

確率的なイベントストリームに対するイベントパターン問合せ

加藤 翔[†] 石川 佳治^{†,††}

[†] 名古屋大学大学院情報科学研究科

^{††} 国立情報学研究所

E-mail: kato@db.itc.nagoya-u.ac.jp, y-ishikawa@nagoya-u.jp

あらまし ストリーム形式で大量に発生するイベントの中から、パターン問合せなどの技術を用いてより高次のイベントを検出しようとする複合イベント処理 (CEP) に注目が集まっている。本研究では特に、各イベントに生起確率が付与された確率的イベントストリームに対する CEP に着目する。本研究は、与えられた正規表現のパターンに対し、ひとまとまりの意味をなす問合せ結果の集合を得るための、二つのパターン問合せのセマンティクスを提案する。

キーワード 確率的イベントストリーム、複合イベント処理 (CEP)

1. はじめに

今日では、センサ機器が広く普及しており、様々な環境下においてセンサから得られたデータを取得できるため、これらのセンサデータの有効活用が重要となってきた。たとえば、ユビキタスコンピューティングや医療情報処理の分野において、センシングに基づいて人々の行動状況をモニタリングしようという試みが盛んに進められている [1]~[3]。しかし、センサデータにはノイズが含まれることがあり、そのようなセンサデータから人々の行動を適切に推定することは容易ではない。また、センサデータにノイズが含まれていなくても、似た動きであるが異なる行動を区別することは困難なことがある。これらは、センサデータから人々の行動を認識した結果には曖昧性が存在することを意味する。行動認識の結果は、たとえば「時刻 1 から 10 まで徒歩、11 から 25 まで階段を上る、25 から 40 まで停止」といったように、時間と検出された行動のセットで一般には与えられるが、認識結果の曖昧性が適切には表現できていない。そこで、このような行動認識結果を**確率的イベントストリーム** (probabilistic event stream) として表現することが考えられる。たとえば、行動認識プログラムが図 1 のように各時点ごとに確率が付与された行動データを出力することが考えられる。より多くの情報が含まれていることから、このようなデータを用いた次のステップにおいてより質の高い判断を行うことが可能となりうる。

時刻 1: walk 70%, sit 20%, car 10%
 時刻 2: walk 80%, sit 20%
 時刻 3: sit 70%, stand 30%
 時刻 4: stand 80%, bicycle 20%
 ⋮

図 1 行動認識の例

他の例として、RFID タグを用いたユーザ追跡の例を挙げる。たとえばワシントン大の RFID Ecosystem プロジェクト [4] では、ユーザに RFID タグを付与して行動モニタリングを行っている。部屋や廊下などに RFID タグリーダを設置しておき、ユーザの位置を推定しデータベース化する。その出力の一部に

は、「時刻 1 において部屋 A に 60 %、廊下 B に 40% の確率で存在」といった形の存在確率の情報が含まれる [5], [6]。このようなデータも、先の行動認識の例と同様、確率的イベントストリームとして表現することができる。

本研究では、確率的イベントストリームの中から、ある特定のイベントパターンを問い合わせる問題を考える。ストリーム形式のデータに対するパターン問合せは、データストリームに関する研究や、近年話題となっている**複合イベント処理** (complex event processing, CEP) [7] における基盤技術となっている [8]。たとえば、図 1 に示したデータにおいて、「歩いていた人が、しばらく腰かけた後で立ち上がった」というイベントパターンを問い合わせたいとする。このような問合せは **walk sit⁺ stand** という正規表現で記述できるが、この問合せ結果は

時刻 1~4 において walk, sit, sit, stand と行動した

確率: $70\% \times 20\% \times 70\% \times 80\% = 7.84\%$

時刻 2~4 において walk, sit, stand と行動した

確率: $80\% \times 70\% \times 80\% = 44.8\%$

となり、重複した時間帯において複数のマッチが発生する。確率的イベントストリームに対するイベントパターン問合せに関する既存研究 [5], [6] では、マッチしたものを確率の高い順に報告するなどのアプローチがとられる。しかし、これらの複数のマッチは、いずれもこの時間帯にパターンに合致した行動が発生したことを表しており、個々を区別することは必ずしも重要ではない。むしろ、これらの問合せ結果を総合的に判断して、この時間帯に該当する行動があったと判断する方が妥当であると考えられる。

このような考え方から、本稿では与えられた問合せパターンに対する複数のマッチを統合し、ひとまとまりの問合せ結果とする手法を提案する。得られた問合せ結果は、それ自体ユーザに提示されることもあれば、次の段階としてデータマイニング等のさらに高次の処理に渡されることもありうる。後者の場合は、イベントパターン問合せは、必要な情報を抽出する一種のフィルタリングシステムとして働くことになる。本稿では複数

のマッチングのセマンティクスを提案し、それらの処理方式について、特にデータストリーム処理の観点から議論する。

2. 確率的イベントストリーム

本研究で対象とする確率的イベントストリームを以下のように定義する。

[定義 1] **確率的イベントストリーム** (probabilistic event sequence) S は、

$$S = e_1, e_2, \dots, e_t, \dots \quad (1)$$

と与えられる無限の系列である。

$$e_t = \{e_{t1}, e_{t2}, \dots, e_{t|V|}\} \quad (2)$$

は、時刻 t における**イベント集合** (event set) であり、各 $e_{ti} \in e_t$ を**イベント** (event) と呼ぶ。 V はイベント値のドメインであり、離散的であるとする。各イベント e_{ti} には生起確率 $\Pr(e_{ti})$ が付与されているとし、

$$\sum_{i=1}^{|V|} \Pr(e_{ti}) = 1 \quad (3)$$

が成り立つとする。 $\square$

以下の例では、表記を簡略化するため、 V の要素を $a, b, \dots$ というアルファベットで表す。また、イベント集合を表す際には、生起確率が 0 である値は省略して、 $e_t = \{a, c, d\}$ のように示すことにする。

先の定義にあるように、本研究では、以下のような想定を行っている。

- (1) イベントの値は離散的なドメイン V からとられる。
- (2) 単位時間ごとにイベント集合が得られる：イベントの欠落は考えない。また、発生したイベントが発生順には到達しない、いわゆるアウトオブオーダー (out of order) 型のイベントストリームは考えない。なお、ここでは「単位時間」という用語を用いたが、実際には一定時間きざみである必要はなく、イベント集合間の前後関係が明確に決まればよい。
- (3) ある時点におけるイベント値の発生確率を足すと 1 になる：状況によっては、未知の値のイベントを考慮したいこともある。その場合には、 $V' = V \cup \{\perp\}$ という、未知の値 $\perp$ で拡張したドメイン V' を考えればよいので、本質的な問題ではない。
- (4) 時刻 t におけるイベントの生起確率 $\Pr(e_{ti})$ ($i = 1, \dots, |V|$) は、他の時刻におけるイベントの生起確率とは独立している：すなわち、Markovian Streams [5], [6] に見られるような、イベントの生起が時間的な相関を持つような状況は考えていない。

以上のとおり、確率が付与されており、同じ時刻にいくつかの値が同時に発生しうることを除けば、単純なイベントストリームを想定している。

3. 確率的イベントストリームに対するパターンマッチング

3.1 パターンマッチングの例

単純な正規表現のパターン $p = ab^+c$ が問合せとして与えら

t	a	b	c	d
1	0.9	0.1		
2	0.2	0.8		
3		1.0		
4		0.4	0.6	
5			0.4	0.6

図 2 確率的イベントストリームの例 (その 1)

	a	b	c	d
1	0.9	0.1		
2	0.2	0.8		
3		1.0		
4		0.4	0.6	
5			0.4	0.6

図 3 単純照合の例

れたとする。これは、イベント a が発生した後、 b というイベントが 1 回以上発生し、 c というイベントが発生したというパターンを表現している。ここで図 2 に示されるような確率的イベントストリームが与えられたとする。時刻 $t = 1$ から $t = 5$ までの各時刻にイベントが発生し、それぞれに対して表に示すような生起確率が付与されているとする。

単純にパターン問合せを行うと、次のような四つの問合せ結果が得られる^(注1)。図 3 にこれらを図示する。個々の問合せ結果のことを、本研究では**マッチ** (match) と呼ぶ。

$$\begin{aligned} m_1 &= \langle (1, a), (2, b), (3, b), (4, c) \rangle & (0.432) \\ m_2 &= \langle (1, a), (2, b), (3, b), (4, b), (5, c) \rangle & (0.1152) \\ m_3 &= \langle (2, a), (3, b), (4, c) \rangle & (0.12) \\ m_4 &= \langle (2, a), (3, b), (4, b), (5, c) \rangle & (0.032) \end{aligned}$$

たとえば m_1 は、 $t = 1, 2, 3, 4$ においてそれぞれ a, b, b, c にマッチしたことを意味している。マッチ m に対応する時区間 (time segment) を $ts(m)$ で表す。たとえば $ts(m_1) = [1, 4]$ となる。また、マッチ m の開始時刻、終了時刻を、それぞれ $start_ts(m), end_ts(m)$ で表す。マッチ m の時刻 t におけるエントリを $m[t]$ で表し、時刻 t におけるイベントおよび確率を $m[t].ev, m[t].pr$ で表す。

マッチに対する確率を計算することができる。そのマッチがどの程度の確率で合致したかを表すことから、マッチの**一致確率** (match probability) と呼ぶ。上に示したマッチ結果の右端の括弧内の数値がこれを表しており、たとえば $\Pr(m_1) = 0.9 \times 0.8 \times 1.0 \times 0.6 = 0.432$ と計算される。以上より、パターン $p = ab^+c$ に対する問合せ結果として、 $\{m_1 : 0.432, m_2 : 0.1152, m_3 : 0.12, m_4 : 0.032\}$ という四つのマッチからなる集合を返すというのが一つの考え方である。このような問合せ結果を、本研究では**単純照合** (simple match) と呼ぶことにする。

(注1)：イベントストリームに対するパターン問合せでは、たとえば $a, f, b, \dots$ というイベント系列に ab というパターンのマッチを許したい場合がある。すなわち、パターンに出現しない文字 (ここでは f) をスキップするというものである。本稿では簡単のため、このようなスキップを許さない処理を想定する。このような問合せ戦略は [8] では *strict contiguity* と呼ばれている。

[定義 2] 確率的イベントストリーム S にパターン p が与えられたときの単純照合に基づくマッチの集合を、 $\text{simple_match}(S, p)$ で表す。□

3.2 単純照合の問題点

単純照合の考え方は分かりやすいものであり、ユーザがそのような問合せ結果を望んでいる場合も十分考えられるので、基本となるパターン問合せのセマンティクスである。しかし、たとえば行動追跡において、ユーザがひとまとまりの行動を取得するためにパターン問合せを行う状況を考えて、単純照合のアプローチは必ずしも適切ではない。図 2 において、イベント a, b, c を行動とみなすと、時刻 $t = 1$ 付近では a 、 $t = 3$ 付近では b 、 $t = 5$ 付近では c が発生していることが分かる。すなわち、図に示されている確率的イベントストリームは実際にはひとまとまりの行動を表しているが、単純照合ではこれをまとまったものとして抽出できないことがわかる。

そこで本稿では、確率的イベントストリームに対する複数のパターン問合せのセマンティクスを示し、それらの処理方式について述べる。ユーザは、処理の要求に応じて適切なセマンティクスを選択することになる。

4. パターン問合せのセマンティクス

以下で定義するパターン問合せのセマンティクスは、いずれもひとまとまりのイベント系列をどのように取り出すかという考え方に基づいている。どのようにまとまりを構成するかで異なるセマンティクスが得られる。

4.1 完全オーバーラップ

完全オーバーラップを次のように定義する。

[定義 3] $M \subseteq \text{simple_match}(S, p)$ を単純照合によるマッチの部分集合とする。 M が**完全オーバーラップ** (complete overlap) の性質を有するとは、 M が

$$\forall m, m' \in M \text{ such that } m \neq m', \text{ts_overlap}(m, m') \quad (4)$$

を満たす極大な集合 (任意の $\text{simple_match}(S, p)$ の要素を追加した場合に上の式が満たされなくなる) である場合をいう。ここで $\text{ts_overlap}(m, m')$ は m と m' の時区間 ($\text{ts}(m)$ と $\text{ts}(m')$) に交わりがあるときに真になる述語である。 M に対応する時区間を、 M に含まれるマッチについて最大の時区間をとることで

$$\text{start_ts}(M) = \min\{\text{start_ts}(m) \mid m \in M\} \quad (5)$$

$$\text{end_ts}(M) = \max\{\text{end_ts}(m) \mid m \in M\} \quad (6)$$

により、

$$\text{ts}(M) = [\text{start_ts}(M), \text{end_ts}(M)] \quad (7)$$

と定義する。□

完全オーバーラップのセマンティクスを用いると、確率的イベントストリームから、それぞれ完全オーバーラップの性質を持つような 0 個以上複数個のマッチの集合が得られる。たとえば、先の例に示した確率的イベントストリームについては、完全オーバーラップであるマッチの集合 $\{m_1, m_2, m_3, m_4\}$ がただ一つ得られる。その対応する時区間は $[1, 5]$ である。本研究では、このようにして提案の照合セマンティクスで得られるマッチの集合を **グループ** (group) と呼ぶことにする。

4.2 グループの確率

ここで問題となるのが、完全オーバーラップに基づいて得られるグループ M についてどのように確率を付与するかである。問合せ結果として意味をなすように確率を与えたい。

ここで図 3 に示したマッチの例を見直してみると、単純照合の経路にオーバーラップがあることがわかる。たとえば m_1 と m_2 には、時刻 $t = 2, 3$ においてはいずれも b が対応している。行動認識のコンテキストで説明すると、複数のマッチが同じ時刻に同じ行動を共有していることになる。そのため、単純照合におけるマッチの一致確率を単純に足し合わせるのは合理的でない。

図を見ると、モニタリングの対象者は、時刻 $t = 1$ では a という行動をとり (確率 0.9)、 $t = 2$ では a または b という行動をとり (両者の確率を足すと 1)、 $t = 3$ では b という行動をとり (確率 1)、 $t = 4$ では b または c という行動をとり (両者の確率を足すと 1)、 $t = 5$ では c という行動をとっている (確率 0.4)。そこで本研究では、この場合のグループ $M = \{m_1, m_2, m_3, m_4\}$ に対する確率を $0.9 \times (0.2 + 0.8) \times 1.0 \times (0.4 + 0.6) \times 0.4 = 0.36$ と与える。この確率は、直観的には、与えられたパターン $p = ab^+c$ がこの確率的イベントストリームにおいてどの程度成立しているかを表しており、パターンに対するグループの一致確率と考えることができる。確率が 1 となるのは、その確率的イベントストリームが与えられたパターンにすべて合致した場合で、逆に 0 となるのはまったく合致しなかった場合である。この点で、確率的にも意味をなす。

以上の議論をもとに一致確率を以下のように定義する。

[定義 4] グループ M の**一致確率** (match probability) を

$$\Pr(M) = \prod_{t \in \text{ts}(M)} \sum_{e \in \{m[t].\text{ev} \mid m \in M\}} e[t].\text{pr} \quad (8)$$

と定義する。□

ただし、 $e[t].\text{pr}$ は、時刻 t におけるイベント e の確率の値を表すとする。たとえば、図 2 においては $a[1].\text{pr} = 0.9$ である。

4.3 部分オーバーラップ

完全オーバーラップのセマンティクスにより、与えられたパターンに対し関連するひとまとまりの問合せ結果をグループという形でまとめることが可能となる。しかし、完全オーバーラップのグループ M において任意の二つの要素が必ずオーバーラップしなければならないという制約は、状況によっては強すぎることもある。例として、図 4 について、先と同様に ab^+c というパターン問合せを考える。単純照合の結果は

$$m_1 = \langle (1, a), (2, b), (3, c) \rangle$$

$$m_2 = \langle (1, a), (2, b), (3, b), (4, c) \rangle$$

$$m_3 = \langle (2, a), (3, b), (4, c) \rangle$$

$$m_4 = \langle (4, a), (5, b), (6, c) \rangle$$

となる。完全オーバーラップのセマンティクスに基づけば、 $M = \{m_1, m_2, m_3\}$ および $M' = \{m_2, m_3, m_4\}$ が得られる。 m_1 と m_4 の時区間がオーバーラップしていないため別々のグループとなる。完全オーバーラップは、相互に関連が強いという点で一つの意味があるが、 M, M' のように重複する要素を含む複数の結果を返してしまうことがある。

t	a	b	c	d
1	0.9	0.1		
2	0.2	0.8		
3		0.7	0.3	
4	0.8		0.2	
5	0.1	0.8		0.1
6		0.1	0.7	0.2

図 4 確率的イベントストリームの例（その 2）

そこで、制約を大幅に緩めた部分オーバーラップを導入する。
 [定義 5] $M \subseteq \text{simple_match}(S, p)$ を単純照合によるマッチの部分集合とする。 M が**部分オーバーラップ**（partial overlap）の性質を有するとは、 M が

$$\forall m \in M, \exists m' \in M \text{ such that } m \neq m', \text{ts_overlap}(m, m') (9)$$

を満たす極大な集合である場合をいう。 M に対応する時区間を完全オーバーラップの場合と同様に定義する。 □

部分オーバーラップのセマンティクスでは、先の例の場合、 $M = \{m_1, m_2, m_3, m_4\}$ という一つのグループが得られる。部分オーバーラップでは、問合せ結果のグループの個数が完全オーバーラップに比べ同等以下になるので、問合せ結果を絞り込んで提示することができるという利点もある。

完全オーバーラップと部分オーバーラップのどちらのセマンティクスを採用するかは、対象とするアプリケーションやユーザの意向に依存する。場合によっては、同じ確率的イベントストリームに二つのセマンティクスを同時に適用し、その問合せ結果を比較することも考えられる。

5. 確率の閾値の導入

確率的イベントストリームを対象としたパターン問合せでは、生起確率が十分大きいもののみを検出したいという要求が大きいと考えられる。そのため、確率に対する閾値を導入する。第一に、グループに対する閾値を設定することが考えられる。この閾値を**問合せ閾値**（query threshold）と呼ぶ。たとえば、4.1 に示した完全オーバーラップの例において、閾値を 20% と設定することができる。例で述べたとおり、その例ではグループの一致確率は 0.36 であったため、閾値を満たすことから問合せ結果が出力されることになる。

一方、単純照合により得られる個別のマッチに対する閾値を設定することも考えられる。これを**マッチ閾値**（match threshold）と呼ぶ。たとえば、同じく 4.1 の例においてマッチの閾値を 5% とした場合、 m_4 が閾値を満たさないため、完全オーバーラップに基づくグループは $M = \{m_1, m_2, m_3\}$ となる。ただし、 m_4 はどの時刻においても M に含まれる他のマッチとオーバーラップがあるため、 M の時区間や一致確率に変化はない。

マッチに対する閾値は、部分オーバーラップのセマンティクスにおいて必要以上にグループが連結してしまうことを抑制するために使用することができる。図 5 の例を用いて説明する。

この確率的イベントストリームについてパターン ab^+c による問合せの単純照合は、

t	a	b	c	d
1	1.0			
2	0.1	0.9		
3		0.2	0.8	
4	0.8	0.1	0.1	
5		1.0		
6			0.8	0.2

図 5 確率的イベントストリームの例（その 3）

$$\begin{aligned} m_1 &= \langle (1, a), (2, b), (3, c) \rangle & (0.72) \\ m_2 &= \langle (1, a), (2, b), (3, b), (4, c) \rangle & (0.018) \\ m_3 &= \langle (1, a), (2, b), (3, b), (4, b), (5, b), (6, c) \rangle & (0.0144) \\ m_4 &= \langle (2, a), (3, b), (4, c) \rangle & (0.002) \\ m_5 &= \langle (2, a), (3, b), (4, b), (5, b), (6, c) \rangle & (0.0016) \\ m_6 &= \langle (4, a), (5, b), (6, c) \rangle & (0.64) \end{aligned}$$

という六つのマッチとなる。これを見ると、 m_1, m_6 の確率は高いのに対し、その間の m_2, m_3, m_4, m_5 は確率がかなり小さいことがわかる。つまり、この場合は m_1 と m_6 が独立したイベント系列であり、 m_2, m_3, m_4, m_5 はノイズのため検出されたのかもしれないと考えられる。しかし、部分オーバーラップのセマンティクスをそのまま用いた場合、 $M = \{m_1, m_2, m_3, m_4, m_5, m_6\}$ が問合せ結果となってしまふ。

そこでマッチに対する確率の閾値を設定する。たとえばマッチ閾値を 5% とすると m_1, m_6 のみが単純照合の結果となり、部分オーバーラップのセマンティクスの下でも $M = \{m_1\}$ と $M' = \{m_6\}$ という分離されたグループが得られる。このように、マッチに対する閾値はノイズの影響を差し引き、グループが過度に連結されてしまうことを防ぐために利用できる。

6. アルゴリズム

6.1 グループに関するアルゴリズム

アルゴリズムで使用する記号を表 1 にまとめる。グループに関するアルゴリズムは、グループ管理とグループ出力の二つがある。

表 1 アルゴリズム中で使用する記号

記号	意味
r_i	アクティブラン
m_j	受理ラン（マッチ）
$x.t_s$	x が開始した時刻（ x は r もしくは m ）
$x.t_e$	x が受理された時刻（ x は r もしくは m ） ^(注2)
A	アクティブランの集合（ $\forall r_i, r_i \in A$ ）
M	受理ランの集合（ $\forall m_j, m_j \in M$ ）
g_k	グループ
G	グループの集合（ $\forall g_k, g_k \in G$ ）
A_k	g_k に属するアクティブランの集合
M_k	g_k に属する受理ランの集合
X_{temp}	ラン（ r や m ）が一時的に格納される集合
t_l	連鎖が途切れた時刻
T	連鎖が途切れた時刻の集合（ $\forall t_l, t_l \in T$ ）

(注2)：その時点では受理された時刻が不明である r については、現時刻を返すことにする。

6.1.1 完全オーバーラップにおけるグループの管理

完全オーバーラップにおける、新規ランの生成時およびランの失敗に対するグループ管理のアルゴリズムを述べる。まず、新規ランの生成時におけるグループ管理の流れをアルゴリズム 1 に示す。入力是新規ランである。まず、2 行目にあるように、グループが一つ以上存在するかどうか調べる。存在しない場合は 16 行目に移る。createNewGroup($\{x_1, x_2, \dots, x_n\}$) は、ランの集合 $\{x_1, x_2, \dots, x_n\}$ を要素に持つ新しいグループを生成するものである。つまり、グループが一つも存在しない場合は、新規ランのみを要素に持つグループを新たに生成する (2 行目, 16 行目)。グループが存在する場合は 3 行目～14 行目に移る。各グループに対して、以下の操作を行う。なお、getNewest($\{x_1, x_2, \dots, x_n\}$) は、 $\min\{x_i.t_e | x_i \in \{x_1, x_2, \dots, x_n\}\}$ を返す。

- グループに属する受理ラン集合から最も若い受理ランを抽出し (5 行目)、その受理ランが受理した時刻と新規ランの開始時刻を比較する (6 行目)。もし、新規ランの開始時刻の方が大きい場合は次のグループに移る (7 行目)。等しい場合もしくは受理ラン集合が空 (4 行目) の場合、そのグループのアクティブラン集合に新規ランを加える (10 行目)。新規ランがどのグループにも加えられることがなかった場合 (12 行目)、13 行目に移り、すべてのアクティブランを要素とする新しいグループを生成する。

次に、ランの失敗に対するグループ管理についてであるが、完全オーバーラップにおいては、失敗したランを含むグループからそれを取り除くだけである。

Algorithm 1 新規ラン生成時のアルゴリズム (complete)

```

1: procedure GROUPMAINTENANCEFORNEWRUN( $r_{new}$ )
2:   if  $G \neq \emptyset$  then
3:     for all  $g_i \in G$  do
4:       if  $M_i \neq \emptyset$  then
5:          $m_{temp} \leftarrow \text{getNewest}(M_i)$ 
6:         if  $m_{temp}.t_e < r_{new}.t_e$  then
7:           continue
8:         end if
9:       end if
10:       $A_i \leftarrow A_i \cup \{r_{new}\}$ 
11:    end for
12:    if  $\forall m \in M, m.t_e < r_{new}.t_s$  then
13:      createNewGroup( $A$ )
14:    end if
15:  else
16:    createNewGroup( $\{r_{new}\}$ )
17:  end if
18: end procedure

```

6.1.2 部分オーバーラップにおけるグループの管理

部分オーバーラップにおける、新規ランの生成時およびランの失敗に対するグループ管理のアルゴリズムを述べる。部分オーバーラップのセマンティクスでは、グループの総数は基本的に 0～1 であり、そのような場合にはグループ管理のアルゴリズムは単純である。しかし、ランの失敗によりグループの総数が一時的に二つ以上になる場合が存在する。この場合を考慮するため以降で述べるアルゴリズムは単純でない部分があるが、実際の

Algorithm 2 新規ラン生成時のアルゴリズム (partial)

```

1: procedure GROUPMAINTENANCEFORNEWRUN( $r_{new}$ )
2:   if  $G \neq \emptyset$  then
3:     for all  $g_i \in G$  do
4:        $x_{temp} \leftarrow \text{getNewest}(A_i \cup M_i)$ 
5:       if  $x_{temp}.t_e == r_{new}.t_e$  then
6:          $A_i \leftarrow A_i \cup \{r_{new}\}$ 
7:         break
8:       end if
9:     end for
10:    if  $\forall x \in A \cup M, x.t_e < r_{new}.t_s$  then
11:      createNewGroup( $\{r_{new}\}$ )
12:    end if
13:  else
14:    createNewGroup( $\{r_{new}\}$ )
15:  end if
16: end procedure

```

処理においては単純な処理が行われることがほとんどであると考えられる。

まず、新規ランの生成時におけるグループ管理の流れをアルゴリズム 2 に示す。入力是新規ランである。まず、2 行目にあるように、グループが一つ以上存在するかどうか調べる。存在しない場合は 14 行目に移り、新規ランのみを要素に持つグループを新たに生成する。存在する場合は 3 行目～12 行目に移る。各グループに対して、以下の操作を行う。

- グループに属するラン集合から最も若いランを抽出し (4 行目)、そのランが受理した時刻 (ランがアクティブランの場合には現時刻) と新規ランの開始時刻を比較する (6 行目)。もし等しければ、新規ランをそのグループのアクティブラン集合に加え (6 行目)、処理を終える (7 行目)。新規ランがどのグループにも加えられることがなかった場合 (10 行目)、11 行目に移り、新規ランを要素とする新しいグループを生成する。

次に、ランの失敗によりランが除去されたグループにおけるグループ管理をアルゴリズム 3 に示す。入力、失敗したためグループから除去されるランである。まず、2 行目～4 行目にあるように、各グループに対して、ランが除去されたことでグループに属するランの連鎖が途切れていないか調べる。途切れていない場合、この処理は終了である。途切れていた場合、途切れた時刻の集合を T として 5 行目～16 行目に移る。途切れた各時刻 t_i に対して以下の操作を行う。

- 対象のグループに属する受理ラン集合のうち、途切れた時刻以前の受理ランをグループの受理ラン集合から除去する (6 行目～11 行目)。除去された受理ラン集合を要素に持つ新しいグループを生成する (12 行目)。

最後に、対象のグループに属する受理ラン集合に除去されずに残っている受理ランとすべてのアクティブランを要素に持つ新しいグループを生成し (15 行目)、16 行目で対象のグループを削除する。

6.1.3 グループの出力

グループの出力は、グループに属するアクティブラン集合が空になった場合に、そのグループの時区間がこれまでに出力さ

Algorithm 3 ラン除去におけるアルゴリズム (partial)

```

1: procedure GROUPMAINTENANCEFORREMOVEDRUN( $r_{delete}$ )
2:   for all  $g_k \in G$  do
3:     Remove  $r_{delete}$  from  $A_k$ 
4:     if  $\neg(\forall x \in M_k \cup A_k, \exists x' \in M_k \cup A_k \text{ such that } x \neq x', ts\_overlap(x, x'))$  then
5:       for all  $t_i \in T$  do
6:         for all  $m_j \in M_k$  do
7:           if  $m_j.ts < t_i$  then
8:              $X_{temp} \leftarrow X_{temp} \cup \{m_j\}$ 
9:             Remove  $m_j$  from  $M_k$ 
10:          end if
11:        end for
12:         $createNewGroup(X_{temp})$ 
13:        Clear  $X_{temp}$ 
14:      end for
15:       $createNewGroup(M_k \cup A)$ 
16:      Delete  $g_k$ 
17:    end if
18:  end for
19: end procedure

```

	a	b	c	d
1	0.9	0.1		
2		1.0		
3		0.2	0.8	
4		0.1		0.9
5				1.0

	a	b	c	d
1	0.9	0.1		
2		1.0		
3		0.8	0.2	
4			0.9	0.1
5				1.0

図 6 一致確率では不適切な確率的イベントストリームの例

れたグループの時区間に包含されていなければ実行される。グループの出力では、グループの一致確率、グループの時区間、グループに属するマッチ集合をユーザに提示する。

6.2 確率の閾値に関するアルゴリズム

6.2.1 マッチ閾値による処理の打ち切り

マッチ閾値による処理の打ち切りは、単純に、入力イベントによりランが遷移した際に、その時点でのランの一致確率とマッチ閾値を比較し、一致確率が閾値を下回った場合にそのランを削除することで処理を打ち切るものである。

6.2.2 問合せ閾値による処理の打ち切り

マッチ閾値による閾値の比較対象はその時点でのランの一致確率であるが、問合せ閾値による閾値の比較対象はその時点でのグループの一致確率では不適切な場合がある。たとえば、照合パターン $p = ab^+c$ に対して図 6 の左側のような確率的イベントストリームを考える。ここで、現時点を 4、問合せ閾値を 0.2 とする。現時点におけるグループの一致確率は、 $0.9 \times 1.0 \times (0.2 + 0.8) \times 0.1 = 0.09$ である。この時点では問合せ閾値を下回っている。しかし、次の時刻 5 では、グループに属するアクティブランが失敗することでグループの一致確率は $0.9 \times 1.0 \times 0.8 = 0.72$ に確定し、問合せ閾値を上回る。よって、この時点で処理を打ち切ることは不適切である。

先の例では、アクティブランを除いたグループにおいて現時点の一致確率を問合せ閾値の比較対象とするとうまくいく。しかし、そのような手法も場合によっては不適切である。たとえ

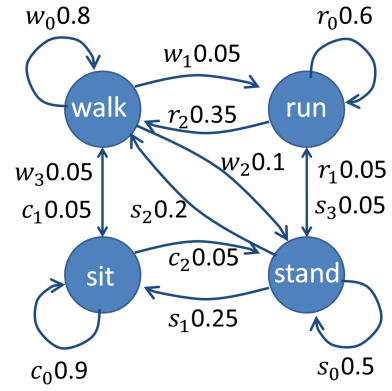


図 7 決定的イベントストリーム生成に用いるモデル

ば、先と同じ照合パターンに対して図 6 の右側のような確率的イベントストリームを考える。現時点を 3、問合せ閾値を 0.2 とする。現時点におけるアクティブランを除いたグループの一致確率は、 $0.9 \times 1.0 \times 0.2 = 0.18$ である。この時点では問合せ閾値を下回るが、次の時刻 4 ではアクティブランが受理に達し、グループの一致確率は $0.9 \times 1.0 \times (0.8 + 0.2) \times 0.9 = 0.81$ に確定し、問合せ閾値を上回る。そこで、受理ランの時区間のみにおいて、グループの一致確率の計算方法を適用して得られる値を **最大一致確率** (maximum match probability) とし、問合せ閾値による閾値の比較対象とする。

問合せ閾値による処理の打ち切りは、最大一致確率が問合せ閾値を下回った場合にそのグループを解消することで行う。

7. 評価実験

7.1 使用した確率的イベントストリーム

今回使用する確率的イベントストリームには人工データを用いた。本実験では、まず初めに曖昧性のない決定的なイベントストリームを生成した。次に、生成した決定的なイベントストリームに擬似的な観測ノイズを与えることで確率的なイベントストリームを生成した。決定的なイベントストリームは図 7 のような、遷移先が各時刻において確率的に決定されるモデルによって生成した。擬似的な観測ノイズとは、正しくない観測値の割合である。本実験では、擬似的な観測ノイズのパラメータとして最小値、最大値を与えた。観測ノイズの値は、各時刻においてその範囲内からランダムに決定される。

本実験ではノイズが与えられる前のイベントストリームを知ることができる。そのため、実験では加工前のイベントストリームの情報も用いて評価を行った。具体的には、確率的イベントストリームの元となった決定的イベントストリームに対して単純照合のセマンティクスで得られるマッチ集合を、検出されるべき **解** (solution) であるとし、解に含まれるマッチを **解マッチ** (solution match) として評価に用いた。なお、解マッチの数は 81 である。実験で用いる人工データのパラメータの基本情報を表 2 に記す。

表 2 人工データのパラメータ

イベントの数	10000
イベントの種類	4
ノイズの最小値, 最大値	[0%, 1%]~[30%, 40%]

8. 実験1：提案手法の機能（閾値設定なし）

まず、提案手法が実際にどのように機能するか調べるため、閾値を設定せずに実験を行った。提案手法は、確率的イベントストリームから意味をなすひとまとまりの結果を出力することが目的である。実験1では、決定的イベントストリームから確率的イベントストリームを生成する際のノイズの範囲を $[0, 1]$, $[1, 2]$, $\dots$, $[3, 4]$ と最大最小共に1%ずつ変化させて、四つの確率的イベントストリームを生成した。本稿の実験では、問合せのイベントパターンとして $\text{walk sit}^+ \text{stand}$ を使用した。生成した四つの確率的イベントストリームに対して、完全オーバーラップと部分オーバーラップのそれぞれでイベントパターン問合せを行った結果を図8に示す。左側の縦軸には折れ線グラフが対応し、右側の縦軸には棒グラフが対応する。groupT

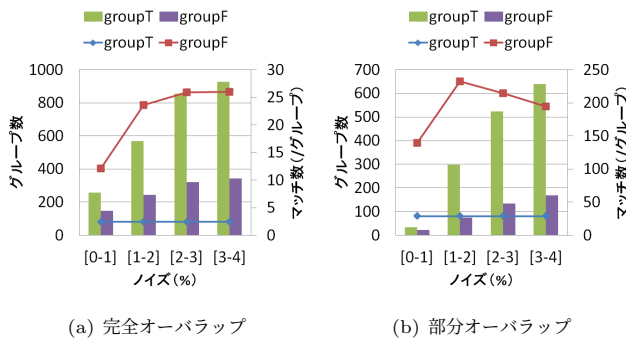


図8 閾値を設定しない場合の提案手法の効果

は、出力されたグループ集合のうち、解マッチとオーバーラップしているグループの集合を表す。解マッチの数は81であるため、groupTのグループ数は81を上限として正しい結果をどれだけ出力したかを表す。groupFは、出力されたグループ集合のうち、解マッチにオーバーラップがないグループの集合を表す。つまり、groupFのグループ数は、間違っただけの結果をどれだけ出力したかを表す。図を見ると、どちらのセマンティクスでもgroupTのグループ数は常に81であり、最高の結果である。一方で、groupFのグループ数についてはどちらのセマンティクスでもgroupTの数倍以上ある。これは、閾値を設定しない問合せでは、問合せ結果における間違いの割合にノイズが多大な影響を及ぼすことを示している。また、間違っただけの結果を出力していることは提案手法が原因なのではなく、単純照合における結果のマッチ集合にそのようなマッチが含まれていることが原因である。

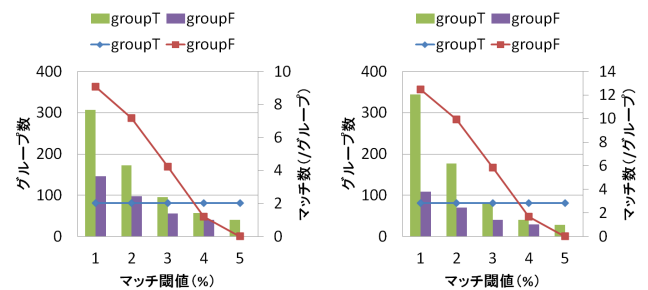
次に、図の棒グラフについて述べる。groupTのマッチ数は、groupTの各グループが平均してどれほどのマッチを含んでいるかを示している。groupFでも同様に、groupFの各グループが平均してどれほどのマッチを含んでいるかを示している。この二つの数値は、ひとつのグループに重複したマッチがどの程度まとめられているかの傾向を見るためのものである。図を見ると、どちらのセマンティクスでもノイズが増えるにつれて、ひとまとめにされるマッチの数が増える傾向にある。これは、単純照合による出力結果においては、ある時区間を対象としたとき、ノイズが増えるにつれてその時区間内で重複するマッチの数がより多くなることを示す。

完全オーバーラップと部分オーバーラップでは大きな差がみられ

る。図のノイズ3~4%のgroupTのマッチ数においては、部分オーバーラップは平均して完全オーバーラップの8倍以上である。これは、ノイズにより部分オーバーラップのグループが過度に連結されたためだと考えられる。部分オーバーラップの過度な連結の制御は次の実験2で述べる。マッチ数に関してgroupTとgroupFの間にも差がみられるが、これは解マッチの近くではより多くの重複したマッチが生成される傾向にあることを示している。

9. 実験2：マッチ閾値の設定

実験1により、提案手法のそれぞれの性質・機能が明らかになった。また、ノイズの存在下では単純照合による出力結果の多くが間違いであることが分かった。そこで実験2では、マッチ閾値の設定により、マッチ集合に含まれる間違いの変化を調べた。ノイズ3~5%の確率的イベントストリームに対するパターン問合せにおいて、マッチ閾値を変化させたときの結果を図9に示す。図のgroupFを見ると、マッチ閾値を増やすに



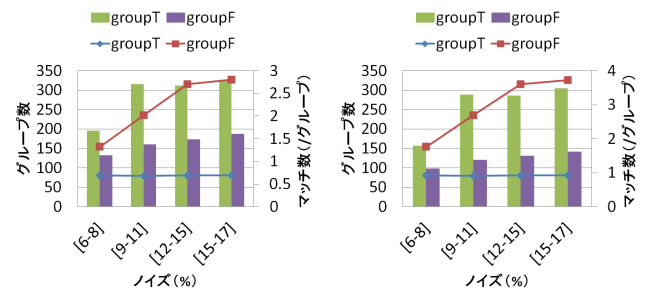
(a) 完全オーバーラップ

(b) 部分オーバーラップ

図9 ノイズ[3,5]固定でマッチ閾値を変化

連れて、間違っただけのグループの数が下がっていき、マッチ閾値が5%の時点で完全に0になっている。一方で、groupTを見ると、解マッチとオーバーラップのある正しいグループの数は下がることなく最高の数値81を保っている。このことから、ノイズの存在下に対してマッチ閾値は有効に働くことが分かった。このときのgroupTのマッチ数を見ると、どちらのセマンティクスでもほぼ1であることが分かる。これは、提案手法のひとまとめにする機能がほとんど作用していないことを示す。すなわち、このような時区間の重複するマッチがほとんどない状況では提案手法は意味を成さない。

次に、マッチ閾値の値を固定しノイズを増加させた場合の結果を図10に示し、実験1と比較することで提案手法に対するマッチ閾値の効果を述べる。実験1と大きく異なる点は、部分



(a) 完全オーバーラップ

(b) 部分オーバーラップ

図10 マッチ閾値0.05固定でノイズを変化

オーバーラップにおける groupT のグループ数が完全オーバーラップのそれとほぼ同じである点である。部分オーバーラップは定義上どこまでも連鎖が続く可能性がある。その一端が現れたのが実験 1 である。今回の結果から、そのような連鎖を望まない場合はマッチ閾値を設定することが有効であることが示された。

実験 2 の最初の実験では間違ったマッチを取り除くにはマッチ閾値が非常に有効であることを示したが、ある程度の大きさのノイズに対しては限界が存在することを以下に示す。ノイズを 15~17% に固定してマッチ閾値を変化させたときの結果を図 11 に示す。図 11 を見ると、図 9 と同様に、マッチ閾値の増加

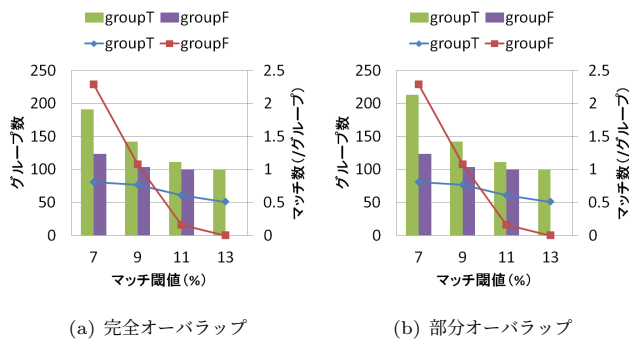


図 11 ノイズ [15-17] 固定でマッチ閾値を変化

に伴って groupF のグループ数が下がっていき、マッチ閾値が 13% の時点で完全に 0 になった。また、そのときの groupT のマッチ数を見ると、どちらのセマンティクスでもほぼ 1 であり、提案手法のひとつめにする機能がほとんど意味をなしていない。しかし一方で、groupT のグループ数は図 9 とは異なり、徐々に下がっていき、マッチ閾値 13% では最高値の 81 から 30 下がって 51 である。これは、マッチ閾値による処理の打ち切りで解マッチとオーバーラップするマッチが削除されたことを示している。よって、マッチ閾値を適切な値に設定しても、ある程度大きなノイズの存在下ではマッチ集合に間違いが含まれてしまうことが分かった。すなわち、ノイズが大きければ、出力結果に間違いが含まれることは必然であり避けることはできない。

10. 考察

提案手法は、ある程度大きな一致確率をもつマッチにオーバーラップする小さな一致確率のマッチを如何にまとめることができるかが重要である。しかし、そのようなマッチはマッチ閾値により削除される傾向にあることが分かった。一方で、マッチ閾値を用いないと、解マッチにオーバーラップのないノイズのようなマッチが削除されず、部分オーバーラップの過度な連鎖を引き起こした。これらのことから、マッチ閾値ではない手法でノイズのようなマッチを削除すべきかもしれない。たとえば、実験 1 の最後で述べたように、解であるべきグループはそうでないグループに比べてマッチの数が平均的に多かったため、グループに含まれるマッチの数の比較から意味を成さないグループを削除できるかもしれない。今後調査が必要である。

11. 関連研究

関連研究について簡単に紹介する。イベントストリームに対するパターン問合せについては多くの研究がある。本研究は SASE+ [8] における問合せ処理方式 (特に NFA^b) をベースと

している。[8] では多くの最適化のアプローチが提案されているが、その一部は本研究にも適用可能と考える。確率的イベントストリームに対するイベントパターン問合せは Markovian Stream プロジェクト [5], [6] において研究されている。特に時間的な相関を持つマルコフ遷移による確率的推移データを対象としている。しかし、彼らの研究では問合せ結果をひとまとまりで検出することはできない。

12. まとめ

本稿では、確率的イベントストリームに対するイベントパターン問合せにおいて、ひとまとまりの結果を抽出するための二つのセマンティクスについて述べた。実験では、本手法によって、確率的イベントストリームに対するイベントパターン問合せからまとまりのある結果を得ることができることを示した。また、各セマンティクスの違いを示した。

今後の課題としては、今回の実験は限られたパターンとデータにおいて行なっているため、様々な種類のパターンやデータにおいても実験を行う必要がある。また、本稿ではストリーミング的な処理を対象としたが、蓄積された確率的イベントストリームに遡及的に問合せを行うことも考えられ、履歴ストリームに対する索引を用いること [12] などが考えられる。これについても今後の課題としたい。

謝辞

本研究の経費の一部は内閣府最先端研究開発プロジェクト (FIRST) による。

文献

- [1] HASC (Human Activity Sencing Consortium) ホームページ, <http://www.hasc.jp/>.
- [2] Y. Hattori and S. Inoue: "A large scale gathering system for activity data using mobile devices", Journal of Information Processing, **20**, 1, pp. 177-184 (2012).
- [3] 栗原, 福田, 菅原: "センサ情報からの系列パターンマイニング", 人工知能学会誌, **27**, 2, pp. 112-119 (2012).
- [4] The RFID Ecosystem, <http://rfid.cs.washington.edu/>.
- [5] J. Letchner, C. Ré, M. Balazinska and M. Philipose: "Challenges for event queries over Markovian streams", IEEE Internet Computing, **12**, 6, pp. 30-36 (2008).
- [6] C. Ré, J. Letchner, M. Balazinska and D. Suciu: "Event queries on correlated probabilistic streams", Proc. ACM SIGMOD, pp. 715-728 (2008).
- [7] G. Cugola and A. Margara: "Processing flows of information: From data stream to complex event processing", ACM Comput. Surv., **44**, 3 (2012).
- [8] J. Agrawal, Y. Diao, D. Gyllstrom and N. Immerman: "Efficient pattern matching over event streams", Proc. ACM SIGMOD, pp. 147-159 (2008).
- [9] R. Sadri, C. Zaniolo, A. Zarkesh and J. Adibi: "Optimization of sequence queries in database systems", Proc. ACM PODS, pp. 71-81 (2001).
- [10] F. Zemke, A. Witkowski, M. Cherniak and L. Colby: "Pattern matching in sequences of rows", ANSI Standard Proposal (2007).
- [11] エイホ, ラム, セシィ, ウルマン: "コンパイラ [第 2 版]: 原理・技法・ツール", サイエンス社 (2009).
- [12] J. Letchner, C. Ré, M. Balazinska and M. Philipose: "Access methods for Markovian streams", Proc. ICDE, pp. 246-257 (2009).