イベント列からの頻出菱形エピソードの多項式遅延・多項式領域抽出

A Polynomial-Delay Polynomial-Space Algorithm for Extracting Frequent Diamond Episodes from Event Sequences

河東 孝[*1] 有村 博紀[*2] 平田 耕一[*3]
Takashi Katoh Hiroki Arimura Kouichi Hirata

[*1][*2]北海道大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

[*3]九州工業大学大学院 情報工学研究院
Department of Artificial Intelligence

In this paper, we study the problem of mining *frequent diamond episodes efficiently* from an input event sequence with sliding a window. Here, a diamond episode is of the form $a \mapsto E \mapsto b$, which means that every event of E follows an event a and is followed by an event b. Then, we design a polynomial-delay and polynomial-space algorithm PolyFreqDmd that finds all of the frequent diamond episodes without duplicates from an event sequence in $O(|\Sigma|^2 n)$ time per an episode and in $O(|\Sigma| + n)$ space, where Σ and n are an alphabet and the length the event sequence, respectively. Finally, we give experimental results on artificial event sequences with varying several mining parameters to evaluate the efficiency of the algorithm.

1. はじめに

データマイニングの一種である**系列マイニング** [2] は，系列データから頻出する**部分系列**を抽出する手法である．一方，Mannila ら [8] による**エピソードマイニング** [8] は，部分系列ではなく，イベント列に同時に出現するイベントの構造である**エピソード**のうち頻出するものを抽出する．

河東らは，エピソードの部分クラスである**菱形エピソード** [7] に対して，イベント列からすべての頻出菱形エピソードを抽出するアルゴリズム FreqDmd [7] を示した．FreqDmd は，頻出菱形エピソードの発見問題を頻出アイテム集合の発見問題に帰着することで，イベント列から効率よく頻出菱形エピソードを抽出する．一方，FreqDmd の時間計算量と領域計算量は示されていない．そこで，本論文では，イベント列からすべての頻出菱形エピソードを，多項式遅延・多項式領域で抽出するアルゴリズムとして，新たに PolyFreqDmd を示す．ここで，多項式遅延とは，アルゴリズムが，1 つの解を出力してから次の解を出力するまでの計算時間が入力の大きさの多項式でおさえられることをいう．次に，アルゴリズム PolyFreqDmd にいくつかの技術的な工夫を加えて，アルファベッΣ と入力イベント列の長さ n に対して，$O(|\Sigma|^2 n)$ 遅延，$O(|\Sigma| + n)$ 領域で動作するアルゴリズムを示す．最後に，人工的に作成したイベント列に対して，PolyFreqDmd を適用することで，アルゴリズムの性能を評価する．

2. 菱形エピソード

有限のアルファベット $\Sigma = \{1, \ldots, m\}$ $(m \geq 1)$ に対して，アルファベットの要素を $e \in \Sigma$ **イベント**という．Σ 上のイベント集合の有限列 $\mathcal{S} = \langle S_1, \ldots, S_n \rangle \in (2^\Sigma)^*$ を**入力イベント列**という．このとき，添字 $1 \leq i \leq n$ を**出現時刻**といい，n を $\mathcal{S}$ の長さという．

イベントの集合 E とイベント a, b に対して，$a \mapsto E \mapsto b$ を**菱形エピソード**という．これは，E 中のすべてのイベントは a の後，b の前に出現することを意味する．

入力イベント列 $\mathcal{S}$ と整数 t_s，t_e に対して，出現時刻 t が，$t_s \leq t < t_e$ を満たす $\mathcal{S}$ の部分列を**ウィンドウ**といい，$t_e - t_s$ をそのウィンドウの**幅**という．

入力イベント列 $\mathcal{S}$ 上の幅 k のすべてのウィンドウの集合を $\mathbf{W}_{\mathcal{S},k}$ と書く．そのうち，エピソード D が出現するウィンドウを $\mathbf{W}_{\mathcal{S},k}(D)$ と書く．このとき，菱形エピソード D の**頻度**と**支持度**をそれぞれ，

$$freq(D) = |\mathbf{W}(D)|, \quad supp(D) = \frac{|\mathbf{W}(D)|}{|W(k)|}$$

と定義する．菱形エピソード D と**最小頻度** $\sigma (0 < \sigma)$ に対して，$freq(D) \geq \sigma$ となるとき，D は**頻出である**という．本論文では，$\mathbf{W}$ の添字は自明な場合省略する．さらに，頻度と支持度は相互に変換可能なので同一視する．

3. 多項式遅延多項式領域アルゴリズム

アルゴリズムが，1 つの解を出力してから次の解を出力するまでの計算時間が入力の大きさの多項式でおさえられるとき，そのアルゴリズムを**多項式遅延**アルゴリズムという．本論文では，入力イベント列からすべての頻出菱形エピソードを多項式遅延・多項式領域で抽出するアルゴリズム PolyFreqDmd を示す．ここで，$\mathcal{S} = (S_1, \ldots, S_n) \in (2^\Sigma)^*$ を長さ n の入力イベント列，$k \geq 1$ をウィンドウ幅，$\sigma \geq 1$ を最小頻度とする．

図 1 にアルゴリズム PolyFreqDmd の概要を示す．アルゴリズム PolyFreqDmd は，すべての頻出菱形エピソードを含む探索空間を深さ優先探索するアルゴリズムである．すべてのイベントの組 $(a, b) \in \Sigma$ に対して，PolyFreqDmd は，最小の菱形エピソード $D_{ab} = (a \mapsto \emptyset \mapsto b)$ と D_{ab} の出現するウィンドウの集合 $\mathbf{W}(D_{ab})$ を引数にして，再帰関数 FreqDmdRec を呼び出すことで深さ優先探索を開始する．FreqDmdRec は，

連絡先: 氏名: 河東 孝, 所属: 北海道大学 大学院情報科学研究科 CS 専攻, 住所: 〒 060-0814 札幌市北区北 14 条西 9 丁目, 場所: 情報科学研究科棟 7 F，情 706 号室, TEL: 011-706-7680, FAX: 011-706-7680, Email: t-katou@ist.hokudai.ac.jp

algorithm PolyFreqDmd($\mathcal{S}, k, \Sigma, \sigma$)
input: 長さ n の入力イベント列 $\mathcal{S} \in (2^{\Sigma})^*$,
ウィンドウ幅 $k > 0$, アルファベット Σ,
最小支持度 $1 \leq \sigma \leq n + k$;
output: 頻出菱形エピソード;
{
1 $\Sigma_0 :=$ 頻出するイベントの集合 ($\Sigma_0 \subseteq \Sigma$);
2 **foreach** ($a \in \Sigma_0$) **do**
3 **output** a;
4 **foreach** ($b \in \Sigma_0$) **do**
5 $D_0 := (a \mapsto \emptyset \mapsto b)$; //直列エピソード
6 $W_0 := W_{\mathcal{S},k}(D_0)$; // D_0 が出現するウィンドウの集合
7 FreqDmdRec($D_0, W_0, \mathcal{S}, k, \Sigma_0, \sigma$);
8 **end for**
}
procedure FreqDmdRec($D = (a \mapsto E \mapsto b), W, \mathcal{S}, k, \Sigma, \sigma$)
output: $a \mapsto E \mapsto b$ という形のすべての頻出菱形エピソード;
{
1 **if** ($|W| \geq \sigma$) **then**
2 **output** D; // **output** D 深さが偶数の場合 (**交代出力**);
3 **foreach** ($e \in \Sigma$ ($e > \max(E)$))) **do**
4 $C = a \mapsto (E \cup \{e\}) \mapsto b$;
5 $U := W_{\mathcal{S},k}(C)$
6 FreqDmdRec($C, U, \mathcal{S}, k, \Sigma, \sigma$);
7 **end for**
8 // **output** D 深さが奇数の場合 (**交代出力**);
9 **end if**
}

図 1: イベント列からすべての頻出菱形エピソードを出力するアルゴリズム PolyFreqDmd と再帰手続き FreqDmdRec

入力された菱形エピソード $D = (a \mapsto E \mapsto b)$ が頻出菱形エピソードのとき，D を出力する．さらにそのとき，$e > \max(E)$ となる任意のイベント $e \in \Sigma$ に対して，E に e を加えることで菱形エピソード D を更新するとともに，D の出現するウィンドウ集合を計算する．以上の手続きを再帰的に繰り返すことで，任意のウィンドウ幅 $k > 0$ と任意の最小頻度 σ に対して，アルゴリズム PolyFreqDmd は，入力イベント列 $\mathcal{S}$ から頻出菱形エピソードを重複無く出力する．

菱形エピソード $D = (a \mapsto E \mapsto b)$ と任意のイベント e，エピソード D が出現するウィンドウの集合 $\mathbf{W}(D)$ に対して，FreqDmdRec は，直列エピソード $(a \mapsto e \mapsto b)$ の出現するウィンドウを計算することで，菱形エピソード $D' = (a \mapsto E \cup \{e\} \mapsto b)$ と D' が出現するウィンドウの集合 $\mathbf{W}(D')$ を E の大きさに依存しない計算時間で**高速更新**する．このとき，D と D', $\mathbf{W}(D)$ と $\mathbf{W}(D')$ の差分のみを**差分リスト**として保存する．さらに，再帰の深さが偶数のときは先順出力，奇数のとき後順出力となるように**交代出力**を行う．FreqDmdRec は，直列エピソードの出現するウィンドウを複数回計算する．そこで，一度計算した直列エピソードの出現位置を記録して，二回目以降に利用する**動的計画法** を利用することで，アルゴリズム全体の高速化を行う．

動的計画法以外の手法を組み合わせた場合，アルゴリズム PolyFreqDmd は，アルファベット Σ と長さ n の入力イベント列 $\mathcal{S}$，ウィンドウ幅 $k \geq 1$，最小頻度 $\sigma \geq 1$ に対して，すべての頻出菱形エピソードを $O(|\Sigma|^2 n)$ 遅延・$O(|\Sigma| + n)$ 領域で出力する．

4. 実験結果

本論文では，アルゴリズム PolyFreqDmd を以下のような手法と組み合わせて実装し，乱数を用いて人工的に作成した入力イベント列に，これらのアルゴリズムを適用して性能を評価した．

DF	:	PolyFreqDmd.
DF-SWO	:	DF と交代出力 (SWO).
DF-FFS	:	DF と高速更新 (FFS).
DF-DIFF	:	DF と差分リスト (DIFF).
DF-DP	:	DF-FFS と動的計画法 (DP).

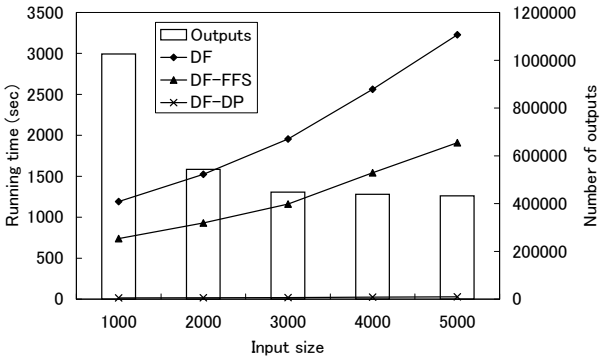

図 2: アルファベットサイズ $s = 20$，ウィンドウ幅 $k = 10$，最小頻度 $\sigma = 0.1n$ における，入力イベント列の長さ n に対する計算時間と出力数．

図 2 に，アルファベットサイズとウィンドウ幅，最小支持度をそれぞれ，$s = 20$ と $k = 10$，$\sigma = 0.1n$ としたときの，アルゴリズム DF と DF-FFS，DF-DP の計算時間と出力数を示す．図 2 より，アルゴリズム DF-FFS は DF より 2 倍程度高速に動作し，アルゴリズム DF-DP は DF より 100 倍程度高速に動作することが分かった．一方，この入力に対して，アルゴリズム DF-SWO と DF-DIFF のアルゴリズム DF に対する計算時間の違いは確認できなかった．さらに，これらのアルゴリズムの計算時間が，入力イベント列の長さに対して線形に増加していることが分かった．

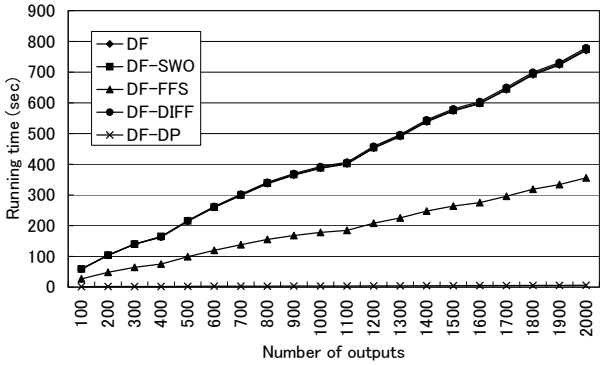

図 3: アルファベットサイズ $s = 20$，入力イベント列の長さ $n = 10,000$，ウィンドウ幅 $k = 30$，最小頻度 $\sigma = 0.3n$ における，出力数に対する計算時間．

図 3 にアルファベットサイズと入力イベント列の長さ，ウィンドウ幅，最小支持度をそれぞれ，$s = 20$ と $10,000$，$k = 30$，$\sigma = 0.3n$ としたときの，アルゴリズム DF と DF-FFS，DF-DP の計算時間と出力数を示す．図 3 より，これらのアルゴリズムが出力数に対して線形の計算時間で動作することが分かった．したがって，これらのアルゴリズムの遅延が線形であることが予想できた．

これらの実験結果より，実験で使用したデータに対して，アルゴリズム PolyFreqDmd を動的計画法を利用して実装することが非常に有効であることが分かった．また，出現集合の高速更新手法を利用すると，計算時間が 2 倍程度高速になることが分かった．

5. まとめ

本論文では，イベント列からすべての頻出菱形エピソードを抽出する問題を多項式遅延・多項式領域で解くアルゴリズム PolyFreqDmd を示した．さらに，アルゴリズムの計算時間を削減するためのいくつかの手法を示した．

アルゴリズム PolyFreqDmd を一般のエピソード [8, 9] に拡張することと，菱形エピソードに対して，閉パターン [3, 4, 9, 12] を定義し，効率のよいアルゴリズムを示すことは今後の課題である．さらに，アルゴリズム PolyFreqDmd を実際の細菌検査データ [6, 7] に適用することは重要な課題である．

参考文献

[1] R. Agrawal, R. Srikant: Fast algorithms for mining association rules in large databases, *Proc. 20th VLDB*, 487–499, 1994.

[2] R. Agrawal, R. Srikant: *Mining sequential patterns*, Proc. 11th ICDE, 3–14, 1995.

[3] H. Arimura: Efficient algorithms for mining frequent and closed patterns from semi-structured data, *Proc. PAKDD'08*, LNAI 5012, 2–13, 2008.

[4] H. Arimura, T. Uno: A polynomial space and polynomial delay algorithm for enumeration of maximal motifs in a sequence, *Proc. ISAAC'05*, LNCS 3827, 2005.

[5] D. Avis, K. Fukuda: Reverse search for enumeration, *Discrete Applied Mathematics* **65**, 21–46, 1996.

[6] T. Katoh, K. Hirata: Mining frequent elliptic episodes from event sequences, *Proc. 5th LLLL*, 46–52, 2007.

[7] T. Katoh, K. Hirata, M. Harao: Mining frequent diamond episodes from event sequences, *Proc. 4th MDAI*, LNAI 4617, 477–488, 2007.

[8] H. Mannila, H. Toivonen, A. I. Verkamo: Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery* **1**, 259–289, 1997.

[9] J. Pei, H. Wang, J. Liu, K. Wang, J. Wang, P. S.. Yu: Discovering frequent closed partial orders from strings, *IEEE TKDE* **18**, 1467–1481, 2006.

[10] J. Pei, J. Han, B. Mortazavi-Asi, J. Wang, H. Pinto, Q. Chen, U. Dayal, M.-C. Hsu: Mining sequential patterns by pattern-growth: The PrefixSpan approach, *IEEE Trans. Knowledge and Data Engineering* **16**, 1–17, 2004.

[11] T. Uno: Two general methods to reduce delay and change of enumeration algorithms, NII Technical Report, NII-2003-004E, April 2003.

[12] M. J. Zaki, C.-J. Hsiao: CHARM: An efficient algorithm for closed itemset mining, *Proc. 2nd SDM*, 457–478, SIAM, 2002.