The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

イベント系列からの有意性を考慮した菱形エピソードマイニング

Mining Significant Diamond Episodes from Event Sequences

谷 陽太 *1 Yota Tani　古谷 勇 *1 Isamu Furuya　平田 耕一 *2 Kouichi Hirata　有村 博紀 *1 Hiroki Arimura

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

*2九州工業大学大学院 情報工学研究院
Department of Artificial Intelligence, Kyushu Institute of Technology

In this paper, we consider the framework of statistically significant ranking for frequent episodes, proposed by Tatti (DMKD, 2015), where an episode is a structured pattern in the shape of vertex labeled directed acyclic graphs. For the class of diamond episodes, we present an algorithm, called MineRankedDmd, that generates frequent episodes with their ranking in a collection of event sequences. In the experiments on synthetic data set, we evaluated the performance of the proposed 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 [1]。一方、Mannila ら [4] によるエピソードマイニングは、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、頻出なものを抽出する手法である。

Tatti [5] は、エピソードの部分クラスである**厳密エピソード**に対して、**ランク**の定義を与えている。ランクは、イベント列の集合と厳密エピソードが与えられた場合に、イベント列の集合におけるエピソードの出現確率と、実際のエピソードの出現数から、そのエピソードの有意性を測る指標である。一方で、Tatti [5] は厳密エピソードを列挙する手法は与えていない。

Katoh ら [3] は、厳密エピソードの部分クラスである**菱形エピソード**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効率的に抽出するアルゴリズム PolyFreqDmd を提案している。本稿では、イベント列の集合からすべての頻出菱形エピソードを、Katoh ら [3] の手法にしたがって列挙すると同時に、解となる頻出菱形エピソードのランクを Tatti [5] の手法に基づいて計算するアルゴリズム MineRankedDmd を与える。最後に、人工的に生成したイベント列にアルゴリズム MineRankedDmd を適用することで、アルゴリズムの性能を評価する。

2. 準備

本節では、Tatti [5] にしたがって、厳密エピソードと呼ばれるエピソードの部分族におけるランク計算の枠組みを導入する。整数 $i \leq j$ に対して、$[i..j] = \{i, i+1, \ldots, j\}$ は i から j までの整数区間を表す。

2.1 イベント列とエピソード

アルファベット Σ に対して、その要素 $e \in \Sigma$ を**イベント**という。Σ 上の長さ k の**イベント列**は、$S = \langle s_1, \ldots, s_k \rangle \in \Sigma^*$ である。ここに、$s_i \in \Sigma$ はイベントである。アルゴリズムの入力は、イベント列集合 $\mathcal{S} = \{S_1, \ldots, S_{|\mathcal{S}|}\} \subseteq \Sigma^*$ である。*1

エピソード [4] は、頂点ラベルつき非巡回有向グラフ $G = (V_G, E_G, lab_G)$ である。ここに、V_G は頂点集合、E_G は辺集合、lab_G はラベル関数 $lab_G : V_G \to \Sigma$ である。エピソードは、出現の半順序関係が定義されたイベントの多重集合とみなせる。

エピソード G がイベント列 $S = \langle s_1, \ldots, s_n \rangle$ に**出現する**ことを $S \succeq G$ で表し、次の条件を満たす関数（マッチング関数と呼ぶ） $f : V_G \to [1..n]$ が存在することと定める：(i) f は一対一である: $(\forall u, v \in V_G)\ f(u) = f(v) \ \Rightarrow\ u = v$; (ii) f は順序を保存する: $(\forall u, v \in V_G)\ (u, v) \in E_G \ \Rightarrow\ f(u) < f(v)$; (iii) f は頂点ラベルを保存する: $(\forall v \in V_G)\ lab_G(v) = s_{f(v)}$. このとき、イベント列 S はエピソード G を**カバーする**ともいう。

イベント列の集合 $\mathcal{S}$ 上で、エピソード G を**カバーするイベント列の集合**を $\mathbf{C}_{\mathcal{S}}(G) = \{\, S \in \mathcal{S} \mid S \succeq G \,\}$ と書く。このとき、G の**観測支持度**（単に**支持度**とも呼ぶ）を $sup(G \mid \mathcal{S}) = |\mathbf{C}_{\mathcal{S}}(G)|$ と定義する。さらに、エピソード G と**最小頻度** $\alpha > 0$ に対して、$sup(G \mid \mathcal{S}) \geq \alpha$ となるとき、G は**頻出である**という。本稿では、$\mathbf{C}$ の添字 $\mathcal{S}$ は自明な場合省略する。

2.2 エピソードの部分族

すべての要素間に順序関係が定義されている場合、そのエピソードは**直列である**という [4]。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** [3] 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。以降、a と b を先頭と末尾と呼び、E を本体と呼ぶ。単に $a \mapsto b$ で、b が a の後に出現することを表す。

一般に与えられたイベント列 S とエピソード G に対して、カバー $S \succeq G$ の判定は NP 困難である [5]。この困難性は主にマッチング関数の一対一性に起因する。そのため、Tatti [5] は、次の厳密エピソードを導入した。*2

連絡先: 谷 陽太，北海道大学大学院 情報科学研究科，〒 060-0814 札幌市北区北 14 条西 9 丁目，TEL/FAX: 011-706-7680，Email: tani@ist.hokudai.ac.jp

*1 Mannila ら [4] は、1 つのイベント系列 S と正整数 k を入力とし、長さ k の滑り窓モデルでのエピソード発見を議論している。Mannila らの枠組みは、S 中の長さ k のすべての部分文字列からなる系列集合を考えることで、本稿の枠組みで扱える。

*2 Katoh と Hirata [2] は、Tatti [5] と独立に厳密エピソードの族を提案し、線形エピソードによるそれらの特徴付けを与えている。

定義 1 (厳密 [5])**.** エピソード $G = (V_G, E_G, lab_G)$ が**厳密**であるとは、任意の異なる頂点 u と $v \in V_G$ に対して、$lab_G(u) = lab_G(v)$ ならば、u から v への有向パス、または v から u への有向パスが存在することをいう。

Tatti [5] の定義では、エピソードの DAG の推移閉包を考えて、二頂点が同じラベルをもつならば両者をつなぐ有向辺があることと定義している。これは本稿の定義と同値である。上記の直列エピソードと菱形エピソードは、いずれも厳密エピソードである。*3

2.3 エピソードのカバー確率とランク

本稿では、イベント列のランダム生成モデルとして、イベントの生起確率のベクトル $\mathbf{q} = (q_e)_{e\in\Sigma} \in [0,1]^{|\Sigma|}$ で定められる記憶のない情報源を仮定する。ここに、$\sum_e q_e = 1$ である。

エピソード G とイベント列生成モデル $\mathbf{q}$ に対して、G **のカバー確率**とは、$\mathbf{q}$ にしたがって生成された長さ k のランダムなイベント列 X が G をカバーする確率

$$p_k = p(G \mid \mathbf{q}, k) = p(X \succeq G \mid |X| = k, X \sim \mathbf{q})$$

である。このとき、G の**期待支持度**は $\mu = \sum_{S\in\mathcal{S}} p_{|S|}$ となる。

イベント列の集合 $\mathcal{S}$ に対して、$|X_i| = |S_i| (1 \le i \le |\mathcal{S}|)$ となるようなランダムなイベント列の集合 $\mathcal{X}$ を考える。非負整数 $n \ge 0$ に対して $sup(G \mid \mathcal{X}) = n$ となる確率は、

$$p(sup(G \mid \mathcal{X}) = n \mid \mathcal{X} \sim \mathbf{q}) = \sum_{\substack{\mathcal{T}\subseteq\mathcal{S} \\ |\mathcal{T}|=n}} \prod_{S\in\mathcal{T}} p_{|S|} \prod_{S\in\mathcal{S}\setminus\mathcal{T}} (1 - p_{|S|})$$

である。この値は n が十分に大きいとき、平均 μ で分散 $\sigma^2 = \sum_{S\in\mathcal{S}} p_{|S|}(1 - p_{|S|})$ の正規分布 $N(\mu, \sigma)$ から推定できる。以降、この値を求める場合、正規分布から推定することとする。

Tatti [5] は、イベント列の集合 $\mathcal{S}$ とイベント列生成モデル $\mathbf{q}$ において、エピソード G の**有意度ランク** (単に**ランク**とも呼ぶ) を次のように定義している。

定義 2 (有意度ランク [5])**.** イベント列の集合 $\mathcal{S}$ とイベント列生成モデル $\mathbf{q}$ において、エピソード G のランクとは、

$$\begin{aligned} r(G \mid \mathcal{S} \sim \mathbf{q}) &= -\log p\big(sup(G \mid \mathcal{X}) \ge n \mid X \sim \mathbf{q}\big) \\ &= -\log\Big(1 - \sum_{k=1}^{n-1} p\big(sup(G \mid \mathcal{X}) = k \mid X \sim \mathbf{q}\big)\Big) \end{aligned}$$

である。ここで、$0 \le r(G \mid \mathcal{S} \sim \mathbf{q}) \le \infty$ である。

2.4 エピソードからの有限オートマトンの構築

Tatti [5] は、イベント列がエピソード G をカバーするかを、有限オートマトンを用いて判定する方法を提案している。さらに、この有限オートマトンをカバー確率と G のランクの計算にも用いる。

ここに、エピソードは有向グラフであることに注意されたい。エピソード $G = (V, E)$ に対して、部分エピソード $H = (W, F)$ が G の**接頭辞部分グラフ**であるとは、それが条件 (i) $v \in W$ かつ $(w, v) \in E$ ならば $w \in W$、かつ (ii) $v, w \in W$ かつ $(v, w) \in E$ ならば $(v, w) \in F$ を満たすことをいう。G **の接頭辞部分グラフ全体の集合**を $Pre(G)$ で表す。例として、$G = a \mapsto \{b, c\} \mapsto d$ のとき、$Pre(G) = \{(), (a), (a \mapsto b), (a \mapsto c), (a \mapsto \{b, c\}), (a \mapsto \{b, c\} \mapsto d)\}$ である。

エピソード $G = (V, E, lab)$ に対して、次のように G **の有限オートマトン** $A(G) = (Pre(G), \Sigma, \delta, H_0, H_f)$ を構築する [5]。

*3 菱形エピソード $a \mapsto E \mapsto b$ は、定義より本体 E が集合であり、同じ要素を重複して含まないため、厳密エピソードとなる。

Algorithm 1 CalcProb($A = (Pre, \Sigma, \delta, H_o, H_f), \mathbf{q} = (q_e)_{e\in\Sigma}, n$)

Input: 有限オートマトン A,
イベント列の生起確率のベクトル $\mathbf{q}$,
非負整数 n.
Output: $p_k (k = 1, \ldots, n)$ の列.

```
p(H_0, 0) := 1;
p(H_0, 0) 以外の p(H, i) := 0 (i := 0 → n)
for all i := 1 → n do
  for all H ∈ Pre do
    p(H, i) := 1;
    for all d ∈ δ such that d(f, e) = H do
      p(H, i) := p(H, i) − q_e;
    end for
    p(H, i) := p(H, i) × p(H, i − 1);
    for all d ∈ δ such that d(H, e) = f do
      p(H, i) := p(H, i) + q_e × p(f, i − 1);
    end for
  end for
end for
output ⟨p(H_f, 1), ..., p(H_f, n)⟩;
```

1. 状態集合は、すべての接頭辞部分グラフのなす集合 $Pre(G)$ である。初期状態は空な頂点集合の誘導するグラフ $H_0 = (\emptyset, \emptyset, \emptyset)$ とし、受理状態は $H_f = G$ である。
2. 状態 $H_1, H_2 \in Pre(G)$ に対し、H_2 から頂点を 1 つ取り除くことで H_1 が導かれる場合、遷移関係 $\delta \subseteq Pre(G) \times \Sigma \times Pre(G)$ は、取り除いた頂点のラベル $e \in \Sigma$ をもつ辺 (H_1, e, H_2) をもつ。
3. 各状態 $H \in Pref(G)$ と任意の $e \in \Sigma$ に対して、e が H から出るいかなる有向辺にもラベルとして出現しない場合、ラベル e をもつ自己ループ辺 (H, e, H) を追加する。

以上の手順で構築した有限オートマトン $A(G)$ の領域は、G のサイズの指数関数である。Tatti [5] は、次の補題を示した。

補題 1. 任意のイベント系列 S と、任意のエピソード G、その有限オートマトン $A(G)$ に対して、S がエピソード G をカバーすることと、$A(G)$ が S を受理することは同値である。

補題 2. G が厳密エピソードならば、その有限オートマトン $A(G)$ は決定性有限オートマトンである。

2.5 カバー確率の計算

Tatti [5] は、2.4 節で導入した有限オートマトンを用いて、与えられた G と、各文字の生起確率 $\mathbf{q}$, 正整数 $k \ge 0$ に対して、独立生成モデルにおける長さ k のランダム文字列による G のカバー確率を効率良く計算する方法を与えている。

$A(G) = (Pre(G), \Sigma, \delta, H_0, H_f)$ の構成より、イベント列 S が $A(G)$ に受理されるならば、かつその場合に限り、S は G をカバーする。よって、イベント列の集合 $\mathcal{S}$ において、$A(G)$ に受理される $S \in \mathcal{S}$ の数が G の観測支持度となる。また、長さ k のイベント列が $A(G)$ の受理状態に到達できる確率を求めることで、カバー確率 p_k が計算できる。

イベント列の生起確率のベクトル $\mathbf{q} = (q_e)_{e\in\Sigma} \in [0,1]^{|\Sigma|}$ と有限オートマトン $A(G)$ において、$\mathbf{q}$ から生成された長さ k のランダムイベント列にしたがい、$A(G)$ を初期状態から遷移し

Algorithm 2 MineRankedDmd($\mathcal{S}, \mathbf{q}, \Sigma, \alpha$)

Input: イベント列の集合 $\mathcal{S}$,
イベント列の生起確率のベクトル $\mathbf{q}$,
アルファベット Σ,
最小頻度 $0 < \alpha \leq |\mathcal{S}|$.
Output: 頻出菱形エピソードとそのランク.

```
Σ0 := 頻出するイベントの集合 (Σ0 ⊆ Σ);
for all ( a ∈ Σ0 ) do
    CaclProb(A(a), q, max_{S∈S} |S|) でカバー確率を計算;
    output ⟨a, a のランク⟩;
    for all ( b ∈ Σ0 ) do
        G0 := (a ↦ ∅ ↦ b);
        A0 := G0から構築した有限オートマトン
        C0 := C_S(G0);
        RecMineRankedDmd(G0, C0, A0, S, q, Σ0, α);
    end for
end for
```

Algorithm 3 RecMineRankedDmd($G = (a \mapsto E \mapsto b), C, A, \mathcal{S}, \mathbf{q}, \Sigma, \alpha$)

Output: $a \mapsto E \mapsto b$ ($E \subseteq \Sigma$) という形の頻出菱形エピソードすべての集合.

```
if ( |C| ≥ α ) then
    CaclProb(A, q, max_{S∈C} |S|) でカバー確率を計算;
    output ⟨G, G のランク⟩;
    for all ( e ∈ Σ (e > max(E)) ) do
        G' := a ↦ (E ∪ {e}) ↦ b;
        B := A から初期状態を取り除いた有向グラフ ;
        B' := A から受理状態を取り除いた有向グラフ ;
        A' := B と B' の対応する頂点間に
              ラベル e を持つ辺を加えた有限オートマトン ;
        C' := C_C(G');   //カバーされるイベント系列集合
        RecMineRankedDmd(G', C', A', S, q, Σ, α)
    end for
end if
```

た結果の状態が $H \in Pre(G)$ である確率を $p(H,k)$ とする。このとき、$p(H,k)$ は以下の式で求められる [5]。

$$p(H,k) = \big(1 - \sum_{(H,e,F)\in\delta} (q_e)\big)p(H,k-1) + \sum_{(F,e,H)\in\delta} (q_e)p(F,k-1)$$

カバー確率 p_k は $p(H_f,k)$ となる。この値は、動的計画法を用いて $\mathcal{O}(v^2k)$ 時間で計算できる。ここに、$v = |Pre(G)|$ である。Algorithm 1 に、カバー確率 p_k の列 $(k = 1,\ldots,n)$ を計算するアルゴリズム CalcProb を示す。

3. 基本アルゴリズム

本節では、イベント列の集合からすべての頻出菱形エピソードとそのランクを重複なく出力するアルゴリズム MineRanked-Dmd を示す。$\mathcal{S}$ をイベント列の集合とし、$\alpha > 0$ を最小頻度とする。

3.1 概要

Algorithm 2 にアルゴリズム MineRankedDmd の概要を示す。MineRankedDmd では、はじめに初期エピソードを計算した後、再帰関数 RecMineRankedDmd を呼び出す。Algorithm 3 に、アルゴリズム RecMineRankedDmd の概要を示す。RecMineRankedDmd は、初期エピソードから開始して、自身を再帰的に呼び出しながら、深さ優先探索により、初期解を拡大して得られるすべての菱形エピソードを重複なく生成する。再帰の各繰り返しで、RecMineRankedDmd は受け取った本体のサイズが m の菱形エピソード（親エピソード）を拡張して、サイズ $m+1$ の菱形エピソード（子エピソード）を生成する。その後、観測支持度を求め、対応する有限オートマトン $A(G)$ を構築する。これを用いてランクを計算する。

3.2 初期エピソードの計算

アルゴリズム MineRankedDmd は、はじめに最もサイズの小さい解として、すべてのイベントの組 $(a,b) \in \Sigma$ に対して、最小の菱形エピソード $G_{ab} = (a \mapsto \emptyset \mapsto b)$ を生成する。さらに、G_{ab} と、G_{ab} をカバーするイベント列の集合 $C = \mathbf{C}(G_{ab})$、および G_{ab} から構築した有限オートマトン $A(G_{ab})$ を引数とし、再帰関数 RecMineRankedDmd を呼び出す。

3.3 エピソードの更新

このステップでは、Katoh ら [3] が提案した菱形エピソードの列挙手法を用いて、再帰的に菱形エピソードを生成する。

基本的なアイディアとして、親エピソード G に新しいイベントを追加して、サイズが一つ大きな子エピソードを生成する（エピソードの拡張）。しかし、このときに単に本体に新しいイベントを追加するだけでは、同じエピソードを異なる経路で複数回生成してしまうことになる。

RecMineRankedDmd では、指数サイズの中間メモリを用いることなく、すべて菱形エピソードを深さ優先探索する。親である菱形エピソード $G = (a \mapsto E \mapsto b)$ が与えられとき、$e > max(E)$ となる任意のイベント $e \in \Sigma$ に対して、本体 E に e を加えて得られる菱形エピソード $G' = (a \mapsto (E \cup \{e\}) \mapsto b)$ を子エピソードとして生成する。

3.4 有限オートマトンの更新

上記のエピソードの更新に伴う、決定性有限オートマトン $A(G)$ の更新は以下の手順で行う。

1. $A(G)$ のコピーを作成し、$A'(G)$ とする。
2. $A(G)$ の受理状態と $A'(G)$ の初期状態を削除する。
3. $A(G)$ 中の初期状態以外の任意の頂点を H とし、H に対応する $A'(G)$ 中の頂点を H' としたとき、ラベル e をもつ辺 (H,H') を追加する。

生成した 1 つの有限オートマトンを更新後の $A(G)$ とする。再帰が 1 つ深くなるごとに、構築する有限オートマトン $A(G)$ のサイズは倍になる。すなわち、エピソード本体 E のサイズが m ならば、$A(G)$ は $2^m + 2$ の状態をもつ。

3.5 カバー確率とランクの計算

上記で更新した決定性オートマトン $A(G)$ を用いて、補題 1 より、G をカバーするイベント列の集合 $\mathbf{C}(G)$ を計算できる。さらに、2.5 節に説明した方法を用いて、G のカバー確率を計算し、G のランクを求める。

以上の手続きを再帰的に繰り返すことで、任意の最小頻度 α に対して、アルゴリズム MineRankedDmd は、イベント列の集合 $\mathcal{S}$ から頻出菱形エピソードとそのランクを重複なく出

力する。生成する各オートマトンのサイズは対応する菱形エピソード $a \mapsto E \mapsto b$ の本体サイズ $|E|$ に指数的であるため、解 1 つあたりの計算時間は $|\Sigma|$ に対して指数的になる。

4. 実験

本節では、3. 節で提案したアルゴリズム MineRankedDmd の評価実験の結果を示す。

4.1 データ

データとして、次のようにランダム生成したイベント列集合を用いた。イベント列集合のサイズ $|\mathcal{S}|$ を 200 から 2000 まで 200 ずつ変化させた 10 種類のデータセットを生成した。アルファベットはサイズ 20 の $\Sigma = \{a, b, \ldots t\}$ であり、イベント列のサイズは $|S_i| = 20$（$i = 1, \ldots, |\mathcal{S}|$）である。さらに、生成したイベント列の約 5% の割合で、菱形エピソード $G_1 = a \mapsto \{b, c, d\} \mapsto e$ を、長さ 5 の部分イベント列として意図的に埋め込んだ。菱形エピソード G_1 が埋め込まれた部分以外のイベントは、Σ から一様ランダムに生成した。

4.2 方法

3. 節で提案したアルゴリズム MineRankedDmd を C++言語で実装し、各データセットに適用した。最小頻度は $\alpha = 100$ とした。アルゴリズムが仮定するイベント列の生成モデルは、入力データセット上の各イベントの出現割合を用いた。

実行環境として、PC(Intel Core i5 2.9GHz, 8GB memory, MacOSX 10.11.6) と g++コンパイラ (Apple LLVM ver8.0.0) を用いた。実験は 5 回繰り返して行い、計算時間を time コマンドで計測した。

4.3 結果

図 1 に、イベント列集合のサイズにおけるアルゴリズム MineRankedDmd の実行時間の平均と、出力解の個数を示す。図 1 より、MineRankedDmd は、解の個数の増加にしたがって、計算時間が増加することがわかる。

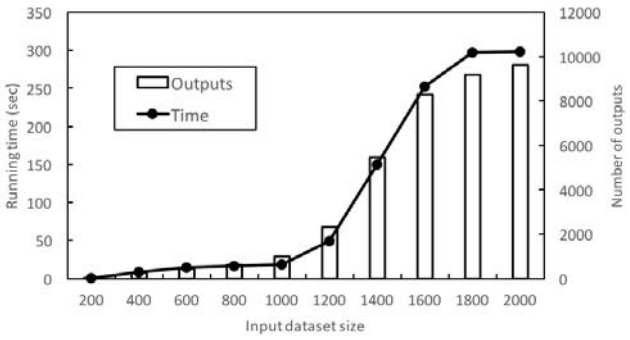

図 1: MineRankedDmd の実行時間と解の個数

$|\mathcal{S}| = 1000$ のデータセットにおいて、解の数は 1011 だった。表 1 に、$|\mathcal{S}| = 1000$ のデータセットにおける有意度ランク上位 5 つのエピソードと、そのランク及び観測支持度を示す。表 1 より、意図的に埋め込んだエピソード G_1 が高いランクを示したことがわかる。表 2 に、$|\mathcal{S}| = 1000$ のデータセットにおける観測支持度上位 5 つのエピソードと、そのランク及び観測支持度を示す。表 2 より、サイズの小さいエピソードが高い観測支持度を示したことがわかる。

表 1: 有意度ランク上位 5 つの解のランクと観測支持度

G	$r(G \mid \mathcal{S} \sim M)$	順位	$sup(G \mid \mathcal{S})$	順位
$a \mapsto \{b, c, d\} \mapsto e$	∞	1	228	409
$a \mapsto \{b, d\} \mapsto e$	513.640	2	245	314
$a \mapsto \{b, c\} \mapsto e$	490.149	3	246	305
$a \mapsto \{c, d\} \mapsto e$	443.951	4	241	338
$a \mapsto \{c, d, e\} \mapsto b$	190.435	5	104	926

表 2: 観測支持度上位 5 つの解のランクと観測支持度

G	$r(G \mid \mathcal{S} \sim M)$	順位	$sup(G \mid \mathcal{S})$	順位
c	1.8377	558	737	1
a	2.9334	492	722	2
d	2.3886	526	721	3
b	4.7831	383	719	4
e	1.5061	638	702	5

5. おわりに

本稿では、Tatti [5] の厳密エピソードのランク計算手法を用いて、イベント列からすべての頻出菱形エピソードを発見しながら、同時にそのランクを計算するアルゴリズム MineRankedDmd を提案した。

今後の課題として、カバー確率の計算を、親エピソードでの計算結果から効率的に計算する手法の開発が挙げられる。アルゴリズム MineRankedDmd では、エピソードの拡張と有限オートマトンの更新については、親エピソードでの計算結果を用いて効率的に行う一方で、カバー確率の計算ははじめから再計算している。この計算の効率化は重要な課題である。加えて、MineRankedDmd を一般のエピソードに拡張することと、実データに対してアルゴリズムを適用し、性能を評価することも今後の課題である。

参考文献

[1] Rakesh Agrawal, Ramakrishnan Srikant. “Mining sequential patterns.” Proc. 11th ICDE, 3-14, 1995.

[2] Takashi Katoh, Kouichi Hirata. “A Simple Characterization on Serially Constructible Episodes.” Proc. PAKDD 2008, 600-607, 2011.

[3] Takashi Katoh, Hiroki Arimura, Kouichi Hirata. “A Polynomial-Delay Polynomial-Space Algorithm for Extracting Frequent Diamond Episodes from Event Sequences.” PAKDD 2009: 172-183

[4] Heikki Mannila, Hannu Toivonen, A. Inkeri Verkamo. “Discovery of Frequent Episodes in Event Sequences.” Data Min. Knowl. Discov. 1(3): 259-289 (1997)

[5] Nikolaj Tatti. “Ranking episodes using a partition model.” Data Min. Knowl. Discov. 29(5): 1312-1342 (2015)