6M-01

確率的イベントストリームにおける最小記述長に基づく代表系列パターンの検出

中村 航規 † 杉浦 健人 †† 石川 佳治 †† 陸 可鏡 ††

† 名古屋大学情報学部コンピュータ科学科 †† 名古屋大学大学院情報学研究科

1 はじめに

イベントストリームからのパターンマイニングの重要性が増してきている．もともと系列データベースからの頻出系列パターンマイニングは 2000 年代ごろから活発に行われてきた．それに加えて，近年ではデータ生成の速度が上がったことで，ストリームからの代表パターンの発見に注目が集まっている [1].

ストリームの 1 種として，各イベントの生起を確率的に表す確率的イベントストリームがある．近年，機械学習の発展により，各イベントに信頼度が付与される不確実データが増大している．不確実データのストリームは確率的イベントストリームとして表現でき，分類結果などのみを利用する場合では欠落しうる情報を漏れなく保持できる．しかし，この確率的イベントストリームから代表パターンを発見する方法はいまだ存在しない．

そこで，本研究では確率的イベントストリームからの最小記述長に基づくパターン発見を提案する．本論文では確率的イベントストリームにおける代表パターンの定義および近似的な最小記述長の計算方法について述べる．

2 関連研究

本章では，最小記述長に基づく代表系列パターン検出を行っている関連研究として SWIFT [1] を紹介する．SWIFT では確率的イベントストリームではなく，各時刻でただ 1 つのイベントが入力されるイベントストリームに対しての代表系列パターン検出を行っている．SWIFT における代表系列パターンの定義を定義 1 に示す．

定義 1. 系列 $S = \langle e_1, e_2, \ldots, e_w \rangle$ が与えられたとき, $\mathrm{L}(S|\mathbb{P})+|\mathbb{P}|$ が最小となるパターン集合 $\mathbb{P} = \{P_1, P_2, \ldots, P_k\}$ を代表系列パターン集合とする．ただし，$\mathrm{L}(S|\mathbb{P})$ は系列 S をパターン集合 $\mathbb{P}$ でエンコードした際の記述長を表し，$\mathbb{P}$ は次の条件を満たすとする．

1. 各イベント $e \in S$ はいずれかのパターン $P_i \in \mathbb{P}$ のマッチにのみ使用される．
2. 各パターン $P_i \in \mathbb{P}$ は系列 S 内で 2 回以上生起する．

■

なお，元論文ではイベントの隣接性に関する条件も含まれるが，ここでは省略する．条件 (1) はパターンにより系列をエンコードする際に，1 つのイベントが 2 つ以上のパターンによって重複して使用されないことを保証する．条件 (2) は頻出パターンとしての制約であり，系列全体を 1 つのパターンとして扱うことによる記述長の最小化を防ぐ．

例として，次の系列 S におけるパターン集合を用いたエンコードを示す．

$$S = \langle s_1, l_2, a_3, s_4, l_5, a_6, s_7, l_8, s_9, l_{10} \rangle$$

代表系列パターン集合の候補としてパターン $P_1 = \langle s, l, a \rangle$ および $P_2 = \langle s, l \rangle$ からなる集合 $\mathbb{P} = \{P_1, P_2\}$ を考える．このとき，系列 S は $\langle P_1, P_1, P_2, P_2 \rangle$ としてエンコード可能であるため，パターン集合 $\mathbb{P}$ の記述長は以下のようになる．

$$\mathrm{L}(S|\mathbb{P}) + |\mathbb{P}| = 4 + 2 = 6$$

この記述長が最小となるようパターン集合を選択するのが SWIFT の基本的な方針である．

また，SWIFT では新しく入力されたイベントに対して一度しか処理を行わない．つまり，再帰的な処理が起こらない．これにより，処理時間が長くならないようにしている．SWIFT は以上のような方針で代表系列パターン集合を考えることによりパターン検出を行った．実験を行い，既存の方法よりも圧縮率や処理時間の面で優れた結果が得られ，有効性と効率性が実証されている．

3 最小記述長に基づくパターン検出

3.1 確率的イベントストリームにおける最小記述長の定義

本研究では SWIFT で定義された代表系列パターンの集合に則ると共に 2 つの拡張を加える．まず，イベント生起の不確実さに対応するためにパターンを正規表現で記述し，より柔軟なパターンを受け入れる．次に，期待値に基づく確率的な最小記述長を用いる．つまり，確率的イベントストリームにおける $\mathrm{L}(S|\mathbb{P})$ を，生起しうる全系列の記述長の期待値とする．

例として，次の確率的イベントストリーム S における系列長を考える．

$$S = \langle ((a, 0.7), (b, 0.3))_1, ((a, 0.6), (b, 0.4))_2, ((a, 0.3), (b, 0.7))_3 \rangle$$

パターン $P = \langle a+ \rangle$ に対して時刻 3 までの系列長を計算したのが表 1 である．全組合せ 8 通りについてそれぞれ確率とエ

Mining Representative Patterns Based on Minimum Description Length from Probabilistic Event Streams

Koki Nakamura†, Kento Sugiura††, Yoshiharu Ishikawa††, and Kejing Lu††

†Department of Computer Science, School of Informatics, Nagoya University

††Graduate School of Informatics, Nagoya University

表 1　系列長の例

時刻 1	時刻 2	時刻 3	確率	エンコード後の長さ	積
a	a	a	0.126	1	0.126
a	a	b	0.294	2	0.588
a	b	a	0.084	3	0.252
a	b	b	0.196	3	0.588
b	a	a	0.054	2	0.108
b	a	b	0.126	3	0.378
b	b	a	0.036	3	0.108
b	b	b	0.084	3	0.252
				L(S|$\mathbb{P}$)	2.4

ンコード後の長さの積を考え，それを加算することで求められる．

3.2　近似的な最小記述長の計算

最小記述長を求める上で，問題となるのは計算量である．単純には，起こりうる全ての系列の確率と記述長を求めれば記述長の期待値も計算できる．しかし，系列を構成するイベントの組合せは指数関数的に増えていくため，全てを列挙するのは現実的ではない．そこで，既存手法 [2] でも用いられている，決定性有限オートマトンの各状態に到達した系列の確率を集約する考え方を利用する．

本研究ではパターン集合によりエンコードされた確率的イベントストリームの記述長を近似的に計算する．ある系列にパターン集合を与えた際の最小記述長の計算自体が NP 困難であり，確率的イベントストリームにおける厳密な記述長の計算は困難である．そこで，パターン集合 $\mathbb{P} = \{P_1, \ldots, P_n\}$ により圧縮された系列の長さを以下の手順で近似的に計算することで，エンコード後の記述長を求める．

1. 各パターン $P_i \in \mathbb{P}$ の選言を取った正規表現 $(P_1|\ldots|P_n)$ で決定性有限オートマトンを生成する．
2. 各時刻，各状態におけるマッチ長とその確率を計算する．ただし，破棄された系列は初期状態へ遷移させる．
3. 受理状態から非受理状態への遷移時に圧縮長を加算する．
4. 処理終了後に受理状態に残っている圧縮長を加算し，系列の長さとの差を取ることで記述長を求める．

例として，系列長を考えた確率的イベントストリーム S について考える．パターン $P = \langle a+ \rangle$ について圧縮長を考える．まず (1) で $a+$ に対応する決定性有限オートマトンを生成する．生成したものが図 1 である．次に (2), (3) でマッチ長とその確率を各状態，各時刻で保持しながら，遷移をさせていき破棄されたときに圧縮長を加算する．加算していく様子を示したものが表 2 である．そして (4) で処理終了後に受理状態に残っている圧縮長を加算する．最後に，元の系列の長さ 3，エンコード後の長さ 1，および上記の処理で得られた圧縮長 1.6 を用いることで記述長 3 + 1 − 1.6 = 2.4 が得られる

オートマトンの遷移は行列として考えられるため，この計算を行列計算で行う．内部状態として，各マッチ長における各状態への到達確率と各状態が保持する候補圧縮長を考える．これに対して，確率的イベントに応じた遷移行列を生成し掛け合わせていくことで計算を進めていく．そして，各パターンについてこれを考えることで最小記述長となるパターン集合を求め，代表系列パターンを検出する．

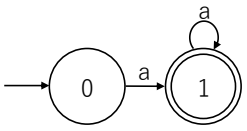

図 1　生成したオートマトン

表 2　圧縮長の計算の様子

時刻	マッチ長	状態 0	状態 1	圧縮長
Init	0	1.0	0	0
1	0	0.3	0	0
	1	0	0.7	
2	0	0.4	0	0.28
	1	0	0.18	
	2	0	0.42	
3	0	0.7	0	0.994
	1	0	0.12	
	2	0	0.054	
	3	0	0.126	
Fin				1.6

4　おわりに

本稿では，確率的イベントストリームにおける最小記述長に基づいたパターン検出の方法について主に述べた．今後は実装および実験を進め，その有用性を検証する．今回は静的なウインドウに対して考えたが，今後はストリーム処理に合わせたアルゴリズムを検討する．また，計算対象となるパターンの増大による時間・空間計算量の増加についても対応方法を検討していく．

謝辞

本研究の一部は JSPS 科研費 JP20K19804，JP21H03555，JP22H03594 の助成の結果得られたものである．

参考文献

[1] Y. Yan, L. Cao, S. Madden, and E. A. Rundensteiner, “SWIFT: Mining representative patterns from large event streams,” *PVLDB*, vol. 12, no. 3, pp. 265–277, 2018.

[2] K. Sugiura and Y. Ishikawa, “Multiple regular expression pattern monitoring over probabilistic event streams,” *Special Section on Data Engineering and Information Management*, vol. E103.D, no. 5, pp. 982–991, 2020.