A-013

巨大イベント系列データの構築を目的としたオンライン型系列マイニングとその高速化

Online sequential mining for building a large corpus of event sequences and its acceleration.

渡井 慎一郎 † Shinichiro Watai
岩沼 宏治 ‡ Koji Iwanuma

1 はじめに

本論文では，一定量のメモリしか利用できないという条件下で頻出部分系列を抽出するオンラインアルゴリズムの高速化手法とその応用例として巨大イベント系列データの構築について考察する．

オンライン型頻出系列マイニングとは，動的なデータストリームから，頻出系列をユーザから要求された時点で即時抽出する技術を言う．データストリームでは絶え間なく連続的にデータが到着するのが特徴であるが，本論文では複数のデータが同時に到着する多重データストリームを取り扱う．オンライン型アルゴリズムは半無限長のデータ系列を扱うため，いかにメモリ使用量を抑えて処理を行うかが最大の課題となる．

本論文では，伊藤らにより提案されたメモリ制限付きオンライン系列マイニング [6] を改良し，周期的削除処理を取り入れた高速な頻出部分系列の抽出アルゴリズムを提案する．新聞記事データベースを用いた評価実験により従来手法よりも高速で処理できることを確認した．実際に構築された巨大イベント系列データが有用なものかを考察する．

本論文の構成は以下の通りである．2 章は準備である．3 章は先行研究について述べ，4 章で提案手法であるオンライン型系列マイニングの周期的削除処理について述べる．5 章で実データを用いた実証実験を通して，従来手法との性能比較や，構築されたイベント系列データの評価を行う．6 章は結論である．

2 準備

本論文で用いる表記法と用語の定義を以下に示す．

定義 1 すべてのアイテムの集合を $I = \{i_1, i_2, \cdots, i_n\}$ とする．**アイテム集合系列** (以下，系列と呼ぶ) とは，アイテム集合の並びであり，$S = \langle s_1 s_2 \cdots s_n \rangle$ と表記する．各 $s_i (s_i \subseteq I, 1 \leq i \leq n)$ を S の**要素**と呼び，$a_1, a_2, \cdots, a_m$ と略記する．S の**系列長** n を $|S|$ で表す．系列 $\alpha = \langle s_1 \cdots s_m \rangle$ が系列 $\beta = \langle t_1 \cdots t_n \rangle$ の**部分系列**であるとは，$s_1 \subseteq t_{j_1}$，$s_2 \subseteq t_{j_2}$，$\cdots$，$s_m \subseteq t_{j_m}$ を満たす整数 $1 \leq j_1 < j_2 < \cdots < j_m < n$ が存在する場合をいい，$\alpha \subseteq \beta$ と表す．

定義 2 **系列データベース**とは，単一のアイテム集合系列である．**アイテム系列**とは，アイテム *1* つからなるアイテム集合の系列である．

本研究では，系列データベースから抽出する部分系列はアイテム系列に限定する．

定義 3 $S = \langle s_1 s_2 \cdots s_n \rangle$ を系列データベース，$\alpha = \langle t_1 t_2 \cdots t_m \rangle$ をアイテム系列とする．S 上の α の**出現頻度関数** $F(S, \alpha)$ を，整数値 $k (0 \leq k \leq |S|)$ を返す関数とする．このとき α の**相対頻度** $R(S, \alpha)$ を以下のように定義する．

$$R(S, \alpha) = \frac{F(S, \alpha)}{|S|}$$

相対頻度 $R(S, \alpha)$ は 1 以下の非負数となる．

定義 4 系列データベース $S = \langle s_1 s_2 \cdots s_n \rangle$ を考える．このとき，部分系列 α が S 中に**頻出**であるとは，与えられた最小相対頻度 $\sigma (0 < \sigma < 1)$ に対し，$R(S, \alpha) \geq \sigma$ が成り立つことをいう．

系列中のアイテム（もしくはアイテム集合）X の出現頻度は，通常は系列データ中の X の単純な出現回数と定義される．一方で部分系列の場合は，単純な出現頻度関数がみつからず，幾つかの工夫が必要である [1, 4]．本論文では，出現頻度関数として**系列先頭頻度** [4] を考える．系列先頭頻度は右逆単調性を持ち，重複数え上げが生じない頻度尺度である [4]．

定義 5 系列データベース $S = \langle s_1 s_2 \cdots s_n \rangle$ があるとき，S の i 番目の要素から始まる長さ w の部分系列を**ウィンドウ**と呼び，$\mathrm{win}(S, i, w)$ と表記し，以下で定義する．また，このときの w を**ウィンドウ幅**という．

$$\mathrm{win}(S, i, w) = \begin{cases} (s_i \cdots s_{i+(w-1)}) & \text{if } i + (w-1) \leq n, \\ (s_i \cdots s_n) & \text{otherwise.} \end{cases}$$

ウィンドウは，データベース中の部分系列を探すときの注目範囲にあたる．

定義 6 $\alpha = \langle a_1 \cdots a_m \rangle$ をアイテム系列，$\beta = \langle b_1 \cdots b_n \rangle$ をアイテム集合系列とするとき，$\alpha \lhd \beta$ を $a_1 \in b_1$ かつ $\alpha \subseteq \beta$ が成り立つ場合と定める．

定義 7 系列データベース $S = \langle s_1 s_2 \cdots s_n \rangle$，アイテム系列 $\alpha = \langle t_1 t_2 \cdots t_m \rangle$，ウィンドウ幅 w $(1 \leq m \leq w < n)$ に対し，S における α の**系列先頭頻度** H-freq(S, α, w) を以下で定義する．

$$\text{H-freq}(S, \alpha, w) = \sum_{i=1}^{n} \lambda(\mathrm{win}(S, i, w), \alpha)$$

ここで λ は以下の関数である．

$$\lambda(\langle s_i \cdots s_n \rangle, \langle t_1 \cdots t_m \rangle) = \begin{cases} 1 & \text{if } \langle t_1 \cdots t_m \rangle \lhd \langle s_1 \cdots s_n \rangle \\ 0 & \text{otherwise} \end{cases}$$

†山梨大学大学院医工農学総合教育部工学専攻コンピュータ理工学コース
‡山梨大学大学院総合研究部

オンライン型データマイニングでは，通常の逆単調性を利用した枝刈りを行うことはできないことに注意して頂きたい．以後，出現頻度関数 F は系列先頭頻度を表すものと約束する．

3 先行研究

本章では先行研究である系列 LC 法 [2] と伊藤らのメモリ制限付き系列 LC 法 [6] について概説する．まず系列 LC 法とはデータストリームから頻出アイテム，もしくは頻出アイテム集合をオンライン近似計算により抽出する Lossy Counting(LC) 法 [3] を拡張し，頻出な部分系列をオンラインで抽出する手法である．最小相対頻度 $\sigma(0 < \sigma < 1)$，頻度の許容誤差 $\epsilon(0 < \epsilon < \sigma)$，ウィンドウ幅 $w(1 < w)$ を受け取り，マイニング対象である系列データベースから相対頻度 σ 以上で長さ w 以下の部分系列（頻出系列）を全てを抽出する．系列 LC 法では幾つかの非頻出な部分系列も抽出するが，読み込んだデータストリームの長さが N である場合，抽出される部分系列の出現頻度は少なくとも $(\sigma-\epsilon)N$ 以上であることが保証される．また，LC 法と同様，相対頻度が ϵ 以下の部分系列を保持しないことで，メモリ使用量の増加を抑制できる．

続いて，メモリ制限付き系列 LC 法について説明する．これは使用するメモリ空間を制限するために頻度表サイズ (登録可能な系列数) に上限値 k を設けた手法である．従来の方法では，頻度表のサイズが上限値に達した際に，データストリームから頻出系列となりうる部分系列が新しく抽出されても頻度表に登録できない構造的欠陥が内在していた．そこで頻度表サイズが上限値に達した際に，表中から頻度が最も低い部分系列を削除し，新たな部分系列を登録可能としている．この手法では，現在時刻 t までに表からあふれて削除された系列の出現頻度で最大なものを**交換頻度** $\delta(t)$ として考慮する．

定義 8 $\Delta(t) = \max(\epsilon(t-1), \delta(t))$

上記の $\epsilon(t-1)$ は，ϵ 近似計算で許容誤差以下の頻度の系列を削除するときの最大の頻度値である．よって，$\Delta(t)$ は時刻 t までに頻度表から削除された系列の頻度の最大値を表す．即ち $\Delta(t)$ は，時刻 t で新たに頻度表に登録される系列の過去の出現頻度の上界となる．よって，もし $\epsilon < \delta(t)$ である場合には，新たに登録される系列の頻度の見積もりは $1 + \delta(t)$ となる．

アルゴリズムが保持する頻度表 D は，四つ組 $(\alpha, C(\alpha), T(\alpha), \Delta(T(\alpha)))$ の集合である．α は登録されたアイテム系列，$C(\alpha)$ は α が D に登録されてからの実頻度，$T(\alpha)$ は α が D に登録された時刻，$\Delta(T(\alpha))$ は α が D に登録される前に出現した可能性のある最大の頻度を表す．以後，$C(\alpha)+\Delta(T(\alpha))$ を**見積り頻度**と呼ぶ．D が保持している四つ組の数を $|D|$ と表記する．保持できる四つ組みの最大数は k，即ち $|D| < k$ と仮定する．

アルゴリズムの概要は以下の通りである．

(1) 現在時刻を t とする．$\delta(t)$ を $\delta(t-1)$ の値で更新，θ を 0 で初期化する．ウィンドウ w を読み，$\alpha \lhd w$ となる全ての系列 α に，以下の 2,3 の処理を行う．

(2) α に対応する四つ組が頻度表 D にあれば，その $C(\alpha)$ を一つ増やす．

(3) (a) α に対応する四つ組が D に無く，$|D| < k$ ならば，D に $(\alpha,\ 1,\ t, \Delta(t))$ を登録する．

(b) α に対応する四つ組が D に無く，$|D| \geq k$ ならば，以下の処理を行う．

i. 頻度表を見積もり頻度で降順にソートする．

ii. D 内から見積り頻度 $C(\beta)+\Delta(T(\beta))$ が最小である部分系列 β(複数ありえる) を抽出する．

iii. A. 抽出した系列 β の中に登録時刻が現在時刻より前，即ち $T(\beta) < t$ なる系列が存在するならば，$\delta(t)$ を $C(\beta)+\Delta(T(\beta))$ で更新する．

B. 抽出した全ての β の登録時刻が現在時刻，即ち $T(\beta) = t$ ならば，θ を $C(\beta)+\Delta(T(\beta))$ で更新する．

iv. 抽出した全ての β の四つ組を D から削除し，代わりに $(\alpha,\ 1,\ t, \Delta(t))$ を登録する．

(4) $\delta(t) < \theta$ であるならば $\delta(t)$ を θ で更新し，D から $C(\alpha)+\Delta(T(\alpha)) \leq \max(\epsilon t, \delta(t))$ を満たす系列 α の四つ組を削除する．

(5) 時刻を 1 進め，ウィンドウをスライドさせる．

出力要請を受けた場合は，$C(\alpha)+\Delta(T(\alpha)) \geq \sigma t$ となる部分系列 α を頻出とみなして出力する．

ステップ (3)-(b)-iii では交換により削除する系列が全て現在時刻に登録したものであるときに交換頻度の冗長計算を防ぐための処理である [6]．上記の状態では $\delta(t)$ を更新せず，**補助頻度** θ に一時保持し，抽出された系列を全て処理し終わった後に $\delta(t)$ を θ で更新することで，$\delta(t)$ が表から削除された系列中で最大出現頻度であることを保証している．

このアルゴリズムは，実行時に交換頻度が最小相対頻度を一度も超えない場合に頻出系列抽出の完全性が保証される [6]．また，完全性が保証できない場合でも一定の保証 (擬似完全性と呼ぶ) を行うことができ，どちらの場合も抽出系列の頻度に関して誤差の上界が保証できる [6]．

4 提案手法

本章では，メモリ制限付き系列 LC 法を周期的に削除することによる高速化を提案する．先行研究ではステップ (4) を毎サイクルごとに近似削除処理を行っているため，毎回の削除処理により時間がかかる．そこで，近似削除処理を $\frac{1}{\epsilon}$ の周期で行うことで高速化を図る．アルゴリズムは Algorithm 1 のようになっている．3 章で紹介したアルゴリズムに加え，16〜22 行目のように周期削除を追加した．

4.1 時間計算量

提案手法の時間計算量を考える．このアルゴリズムではウィンドウから抽出した部分系列が頻度表にあるかを確認するためにハッシュ法を使用し，最小頻度の部分系列を削除するためのソートはクイックソートで実装されている．データストリームの長さを N，各時刻で抽出するアイテム系列の個数の最大値を $n(n \leq L)$，頻度表のサイズ制限値を L とする．

ある時刻 t における処理は，ストリームから抽出したある系列 α に対して，(1) 頻度表における α の存在

Algorithm 1 ストリームのマイニングアルゴリズム

Input: $S = \langle A_1, A_2, \cdots A_N \rangle$:ストリーム, ϵ:許容誤差,
σ:最小相対頻度, $window\_size$:ウィンドウサイズ,
k:頻度表制限値
Output: 部分系列の頻度表

```
%% 使用する大域変数 : t                    ▷ 時刻情報の初期化
%%                  : δ(t)                 ▷ 交換頻度
%%                  : θ                    ▷ 補助頻度

Table                                      ▷ 頻度表を保持する構造体
%%  α                                      ▷ 部分系列
%%  C(α)                                   ▷ 登録されてからの頻度
%%  T(α)                                   ▷ 登録された時刻
%%  Δ(T(α))                                ▷ 部分系列の最大頻度誤差

Initialize(W)                              ▷ 部分系列を抽出するためのウィンドウ
t ← 1                                      ▷ 時間情報の初期化
δ(t) ← 0                                   ▷ 初期化
θ ← 0                                      ▷ 初期化

for t ≤ N do
    W ←MakeWindow(S, t, window_size)
                                           ▷ ストリームからウィンドウを切り出す関数
    Update(Table, W, k, θ)                 ▷ 3 章の (1)(2)(3)
    if t % ⌊(1/ϵ)⌋ = 0 then                ▷ 周期削除処理 (追加)
        Reduce(Table, ⌊ϵt⌋)                ▷ 3 章の (4)
        if θ > δ(t) then                   ▷ 交換頻度の更新
            δ(t) ← θ
            θ ← 0
        end if
    end if
    t++                                    ▷ 3 章の (5)
end for
FreeStream()                               ▷ ウィンドウを破棄
```

確認とカウンタ処理，(2) 存在しない場合の頻度表の更新 (表から頻度最小の系列エントリの削除を含む)，(3) 周期削除処理するために表中の系列エントリを走査し，頻度が ϵt 以下ならば削除する．

従って，時刻 t における計算量は (1) と (2) の処理は $n \leq L$ より $O(n + L\log L + L) \approx O(L \log L)$，(3) の走査に $O(L)$ となる．(1)，(2) は毎サイクルごと行われるため，$O(N \times (L \log L))$ となる．(3) は周期的に行われるので $O(\epsilon N \times L)$ となるため，全体の時間計算量は

$$\begin{aligned} &O(NL \log L) + O(\epsilon NL) \\ &= O(NL \log L + \epsilon NL) \\ &= O(LN \log L) \end{aligned}$$

となる．

5 評価実験

提案手法を C 言語で実装し，系列データベースから頻出系列を全て抽出する実験を行った．対象データは 2000〜08 年までの毎日新聞のスポーツ記事の見出しの単語を TF-IDayF 法 [5] により一日単位で各単語に重み付けし，日単位で上位 20 単語を抽出し，日付順に並べたアイテム集合系列である．データ長 3,280，アイテム (重要単語) 種類数 1,319，各日に出現するアイテム集合サイズは均一で 20 である．実験時の各パラメータはウィンドウ幅 3，最小相対支持度 σ=0.01，許容誤差 $\epsilon = 0.001$ とする．先行研究と比較し，高速化の程度と精度を確認する．

5.1 時間

図 1 は先行研究と比較した結果である．提案手法が全体的に速くなっていることがグラフから読み取れる．先行研究では頻度表サイズが 7000〜8000 万で急激に時間が短くなり，8000 万以上では一定になっている．これは頻度表サイズが十分大きいため，サイズ制限削除が行われなることで速くなる．

提案手法では周期的近似削除のため，先行研究よりも余分に頻度表を使ってしまうが，頻度表サイズが大きくなるにつれ，サイズ制限削除数が減り，一定になることが分かる．

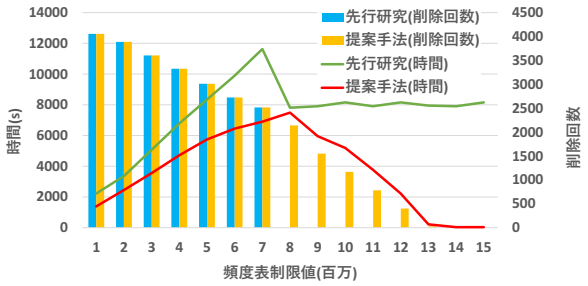

図 1: 提案手法における頻度表サイズの時間推移

5.2 精度

精度の評価には適合率と再現率を用いた．頻度表制限値なし，許容誤差 ϵ=0 で抽出された系列の集合を正解とし，制限値，許容誤差をつけて抽出された系列集合の再現率と適合率を以下で定義する．

$$\text{再現率} = \frac{\text{抽出したうちの正しい系列の数}}{\text{正解の系列数}}$$

$$\text{適合率} = \frac{\text{抽出したうちの正しい系列の数}}{\text{抽出データの系列数}}$$

表 1 は提案手法における頻度表制限値を変化させた場合の再現率などの変化を示している．抽出結果が完全となる再現率 100 %である制限サイズの下限は 100 万程であることが分かった．よって，頻度表制限値なし，許容誤差 ϵ=0 の場合よりも約 $\frac{1}{20}$ 倍に頻度表サイズを抑制している．また，最小頻度 (σt) より最大交換頻度が小さい場合に再現率 100 %となっていることから先行研究の完全性が提案手法にも保証されていることが分かった．

適合率について，先行研究同様，最小頻度 (σt) より最大交換頻度が小さいサイズ制限 100 万以上は高い値を示している．最小頻度 (σt) より最大交換頻度が大きい場合，頻度表にある全ての系列が頻出系列とみなされるため，急激に落ちている．

頻度表サイズが 700 万以下では先行研究と提案手法の頻度表最終サイズが同じになっている．これは，毎サイクルごとにサイズ上限削除処理が起きており，許容誤差が小さい故にサイズ上限削除処理によって周期的削除できる系列が残っていないためだと考えられる．また，頻度表サイズ 900 万以上では頻度表サイズが十分に大きいため，サイズ上限削除処理が行われなくなり，先行研究と提案手法で最終サイズが変わると考えられる．

表 1: 提案手法における頻度表制限値と再現率などの変化

頻度表	抽出系列数		頻度表最終サイズ		最終交換	再現率 (%)		適合率 (%)	
最大値	先行研究	提案手法	先行研究	提案手法	頻度	先行研究	提案手法	先行研究	提案手法
なし	11,548	-	20,092,732	-	0	-	-	-	-
10,000,000	11,563	115,63	7,281,659	8,880,819	0	100	100	99.87	99.87
9,000,000	11,563	11,563	7,281,659	8,880,819	0	100	100	99.87	99.87
8,000,000	11,563	12,191	7,281,659	7,995,989	0	100	100	99.87	94.73
7,000,000	11,563	12,191	6,996,289	6,996,289	3.39	100	100	99.87	94.73
6,000,000	11,563	12,195	5,996,154	5,996,154	4.03	100	100	99.87	94.69
5,000,000	11,583	12,319	4,999,973	4,999,973	4.65	100	100	99.70	93.74
4,000,000	11,615	11,615	3,993,410	3,993,410	6.16	100	100	99.42	99.42
3,000,000	11,747	12,524	2,996,683	2,996,683	8.24	100	100	98.30	92.21
2,000,000	12,175	12,946	1,996,732	1,996,732	13.05	100	100	94.85	89. 20
1,000,000	17,584	20,146	996,763	996,763	27.01	100	100	65.67	57.32
800,000	796,737	796,737	796,737	796,737	33.10	99.69	99.69	1.44	1.44
600,000	595,402	595,402	595,402	595,402	45.04	92.01	92.01	1.93	1.93

注) 処理終了時の最小頻度 (σt) は 32.80 である

5.3 評価

表 2 はオンライン型系列マイニングによって抽出されたアイテム系列の例である．毎日新聞記事コーパスからウィンドウ幅 3，つまり 3 日のウィンドウ幅で抽出した高頻出なアイテム系列パターンである．例として，頻出部分系列の中の上位と下位から各 10 個ずつを取り出している．上位 10 個は単一アイテムがほとんどを占めており，アイテム系列パターンとして有益なものではないと考えられる．下位 10 個は様々な単語から構成された部分系列が抽出されており，3 日間を通して何が起きているか容易に想像することができる．また，実際に新聞記事にはオリンピックや選抜高校野球大会の記事が連日記載されていることが確認できる．

表 2: 抽出されたアイテム系列の例

上位 10 個	下位 10 個
〈 大阪 〉	〈 センバツ ＪＯＣ ノルディック 〉
〈 スキー 〉	〈 スペイン 選抜 土曜 〉
〈 カップ 〉	〈 パ・リーグ カップ 賞 〉
〈 スキー スキー 〉	〈 春 距離 スキー 〉
〈 選抜 〉	〈 準決勝 カップ 東都 〉
〈 スペイン 〉	〈 特別 大阪 毎日 〉
〈 賞 〉	〈 各 カップ 〉
〈 毎日 〉	〈 大阪 韓国 賞 〉
〈 準決勝 〉	〈 合意 日曜 大阪 〉
〈 清水 〉	〈 販売 大阪 〉

6 まとめ

本論文ではメモリ制限付きオンライン型頻出系列マイニングに周期的削除処理を取り入れる改良について提案した．提案手法は先行研究に比べて高速かつ同程度でメモリ使用量を削減できることを示した．また，応用例として頻出部分系列によって構築されたアイテム系列データでは中下位に存在する部分系列からある程度イベントを読み取ることができた．

謝辞

本研究は一部，JSPS 科学研究費補助金（基盤 C：No.19K12096）の援助を受けている．

参考文献

[1] A. Achar, S. Laxman and P. S. Sastry: A unified view of the apriori-based algorithms for frequent episode discovery *Knowl. Inf. Syst*(KAIS), DOI 10.1007/s10115-011-0408-2 (2011)

[2] 村田順平, 岩沼宏治, 鍋島英知: 精度保証付きオンライン型高速近似系列マイニング, 第 8 回情報科学技術フォーラム (FIT2009) 講演論文集, F-043 (2009)

[3] G. S Manku and R. Motwani: Approximate frequency counts over data streams. *Proc VLDM'02*, pp.346–357 (2002)

[4] K. Iwanuma, R. Ishihara, Y. Takano and H. Nabeshima: Extracting frequent subsequences from a single long data sequence: a novel anti monotonic measure and a simple on-line algorithm. *Proc. of IEEE Inter. Conf. on Data Mining* (ICDM 2005), pp.186–193 (2005)

[5] 多田知道, 岩沼宏治，鍋島英知: イベント系列マイニングを目的とする新聞記事からの時間情報に基づく単語抽出．人工知能学会論文誌，24 巻 6 号 p. 488-493(2009)

[6] 岩沼宏治，山本泰生，伊藤秀志: ストリームデータ上の各種の頻出データのオンライン型マイニングに関する一考察．電子情報通信学会技術研究報告 = IEICE technical report : 信学技報 113(332), 83-88 (2013)