F-021

情報量と頻度に基づく系列データマイニングにおける非同期パターンの抽出と効率化

A Fast Asynchronous Pattern Mining Based on Self-Information and Frequency

村田順平[1]　岩沼宏治[2]　大塚尚樹[13]　鍋島英知[2]
Junpei Murata[1]　Koji Iwanuma[2]　Naoki Ohtsuka[13]　Hidetomo Nabeshima[2]

1　はじめに

本論文では，長大な単一系列データベースから，頻出かつ情報量的に有用な非同期パターンの抽出を目的として，高速マイニングアルゴリズムを提案する．

系列データマイニングとは，巨大なデータ系列から有用な系列パターンを高速抽出する技術である．従来研究[1]では，出現回数の多いパターンを有用とする手法が開発されている．しかし，我々が従来から研究している新聞記事コーパスなどでは，高頻出な系列パターンの殆どは意味の無いものであり，中程度以下の頻度をもつ系列パターンの方に有用なものが多い．そこで本研究では，自己情報量と頻度を評価尺度として，双方のバランスがとれた有用なパターンの抽出に取り組む．先行研究のInfoMiner [3]は非同期パターンの抽出ができず，アイテム集合の系列も取り扱わないので，我々の研究対象である新聞記事コーパスのマイニングに不向きである．本研究では情報量と頻度に基づき，非同期パターンを抽出する高速マイニングアルゴリズムを新たに提案する．

2　準備

定義 1 すべてのアイテムの集合を $E = \{e_1, e_2, ..., e_n\}$ とする．$|E|$ で集合 E の大きさを表す．**系列データベース**とはアイテム集合の順序付きリスト $S = (s_1, s_2, \cdots s_m)$ である．このとき，$s_i \subseteq E$ $(1 \leq i \leq m)$ である．各 s_i $(1 \leq i \leq m)$ は S の**要素**と呼ばれ，S の**系列長**を $|S|$ で表す．また，$*$ を仮想アイテムと呼ぶ．$E^+ = E \cup \{*\}$ を**拡張アイテム集合** と呼ぶ．

$*$ はワイルドカードに相当するアイテムである．以下では系列の要素で単一の要素からなるものは，括弧をはずして表記する．即ち $(\{a\}, \{a, b\}, \{c\}, \{a, b, e\}, \{a, c\})$ を $(a, \{a, b\}, c, \{a, b, e\}, \{a, c\})$ のように略記する．

定義 2 パターン P とは，拡張アイテム集合 E^+ 中のアイテムの順序付きリスト $P = (p_1, p_2, \cdots, p_m)$ である．

定義 3 パターン $P = (p_1, p_2, \cdots, p_m)$ とアイテム $e (e \in E^+)$ に対して，P の末尾に e を追加したパターン $(p_1, \cdots, p_m, e)$ を**右拡張**と呼び，$P \circ e$ で表す．また，P を有限回右拡張したパターンを P の**右拡張パターン**と呼ぶ．以降，拡張とは全て右拡張を指すものとする．

[1]山梨大学大学院コンピュータ・メディア工学専攻
[2]山梨大学大学院医学工学総合研究部
[3]現在 三洋電機株式会社

3　非同期パターンの出現頻度と正規化

パターン P の**出現頻度**とは，系列データベース S 中での P の出現回数である．本論文では非同期パターンを抽出するためにスライド窓 (sliding window) [5] を導入する．

例 1 系列データベース $S = (a, \{a, b\}, c, \{a, b, e\}, \{a, c\}, \{b, a\}, b)$ でパターン $P = (a, *, b)$ は，以下のように非同期に3回出現している．

$$(a, \{\underline{a}, b\}, c, \{\underline{a}, \underline{b}, e\}, \{\bar{a}, c\}, \{\underline{b}, a\}, \bar{b})$$

定義 4 (Iwanuma [5]) 系列データベース $S = (S_1, \ldots, S_m)$ と整数 i $(1 \leq i \leq m)$ に対して，S の i 番目の要素を開始位置とする長さ k $(1 \leq k)$ のウィンドウ系列 $win(S, i, k)$ を以下のように定義する．このときの k を**窓幅**と呼ぶ．

$$win(S, i, k) = \begin{cases} (s_i \cdots s_{i+(k-1)}) & \text{if } i + (k-1) \leq m, \\ (s_i \cdots s_m) & \text{otherwise.} \end{cases}$$

定義 5 s をアイテム集合 E の部分集合，e を拡張アイテム集合 E^+ の要素とするとき，二項演算 $s \models e$ を以下のように定める．

$$s \models e = \begin{cases} true & if & e = * \\ true & if & e \neq * \text{かつ} e \in s \\ false & otherwise. & \end{cases}$$

定義 6 系列 S 中でのパターン P の出現頻度関数 $F(S, P)$ を以下のように定める．

$$F(S, P) = \sum_{i=1}^{|S|} \delta(win(S, i, |P|), P)$$

ここで δ は以下の関数と定める．

$$\delta((s_1 \cdots s_l), (p_1 \cdots p_m)) =$$

$$\begin{cases} 1 & \text{if } l = m \text{ かつ } s_1 \models p_1, \cdots, s_m \models p_m \\ 0 & \text{otherwise} \end{cases}$$

$\delta(win(S, i, |P|), P) = 1$ となる i を P の S 中の**出現位置**と呼ぶ．

以上の出現頻度関数により非同期なパターンを抽出する．例えば例 1 の S と P においては，スライド窓 $win(S, 2, |P|)$, $win(S, 4, |P|)$, $win(S, 5, |P|)$ において，δ 関数が1になり，パターン P を検出する．

M をパターンに対して非負数を返す関数とする．M が**右逆単調**であるとは，パターン P_1 の右拡張パターン P_2 に対して必ず $M(P_1) \geq M(P_2)$ となる場合を言う．上の F に対して以下が成立つ．紙面の都合上，証明は省略する．詳細は [2] を参照して頂きたい．

補題 1 出現頻度関数 F は右逆単調である．

パターン $P_1 = (*, a, b, *, c, *)$ と $P_2 = (*, *, a, b, *, c)$ シフト同値な関係にあり，スライディング窓を用いた場合，この 2 つのパターンの抽出は本質的に同じと考えられる．よってこの 2 つのパターンを同時に考えて，データマイニングを行うことは無駄である．これを防止するため，パターンの**正規化**を考える．

定義 7 パターン P のスケルトンとは，P に出現する E の元の中で最初と最後のものの間の部分系列を言い，$Sk(P)$ で表す．

上記の P_1 と P_2 に対して，スケルトン $Sk(P_1), Sk(P_2)$ は $Sk(P_1) = (a, b, *, c)$ および $Sk(P_2) = (a, b, *, c)$ となり，同一である．スケルトンはパターンを特徴付けるものと考えられる．

定義 8 長さ n に対するパターン P の $Sk(P)$ を $(p_1, \cdots, p_k)$ $(1 \leq k \leq n)$ とする．このとき P の**正規パターン** $N(P)$ とは，$(p_1, \cdots, p_k, *, \cdots, *)$ なる長さ n のパターンを言う．

例えば，パターン $(*, a, b, *, c, *)$ の正規パターンは $(a, b, *, c, *, *)$ となる．正規パターンのみを考えることにより，データマイニングにおける抽出パターンの数を押さえ込める．

4 情報量利得

本章では，過度に出現するパターンの抽出を抑制するために，Yang ら [3] にならい，情報量と頻度を組み合わせた情報量利得を導入する．

定義 9 系列データベース S 上の出現アイテム $e \in E^+$ の**自己情報量**を $I(S, e)$ と表し，以下のように定義する．

$$I(S, e) = -\log_{|E|} \frac{F(S, (e))}{|S|} \quad (1)$$

式 (1) の形式は Yang らの定義と同じであるが，出現頻度関数が異なる点に注意して頂きたい．またワイルドカードの役割を果たす $*$ は，$F(S, *) = |S|$ であるため，$I(S, *) = 0$ となる．

定義 10 ([3]) 系列データベース S 上のパターン $P = (p_1, p_2, \cdots p_m)$ の**情報量** $I(S, P)$ を，以下のように定義する．

$$I(S, P) = I(S, p_1) + I(S, p_2) + \cdots + I(S, p_m)$$

定義 11 **情報量利得関数** $G(S, P)$ を以下で定義する．MS は予め与えられた最小出現頻度である．

$$G(S, P) = I(S, P) \times (F(S, P) - (MS - 1))$$

情報量利得とは，系列データベース S 上のパターン P の価値である．本論文では，与えられた閾値 MG 以上の利得をもつ S 中の正規パターンの抽出を考える．一般に情報量と出現頻度は相反するので，双方のバランスの取れたパターンの利得が高くなる．また，パターン P の情報量は必ず正値であるので，最小出現頻度 MS を正値として与えれば，出現頻度が MS 未満のパターン P の情報量利得は必ず負になり，P は抽出されないことに注意して頂きたい．InfoMiner では最小出現頻度を 2 に固定しており，変更できない．

5 非同期パターンの高速抽出

5.1 探索空間の再構成

本論文では木構造の探索空間を考え，深さ優先探索を行う．メモリ使用量を削減するため，先行研究 InfoMiner の探索空間を再構成する．アイテム集合 E，抽出パターンの長さ W と仮定するとき，本論文で新たに提案するの探索木は以下の条件を満たす木である．

1. 各節点は長さ W 以下の正規パターンである．
2. 根は空列パターン ϵ である．
3. 葉でない各節点は，その節点のパターン P を各 $e \in E^+$ で右拡張したパターン $P \circ e$ を子節点として持つ．

この探索木は，E^+ 上の W 以下の長さの正規パターン全てを網羅する．深さ t の節点は長さ t のパターンであり，深さ W の節点は葉である．深さ優先探索を行い，葉のパターンの情報量利得が閾値以上ならば抽出する．

InfoMiner の探索木は，上の条件 3 を以下の 3' に置き換えた木である

3' 葉でない各節点は，その節点のパターン P を以下のように拡張したパターン P_c 全てを子節点に持つ．

$$P_c = P \circ (*, \cdots, *, e)$$

但し，P_c の長さは n 以下であり，$e \in E$ である．

InfoMiner と提案手法の最大メモリ使用量を考える．アイテム集合 E，抽出パターン長を W とする．提案手法の場合，それぞれの節点が持つ子節点の数は $|E| + 1$ である．深さ W になるまで子節点を作成するので，最大の節点数は以下の式で求めることができる．

$$W \times (|E| + 1)$$

これに対して，InfoMiner の探索木の最大の節点数は以下のようになる (紙面の都合のため，詳細は省略する．[2] を参照して頂たい)．

$$\sum_{i=1}^{W} (|E| \times (W - i + 1)) = \frac{|E| \times (W^2 + W)}{2}$$

InfoMiner と提案手法が保持すべき節点数の最大数はそれぞれ $O(|E| \times W^2)$，$O(|E| \times W)$ となる．本提案法の方が格段に効率がよい．

5.2 情報量利得に基づく枝刈り手法

有用パターンの探索は，深さ優先戦略に従って探索木を順次拡張しながら行う．探索を効率化するために，大塚ら [4] にならい以下の二つの枝刈手法を導入する．

パターン拡張判定: 節点（パターン）を拡張して子節点を生成するかどうか判定する．

アイテム選別: パターン拡張をする際に用いるアイテムを選別判定する．

InfoMiner では，アイテムの選別のみ行っているが，提案手法では，1 の手法を併用する．また 2 つの判定を精密化し，より効果的なものに改良する．本論文では紙面の都合上，パターン拡張判定のみを説明する．アイテム選別については [3, 2] を参照して頂きたい．

5.3 パターン拡張判定

パターン拡張判定は「どのアイテムを用いて右拡張しても情報量利得を超えないパターン (節点) は拡張しない」ことを判定するものである．

本手法で用いる情報量利得は逆単調性を持たない．逆単調性によらないで探索空間を絞り込む必要があり，Yang ら [3] は最小必要反復回数 $min\_rep(S,P,k)$ を導入している．$min\_rep(S,P,k)$ とは，パターン P の右拡張パターンが情報量利得が閾値 MG を満たすために必要な P の最小の出現頻度であり，以下で定義する．

定義 12 系列データベース S の要素 $s_i = \{e_1, e_2, \cdots, e_n\}$ に対して，s_i の中で最も情報量が高いアイテムの情報量 $I\_max(S, s_i)$ と定める．即ち，

$$I\_max(S, s_i) = \max_{e \in s_i}(I(S, e))$$

定義 13 系列データベース S，パターン P に対して，S 上の P の出現位置からなる集合を出現位置集合と呼び，$Occ(S,P)$ と表記する．

例 2 系列データベース $S = (a, \{a,b\}, c, \{a,b,e\}, \{a,c\}, \{b,a\}, b)$，パターン $P = (a,b)$ とするとき，P は

$$(\underline{a}, \{a,\underline{b}\}, c, \{a,b,e\}, \{\underline{a},c\}, \{\underline{b},\bar{a}\}, \bar{b})$$

のように出現しており，$Occ(S,P) = \{1,5,6\}$ となる．

定義 14 S を系列データベース $(s_1, \cdots, s_n)$，P をパターン，W を抽出パターン長とする．P を接頭辞とする長さ W のパターンの S 上での最大情報量 $I\_Max(S,P,W)$ を以下の式で定める．

$$I\_Max(S,P,W) = I(S,P) + \max_{i \in Occ(S,P)}\{\sum_{j=i+|p|}^{W+i-1}(I\_max(s_j))\}$$

上記の I_max は InfoMiner[3] の I_Max の精密化になっている．本論文では S に実存するパターンの情報量を用いているが，InfoMiner では実際には出現しない仮想のパターンの情報量も用いているために，本手法より荒く (より大きく) 算出される．詳細は [2] を参照していただきたい．

例 3 $S = (a,b,c,b,f,i,b,a,c,b,f,c,d,i,a)$，抽出パターン長 $W = 3$，パターン $P = (c)$ に対する $I\_Max(S,P,W)$ を算出する．まず各アイテムの出現頻度と自己情報量を求めると以下の表のようになる．

アイテム e	$F(S,(e))$	$I(S,e)$
a	3	0.893
b	4	0.738
c	3	0.898
d	1	1.511
i	2	1.125
f	2	1.125

(c) の出現は $(a,b,\underline{c},\underline{b},\underline{f},i,b,a,\underline{c},\underline{b},\underline{f},\underline{c},\underline{d},\underline{i},a)$ なので，$Occ(S,(c)) = \{3,9,12\}$ である．よって $I\_Max(S,(c),3)$ は以下のようになる．

$$I\_Max(S,(c),3) = I(S,(c)) + \max\begin{Bmatrix} I(S,b)+I(S,f) \\ I(S,b)+I(S,f) \\ I(S,d)+I(S,i) \end{Bmatrix} =$$

$$= 0.898 + \max\{1.963, 1.963, 2.636\} = 3.534$$

定義 15 系列データベース S，パターン P，抽出パターン長 W，最小情報量利得 MG，最小出現頻度 MS に対し，最小必要反復回数 $min\_rep(S,P,W)$ を次の式で定義する．

$$min\_rep(S,P,W) = \frac{MG}{I\_Max(S,P,W)} + (MS-1)$$

定理 1 S を系列データベース，P をパターン，W を抽出パターン長，MG を最小情報量利得とする．このとき P の出現頻度 $F(S,P)$ が

$$min\_rep(S,P,W) \leq F(S,P)$$

を満たさないならば，いかなる P の 右拡張パターンも MG を満たすことは無い．

証明 パターン P の右拡張パターンで，S 中で最大の情報量を持つパターンを P_{max} とする．P_{max} が

$$min\_gain \leq G(S,P)$$

を満たすと仮定すれば，以下が成立つ．

$$min\_gain \leq I(S,P_{max}) \times (F(S,P_{max}) - (MS-1))$$

整理すると以下を得る．

$$\frac{min\_gain}{I(S,P_{max})} + (MS-1) \leq F(S,P_{max})$$

補題 *1* より，出現頻度関数 F は右逆単調性を持つため，$F(S,P_{max}) \leq F(S,P)$ が成立つ．よって

$$\frac{min\_gain}{I(S,P_{max})} + (MS-1) \leq F(S,P) \qquad (2)$$

が成立つ．よって P が式 *2* を満すことは，P の右拡張が min_gain を越える情報量利得を持つための必要条件となっている．

$I(S,P_{max}) = I\_Max(S,P,W)$ であるので，式 *2* の左辺は $min\_rep(S,P,W)$ と同義である．よって P の出現頻度が $min\_rep(S,P,W)$ を越えない場合，どのように拡張しても min_gain を満すことは無い． □

以上より，系列データベースを S，パターンを P，抽出パターン長を W とするとき，

$$min\_rep(S,P,W) \leq F(S,P)$$

を満す P のみを拡張すれば，探索は完全であることが分る．

6 評価実験

本章では提案手法の実験的評価を行う．提案手法を C 言語で実装した．実験環境は，PentiumCPU3GHz，メモリ 2GB である．実験データは，毎日新聞 2004 年上半期社会面の各記事から TFIDF 上位 1 単語を抜き出しアイテムとした系列データベースである．系列データベースの要素は同日の記事から抽出したアイテムの集合で，系列の長さは 181，系列に出現するアイテムは 6825 種類，アイテムの出現総数は 11857 である．

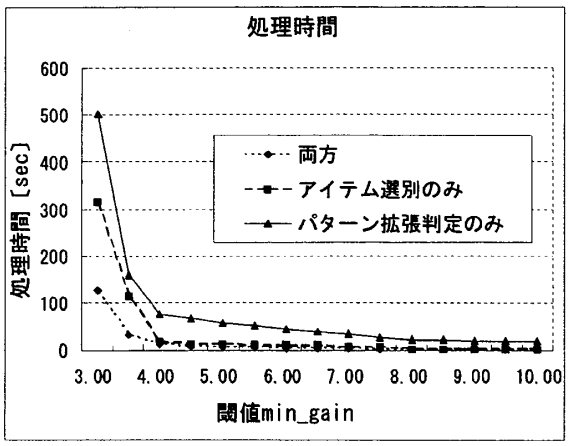

図 1: 枝刈り性能

6.1 頻度尺度によるマイニングとの比較

抽出パターン数を比較することで，無駄と思われるパターンの抽出をどれだけ抑制できたのかを調べる．

提案手法は，抽出パターン長 5，最小出現頻度 $MS = 3$ とし，情報量利得閾値 MG を変化させマイニングを行う．提案手法により抽出されたパターンをすべて含むように頻度尺度基準だけを用いてパターンの抽出を行う．実験結果を表 1 に示す．表 1 は，閾値と対応して抽出されたパターン数をそれぞれ表す．

表 1: 提案手法と頻度尺度手法の比較

提案手法		頻度尺度手法	
MG	抽出パターン数	MS	抽出パターン数
10.0	1	20	23
8.00	9	13	73
5.00	93	7	619
3.00	407	4	4508

提案マイニング方により抽出されるパターン数は，実験の全ての条件において，頻度尺度基準で抽出されるパターン数を大きく下回った．これにより，提案手法は情報量的に価値の低いノイズパターンの多くを取り除いていることが分る．

6.2 提案手法の枝刈り性能の評価

以下の 3 つの枝刈り方法でのマイニング実行時間を比較する．

1. アイテム選別とパターン拡張判定を両方適用
2. アイテム選別だけを適用
3. パターン拡張判定だけを適用

抽出パターン長 $W = 7$，最小出現頻度 $MS = 2$ とし，情報量利得 MG を変化させマイニングし，実行時間を計測する．実験結果を図 1 に示す．縦軸が処理時間，横軸が閾値 MG の値である．すべての MG において，処理速度は両方を用いる場合が最も速い．このことから，二つの枝刈り手法は効果的であることがわかる．二つの枝刈り手法を組み合わせることで，より高速な処理が行えることが分かる．

7 むすび

本論文では，大規模な系列データベースから，頻度と自己情報量のバランスの取れた非同期パターンを抽出することを目的とし，新たな高速アルゴリズムを提案した．新聞記事コーパスのマイニングを可能とするために，対象系列をアイテム集合系列とし，スライド窓機構を導入することで，非同期的なパターンの抽出した．更に，利用者の目的にあわせて，閾値以上の出現頻度をもつパターンの抽出を可能にした．効率化手法として，I_Max の精度の向上，二重の枝刈りを実装した．枝刈りを二重にすることで処理速度が向上することを実験により示した．

今後の研究として，大塚ら [4] の提案する対数頻度など，情報量とのバランスを取るための出現頻度関数を用いて情報量利得を算出することで，よりノイズの少ないパターンの抽出を目指す．

8 謝辞

本研究の一部は，科学研究費補助金 (No.17300051) の援助を受けている．

参考文献

[1] R. Agrawal and R. Srikant: Mining Sequential Patterns, *Proc. 1995 Int. Conf. on Data Engineering (ICDE'95)*, pp.3–14 1995.

[2] 村田順平, 岩沼宏治, 大塚尚樹, 鍋島英知: 情報量と頻度に基づく系列データマイニングにおける非同期パターンの抽出と効率化，人工知能学会データマイニングと統計数理研究会，SIG-DMSM-A703-10, pp.61–68, 2008.

[3] Jiong Yang, Wei Wang, Philip S.Yu: InfoMiner:Mining Surprising Periodic Patterns, *Data Mining and Knowledge Discovery*, Vol.9, No.2, pp.189-216, 2004.

[4] 大塚尚貴, 岩沼宏治, 鍋島英知, 情報量と頻度に基づく知的系列データマイニング手法, 人工知能学会 データマイニングと統計数理研究会, SIG-DMSM-A603-12, pp.81–88, 2007.

[5] K.Iwanuma,R.Ishihara,Y.Takano,H.Nabeshima: Extracting Frequent Subsequences from a Single Long Data Sequence: A Novel Anti-Monotonic Measure and a Simple On-line Algorithm. *Proc of ICDM 2005*, pp.186-193, 2005.