $k = 1$) **then**
        $\beta^k \leftarrow \beta^k \cup sub$
      end if
    end for
  end if
end for
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
registerMG(TS, β^k, MGS)
for each $sub \in \beta^k$ **do**
  $\gamma \leftarrow \{X \in TS \mid sub \subset X\}$
  $X \leftarrow arg\max_{X \in \gamma} c(X)$
  $MGS \leftarrow MGS \ \cup \ < sub, X >$
  $\gamma \leftarrow \emptyset$
end for
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
UpdateElement(MGS, α)
for each $X \in \alpha$ **do**
  $X' \leftarrow X$
  for each $\{mg \in MGS \mid X = X \in MGS\}$ **do**
    $X' \leftarrow \{X'\} \cup \{mg\}$
  end for
  $\alpha \leftarrow (\alpha - X) \cup (X - X')$
end for

まず MakeSubset において, 頻出な飽和アイテム集合の元から部分集合を計算する. 次に registerMG で部分集合について, 自身のスーパーセットであり, 同じ頻度を持つ飽和アイテム集合を求める. この組の情報は, 極小生成子から全てのアイテム集合を復元する際に必要となる. UpdateElement は, 既に抽出された極小生成子を元から除くことで, 冗長な部分集合の計算を効果的に省く. アイテム数の小さい部分集合から計算することで, 極小生成子になるアイテム集合にのみに計算対象を絞ることが出来る. 以上の手続きにより, 飽和アイテム集合から極小生成子を効果的に抽出する.

ここであるアイテム集合 X とその頻度 $c(X)$ を X: $c(\alpha)$ と表す. 例として, 飽和アイテム集合 $abcd$: 2, def: 2, bd: 3, c: 4 , d: 5, e: 3 から極小生成子を計算する場合を考える. 1 度目のループで $k = 1$ とし, MakeSubset でにおいて部分集合として a, b, c, d, e, f が計算される. 次に registerMG で自身のスーパーセットであり, 最大の頻度を持つ飽和アイテム集合を計算する. a は $abcd$: 2, b は bd: 3, c は c: 4, d は d: 5, e は e: 3, f は def: 2 を組として登録する. 最後に UpdateElement により, 飽和アイテム集合の元から極小生成子を削除し, ~~a~~bcd: 2, de~~f~~: 2, ~~b~~d: 3, ~~c~~: 4, ~~d~~: 5, ~~e~~: 3 と更新する. $k = 2$ 以降も同様にこれを繰り返す. 最終的に飽和アイテム集合から極小生成子として, $abcd$: 2 から a と bc と cd, def: 2 から f と de, bd: 3 から b, c: 4 から c, d: 5 から d, e: 3 から e が得られる.

4.2 極小生成子上の負ルールの評価尺度

LC-CloStream は近似解法であるために, あるアイテム集合 X の頻度として見積もり頻度 $c(X)$ を計算する. したがって, 見積もり頻度を基に負ルールを抽出する場合, 同様に負ルールの評価尺度にも誤差が入り込むため, その影響を考慮する必要がある. 見積もり頻度を元に計算するアイテム集合及び負ルールの支持度, 確信度を以下のように定義する.

定義 5 推定支持度 esup, **推定確信度** econf は, 見積もり頻度から計算される以下の 5 つの評価尺度である.

(1) $\text{esup}(X) = \frac{c(X)}{N}$

(2) $\text{esup}(\neg X) = 1 - \frac{c(X)}{N}$

(3) $\text{esup}(X \Rightarrow \neg Y) = \frac{c(X) - c(X \cup Y)}{N}$

(4) $\text{esup}(\neg X \Rightarrow Y) = \frac{c(Y) - c(X \cup Y)}{N}$

(5) $\text{econf}(X \Rightarrow \neg Y) = \frac{c(X) - c(X \cup Y)}{c(X)}$

(6) $\text{econf}(\neg X \Rightarrow Y) = \frac{c(Y) - c(X \cup Y)}{N - c(X)}$

(1), (2) はアイテム集合及び負アイテム集合の推定支持度, (3), (4) は負ルールの右否定形及び左否定形の推定支持度, (5), (6) は負ルールの右否定形及び左否定形の

推定確信度の定義である. 先に定義した推定支持度, 推定確信度は, 見積もり頻度から計算されるために, 厳密解法で求めた支持度, 確信度に対して誤差を生じる. それぞれの評価尺度について, その誤差は以下の範囲内であることが保証できる.

補題 1 (推定支持度, 推定確信度における誤差保証) 各推定評価尺度における真の支持度, 確信度との誤差は以下の範囲で保証される. ただし, ms は最小支持度, ϵ は許容誤差である.

(1) $\mathrm{esup}(X) - \sup(X) \in [0, \epsilon]$

(2) $\mathrm{esup}(\neg X) - sup(\neg X) \in [-\epsilon, 0]$

(3) $\mathrm{esup}(X \Rightarrow \neg Y) - \sup(X \Rightarrow \neg Y) \in [-\epsilon, \epsilon]$

(4) $\mathrm{esup}(\neg X \Rightarrow Y) - \sup(\neg X \Rightarrow Y) \in [-\epsilon, \epsilon]$

(5) $\mathrm{econf}(X \Rightarrow \neg Y) - \mathrm{conf}(X \Rightarrow \neg Y) \in [-\frac{\epsilon}{ms}, \frac{\epsilon}{ms}]$

(6) $\mathrm{econf}(\neg X \Rightarrow Y) - \mathrm{conf}(\neg X \Rightarrow Y) \in [-\frac{\epsilon}{1-(\sup(X)+\epsilon)}, \frac{2\epsilon}{1-(\sup(X)+\epsilon)}]$

全ての尺度において, 想定される誤差が許容誤差 ϵ に大きく影響を受け, 許容誤差 ϵ が小さい場合にその誤差も減少する. (5) については最小支持度 ms と ϵ の関係により計算されるが, 一般に ms は ϵ より 1 桁以上大きい値を取る. 全ての評価尺度について, 現実的な範囲で誤差を見積もることが可能となった.

4.3 極小生成子による負ルール抽出

本節では, 極小生成子を用いた素朴な負ルールの抽出計算について示す. 以下にそのアルゴリズムを示す.

simpleMiningFromMG(TS) :
入力: TS: LC-CloStream より計算される飽和アイテム集合の集合
出力: RL: 現時刻における妥当な負ルール集合
変数: MGS: TS から抽出される頻出な極小生成子の集合

```
1: MGS ← calculateMinimalGenerate(TS)
2: for each X ∈ MGS do
3:   for each Y ∈ MGS do
4:     X, Y からなる負ルールの妥当性検査
5:   end for
6: end for
```

以下, TS のうち頻出飽和アイテム集合から計算される極小生成子の集合を MGS とする. 素朴に負ルールを抽出する場合, 負ルール抽出における候補解の組合せ数は極小生成子の組合せ数, 即ち $|MGS| \times |MGS|$ となる. 抽出時毎に全ての組合せを計算するのは非効率である. そこで次節では, 組合せ数を減らす手法について述べる.

4.4 差分情報による効果的な負ルール抽出

素朴な負ルール抽出は, 各トランザクションが到着するたびに全ての極小生成子の組合せを計算するため非効率である. そこで, 前回計算した結果を利用した負ルールの抽出計算を検討する.

4.4.1 時刻経過による絶対頻度の変化

まず, 時刻変化による各アイテム集合の絶対頻度の変化に着目する. 例として, 図 1 のストリームから計算した 2 つの時刻の極小生成子の集合 MGS を図 2 に示す.

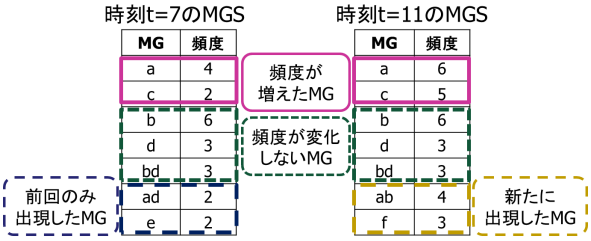


図 2: 極小生成子の集合 MGS の時刻経過による変化

図 2 は, 現時刻 $t = 11$ と前回 $t = 7$ の極小生成子の集合を示している. 図 2 のように, 極小生成子は 4 つの区分に分けられる. まず, 1 つ目に前回のみ出現した極小生成子は, 現時刻では負ルールの前件, 後件の極小生成子として含まれることはないため, それらを含む負ルールは表から削除する必要がある. 2 つ目に新たに出現した極小生成子について, その極小生成子を含む負ルールは抽出されていないため, 新たに抽出する必要がある. 3 つ目に頻度が変化しない極小生成子, 4 つ目に頻度が増えた極小生成子がある.

頻度が変化しない, または増えた極小生成子について, それぞれ評価尺度である支持度, 確信度に対する影響に関して検討する.

4.4.2 時刻経過による評価尺度の変化

時刻経過により, 負ルール $C_X \Rightarrow \neg C_Y$ について, X, Y の絶対頻度 $s(X)$, $s(Y)$ 及びストリーム数 N が変化する. 時刻変化により, X と Y の評価尺度の値がどのように変化するか調べた結果を図 3 に示す.

	$s(X)$ 増加 $s(Y)$ 増加	$s(X)$ 増加 $s(Y)$ 変化なし	$s(X)$ 変化なし $s(Y)$ 増加	$s(X)$ 変化なし $s(Y)$ 変化なし
$\sup(X)$	—	—	単調減少	単調減少
$\sup(Y)$	—	単調減少	—	単調減少
$\sup(X \to \neg Y)$	—	—	単調減少	単調減少
$\sup(\neg X \to Y)$	—	単調減少	—	単調減少
$\mathrm{conf}(X \to \neg Y)$	—	単調増加	不変	不変
$\mathrm{conf}(\neg Y \to X)$	—	—	単調減少	単調減少
$\mathrm{conf}(Y \to \neg X)$	—	不変	単調増加	不変
$\mathrm{conf}(\neg X \to Y)$	—	単調減少	—	単調減少

図 3: 評価尺度の時刻経過における性質

列は, アイテム集合 X と Y の頻度が増加した場合としなかった場合の 4 つの組合せである. 行は各評価尺度を示す. 評価尺度の値が単調増加, 単調減少, 不変であるかを図 3 に示した. 未記入の箇所は傾向が不定であることを示す. 図 3 より, $s(X)$, $s(Y)$ が共に不変で

ある場合に着目すると, 各評価値は単調減少もしくは不変である. よって, 前回抽出計算した際に妥当な負ルールとして抽出されなかった負ルール $C_X \Rightarrow \neg C_Y$ は, 現時刻でも同様に妥当な負ルールになることはない. また, 現時刻で妥当な負ルールとして抽出されるならば, 必ず前回の負ルール表に存在する.

したがって, X, Y が共に出現しない場合については, その組み合わせを検査せずに, これまでの妥当な負ルールを記録した表にあるため, 負ルールの評価値を更新するのみで良い.

ここで図 2 のうち, 新たに出現した極小生成子及び頻度が増えた極小生成子の集合を**更新表 (US)** とする.

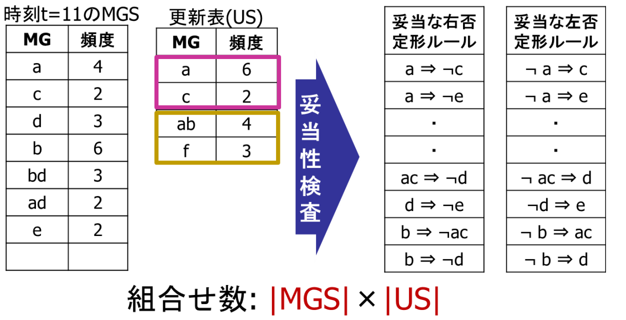


図 4: 差分情報を用いた負ルール抽出

頻度が変化しない極小生成子の組合は計算する必要がない. したがって, 図 4 の様に, MGS と更新表の組合せにより負ルールをするだけでよい.

したがって, 素朴な負ルール抽出では $|MGS| \times |MGS|$ 個の極小生成子の組合せを検査する必要があるのに対し, 差分情報を用いた負ルール抽出では $|MGS| \times |US|$ の極小生成子の組合せと負ルール表の走査のみで十分である.

4.5 準オンライン型マイニングアルゴリズム

LC-CloStream をベースとした負ルールの準オンライン型マイニングアルゴリズムを示す. まず, 1 回目の処理として, 素朴な極小生成子の組合せ計算により負ルールを抽出する. 2 回目以降の負ルール抽出については, 前回の計算結果の差分情報から組合せ数を削減した効果的な抽出を行う.

LC-CloStream で呼び出される, 負ルールを抽出する準オンラインアルゴリズムのメインルーチンを示す.

負ルールを抽出する準オンラインアルゴリズム：

入力: TS: 頻度表 TS;
PMG: 前回抽出時の極小生成子集合;
RL: 妥当な負ルール集合 RL

出力: RL: 現時刻における妥当な負ルール集合

変数: N: ストリーム長
$turn$: 負ルールの抽出計算をした回数;
$interval$: 負ルールを抽出する間隔;
MGS: 現時刻の極小生成子の集合;
US: 負ルールの要素として妥当性を検査すべき極小生成子の集合;
$rule(X, Y)$: ある極小生成子 X, Y からなる負ルール;

```
turn ← 1
N ← 1
if turn = 1 and interval × turn = N then
  simpleMiningFromMG(TS)
  PMG ← MGS
  turn++
else if interval × turn = Nthen
  UpdateRules(TS, PMG, RL)
  PMG ← MGS
  turn++
end if
N++
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
UpdateRules(MGS, PMG, RL)
MGS ←calculateMinimalGenerate(TS)
US ← (MG − PMG) ∪ {mg|mg は頻度が増加 }
for each X ∈ MGS do
  for each Y ∈ US do
    X, Y からなる負ルールの妥当性検査
  end for
end for
for each rule(X, Y) ∈ RL do
  if X ∉ MGS or Y ∉ MGS then
    RL 内から負ルール rule(X, Y) を削除
  else
    負ルール rule(X, Y) の評価尺度の更新
  end if
end for
```

---

2 回目以降の負ルール抽出では, 新たに妥当になりうる負ルールに属する極小生成子の集合 US と MGS の組合せ及び, 妥当な負ルール集合 RL を一度走査することで負ルールを更新している.

負ルールの抽出間隔 $interval$ が短い場合, 差分情報から得られる極小生成子の集合 US は小さくなることが予想されるため, 差分情報の利用が負ルールの候補の削減に効果的に働くと考えられる.

5 評価実験

提案した負ルールを抽出する準オンラインアルゴリズムについて実装し, 負ルールの抽出実験を行った. 本章では, 実験の結果とその考察を示す.

実験にはデータセットとして, retail(sparse)[10] と mushroom(dence) [11] の 2 種を使用した. 各データセットの詳細を表 1 に示す.

表 1: 実験に使用したデータ

データセット	#(item)	#(trans.)	ave(item)
retail	16,470	88,162	10.3
mushroom	119	8,124	23

表 1 の #(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

表 2: 負ルールの準オンライン抽出実験結果

データセット	抽出間隔	抽出回数	ave(更新表割合)(%)	ave(候補ルール削減率)(%)	ave (MG)	ave (右否定形)	ave (左否定形)
retail[1]	1,000	88	91.2	6.9	236	15,435	469
	100	872	51.3	40.1	236	15,263	469
	50	1,744	34.1	53.9	236	15,343	469
	10	8,717	10.8	72.0	236	15,453	469
	1	87,163	1.9	78.2	236	15,423	469
mush room[2]	1,000	8	85.0	14.5	290	818	835
	100	72	91.7	8.1	308	972	988
	10	713	79.6	20.1	296	1,023	1,055
	1	7,125	38.2	61.0	296	1,029	1,061

1 最小支持度 ms=0.005, 最小確信度 ms=0.4, 許容誤差 ϵ=0.002
2 最小支持度 ms=0.4, 最小確信度 ms=0.5, 許容誤差 ϵ=0.04

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.

本実験ではストリームを考慮するため, 予めトランザクションストリームを 1000 だけ読み込んだ時点から負ルールの抽出を行った.

表 2 について, 抽出間隔は負ルールの抽出計算を行うストリームの間隔, 抽出回数は負ルールの抽出計算を行った回数である. 更新表 US 割合は極小生成子集合のうち, 妥当性の組合せ計算をすべき極小生成子の割合である. 更新表割合は小さいほど負ルールの妥当性検査の回数は少なくなるため望ましい. 候補ルール削減率を以下のように定義する.

$$\text{候補ルール削減率} \quad = \quad 1 - \frac{\text{実際に検査した負ルール数}}{\text{直積 } |MGS|^2} \quad (\%)$$

これは素朴な抽出に対し, 差分計算が計算すべきルールをどれだけ削減できたかを示す. $ave(MG)$ は頻出な極小生成子の数の平均, ave(右否定形) 及び ave(左否定形) はそれぞれ妥当な負ルールの数の平均を示す.

表 2 より, retail, mushroom ともに抽出間隔が狭くなるにつれ, 更新表 (US) 割合が減少し, 候補ルール削減率が増加している. 特に retail では, 抽出区間 10, 1 のとき, 70%の計算を削減されている. mushroom は, retail に比べてその効果は低い結果となった. これは mushroom が dence なデータセットであるため, 飽和アイテム集合の頻度の更新が頻繁に起きているためと考えられる.

6 おわりに

本論文では準オンラインの負ルール抽出アルゴリズムを提案した. 頻度に許容誤差を含むアイテム集合に対して, そのアイテム集合からなる負ルールの評価尺度を定義し, その評価値の誤差保証を行った. また, 差分情報による効果的な負ルールの組合せ計算の効率化手法を示し, 評価実験によりその有効性を示した.

今後の課題として高速化するためのデータ構造の検討や極小生成子のオンライン計算が挙げられる.

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

[1] 福田翔士, 岩沼宏治, 山本泰生: トランザクションストリーム上のオンライン型頻出飽和集合マイニング, 人工知能学会研究会資料, SIG-FPAI, B404, pp.1-6, (2015)

[2] 佐生隼一, 岩沼宏治, 黒岩健歩, 山本泰生: 極小生成子を用いた負の相関ルール抽出の高速抽出アルゴリズム, 情報科学技術フォーラム, FIT, D-001, (2015).

[3] 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, JSAI, 29 巻 4 号 A, pp.406-415, (2014).

[4] 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).

[5] 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).

[6] 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).

[7] 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).

[8] 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).

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

[10] Frequent Itemset Mining Dataset Repository. http://fimi.ua.ac.be/data/. (最終アクセス日: 2016/2/4).

[11] Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets/Mushroom. (最終アクセス日: 2016/2/4).