DBSJ Journal, Vol.12, No.1
June 2013

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
Event Pattern Queries on Probabilistic Event Streams

加藤 翔♡ 石川 佳治♢

Sho KATO　Yoshiharu ISHIKAWA

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．

Complex event processing **(CEP) is a task to detect high-level events from a large volume of stream data. In this paper, we focus on CEP for *probabilistic event streams* in which each event is assigned its occurrence probability. We propose two types of pattern query semantics to get a *group of matches* for a given regular expression pattern. A group of matches represents a semantic unit for considering high-level events.**

1. はじめに

今日ではセンサデータの有効活用が重要となってきている．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，センシングにより人々の行動状況をモニタリングする試みが進められている [3, 4, 5]．しかし，センサデータにはノイズが含まれることがあるため，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．つまり，行動認識の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，時区間と検出された行動の組で一般には与えられるが，認識結果の曖昧性を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，図 1 のように，行動認識プログラムが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．

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
Figure 1: Example of activity recognition

♡ 正会員　名古屋大学大学院情報科学研究科
現在の所属：古河機械金属
♢ 正会員　名古屋大学大学院情報科学研究科／国立情報学研究所
ishikawa@is.nagoya-u.ac.jp

本研究では，確率的イベントストリームに対し，イベントパターンの出現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，複合イベント処理（complex event processing, CEP）[2] においても基盤技術となっている [1]．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 [6, 8] 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

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 S は，$S = \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_t, \ldots$ と与えられる無限の系列である．$\boldsymbol{e}_t = \{e_{t1}, e_{t2}, \ldots, e_{t|V|}\}$ は，時刻 t におけるイベント集合（*event set*）であり，各 $e_{ti} \in \boldsymbol{e}_t$ 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，$\sum_{i=1}^{|V|} \Pr(e_{ti}) = 1$ が成り立つとする．　□

以下では，表記の簡略化のため，V の要素を a, b, ... 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，$\boldsymbol{e}_t = \{\mathsf{a}, \mathsf{c}, \mathsf{d}\}$ のように示す．

本研究では以下のような想定を行う．

1. 単位時間ごとにイベント集合が得られ，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．

2. 時刻 t におけるイベントの生起確率 $\Pr(e_{ti})$ $(i = 1, \ldots, |V|)$ 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，[6, 8] 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．

すなわち，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である．

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 $p = \mathtt{ab^+c}$ が問合せであるとする．ここで図 2 の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．

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）
Figure 2: Sample probabilistic event stream (1)

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[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．

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

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 $\mathrm{ts}(m)$ で表す．たとえば $\mathrm{ts}(m_1) = [1,4]$ となる．また，マッチ m の開始時刻，終了時刻を，それぞれ $\mathrm{start\_ts}(m), \mathrm{end\_ts}(m)$ 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 $m[t].\mathrm{ev}, m[t].\mathrm{pr}$ で表す．

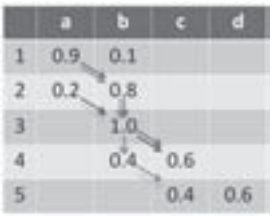


図 3: 単純照合の例
Figure 3: Simple match

マッチに対し，それがどの程度の確率で合致したかを表す一致確率（match probability）を計算できる．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 $p = \mathtt{ab^+c}$ 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．

定義 2 確率的イベントストリーム S にパターン p を適用したときの単純照合によるマッチの集合を $\mathrm{simple\_match}(S, p)$ で表す．□

[1]データ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, ... というデータに 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，[1] では *strict contiguity* と呼ばれている．

3.2 単純照合の問題点

単純照合は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検出したいという状況を考えると，それ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，実際にはひとまとまりの行動であるが，単純照合ではこれをまとまりとして抽出できていない．

そこで以下では，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

4.1 完全オーバラップ

完全オーバラップを次のように定義する．

定義 3 $M \subseteq \mathrm{simple\_match}(S, p)$ を単純照合によるマッチの部分集合とする．M が完全オーバラップ（*complete overlap*）の性質を有するとは，M が

$$\forall m, m' \in M \textit{ such that } m \neq m', \mathrm{ts\_overlap}(m, m') \tag{1}$$

を満たす極大な集合である場合をいう．$\mathrm{ts\_overlap}(m, m')$ は m と m' の時区間（$\mathrm{ts}(m)$ と $\mathrm{ts}(m')$）が交わるときに真になる．M に対する時区間を，M に含まれるマッチについて最大の時区間をとり，

$$\mathrm{start\_ts}(M) = \min\{\mathrm{start\_ts}(m) \mid m \in M\} \tag{2}$$

$$\mathrm{end\_ts}(M) = \max\{\mathrm{end\_ts}(m) \mid m \in M\} \tag{3}$$

により，

$$\mathrm{ts}(M) = [\mathrm{start\_ts}(M), \mathrm{end\_ts}(M)] \tag{4}$$

と定義する．□

完全オーバラップのセマンティクスを用いると，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，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．以下では，このようにして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

完全オーバラップにより得られるグループ M についてどのように確率を付与するかを考える．図 3 に示したマッチの例を見ると，単純照合の経路に重複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．行動認識のコンテクストで説明すると，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，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 $p = \mathtt{ab^+c}$ 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いる．

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

$$\Pr(M) = \prod_{t \in \mathrm{ts}(M)} \sum_{e \in \{m[t].\mathrm{ev} \mid m \in M\}} e[t].\mathrm{pr} \tag{5}$$

と定義する．□

ただし，$e[t].\mathrm{pr}$ 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 $\mathtt{a}[1].\mathrm{pr} = 0.9$ である．

4.3 部分オーバラップ

完全オーバラップのグループ 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がある．例として，図 4 について，先と同様に $\mathsf{ab}^{+}\mathsf{c}$ というパターン問合せを考える．単純照合の結果は

$$\begin{aligned} m_1 &= \langle(1,\mathsf{a}),(2,\mathsf{b}),(3,\mathsf{c})\rangle \\ m_2 &= \langle(1,\mathsf{a}),(2,\mathsf{b}),(3,\mathsf{b}),(4,\mathsf{c})\rangle \\ m_3 &= \langle(2,\mathsf{a}),(3,\mathsf{b}),(4,\mathsf{c})\rangle \\ m_4 &= \langle(4,\mathsf{a}),(5,\mathsf{b}),(6,\mathsf{c})\rangle \end{aligned}$$

となる．完全オーバラップに基づけば，$M = \{m_1, m_2, m_3\}$ および $M' = \{m_2, m_3, m_4\}$ が得られる．完全オーバラップは，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）
Figure 4: Sample probabilistic event stream (2)

そこで，制約を大幅に緩めた部分オーバラップを導入する．

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

$$\forall m \in M, \exists m' \in M \textit{ such that } m \neq m', \text{ts\_overlap}(m, m') \quad (6)$$

を満たす極大な集合であるとき，部分オーバラップ（*partial overlap*）の性質を有するという．M に対応する時区間を完全オーバラップの場合と同様に定義する． □

部分オーバラップの場合，先の例について $M = \{m_1, m_2, m_3, m_4\}$ という一つのグループが得られる．部分オーバラップでは，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．

4.4 確率の閾値の導入

絞り込みを可能とするため，確率の閾値を導入する．まず，グループに対する閾値を設定できる．この閾値を問合せ閾値（query threshold）と呼ぶ．たとえば，先に示した完全オーバラップの例で閾値を 20%と設定できる．その例ではグループの一致確率は 0.36 であったため，閾値を満たすため問合せ結果に含まれる．

一方，単純照合の個別のマッチに対する閾値をマッチ閾値（match threshold）と呼ぶ．先の例でマッチ閾値を 5%とした場合，m_4 が閾値を満たさないため，完全オーバラップに基づくグループは $M = \{m_1, m_2, m_3\}$ となる．マッチに対する閾値は，部分オーバ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．

5. アルゴリズム

5.1 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．

5.1.1 完全オーバラップにおけるグループの管理

まず，完全オーバラップにおける，新規ランの生成時のグループ管理の流れを図 5 に示す．入力は新規ランである．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*$(\{x_1, x_2, \ldots, x_n\})$ は，ランの集合 $\{x_1, x_2, \ldots, x_n\}$ 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，

表 1: 記号とその意味
Table 1: Symbols and their definitions

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）
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（$\forall t_l, t_l \in T$）

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，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）．なお，*getNewest*$(\{x_1, x_2, \ldots, x_n\})$ は，$min\{x_i.t_e | x_i \in \{x_1, x_2, \ldots, x_n\}\}$ を返す．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．

ランの失敗については，完全オーバラップの場合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．

```
1: procedure GroupMaintenanceForNewRun(r_new)
2:     if G ≠ ∅ then
3:         for all g_i ∈ G do
4:             if M_i ≠ ∅ then
5:                 m_temp ← 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 ← A_i ∪ {r_new}
11:        end for
12:        if ∀m ∈ 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
```

図 5: 完全オーバラップについての新規ラン生成時の処理
Figure 5: Processing a new run for complete overlap

5.1.2 部分オーバラップにおけるグループの管理

部分オーバラップのセマンティクスでは，グループの総数は 0 か 1 であり，グループ管理のアルゴリズムは単純になる．ただし，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．まず，新規ランの生成時におけるグループ管理の流れを図 6 に示す．

入力は新規ランである．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，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）

```
1: procedure GroupMaintenanceForNewRun(r_new)
2:     if G ≠ ∅ then
3:         for all g_i ∈ G do
4:             x_temp ← getNewest(A_i ∪ M_i)
5:             if x_temp.t_e == r_new.t_e then
6:                 A_i ← A_i ∪ {r_new}
7:                 break
8:             end if
9:         end for
10:        if ∀x ∈ A ∪ 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
```

図 6: 部分オーバラップについての新規ラン生成時の処理
Figure 6: Processing a new run for partial overlap

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．

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手法を図 7 に示す．

```
1: procedure GroupMaintenanceForRemovedRun(r_delete)
2: for all g_k ∈ G do
3:     Remove r_delete from A_k
4:     if ¬(∀x ∈ M_k ∪ A_k, ∃x′ ∈ M_k ∪ A_k such that x ≠ x′, ts_overlap(x, x′))
   then
5:         for all t_i ∈ T do
6:             for all m_j ∈ M_k do
7:                 if m_j.t_s < t_i then
8:                     X_temp ← X_temp ∪ {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 ∪ A)
16:        Delete g_k
17:    end if
18: end for
```

図 7: ラン除去の際のグループ管理（部分オーバラップ）
Figure 7: Processing run deletion for partial overlap

入力は，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，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し（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．

5.1.3 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．

5.2 確率の閾値に関するアルゴリズム

5.2.1 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．

5.2.2 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をその時点でのグループの一致確率とすると不適切な場合がある．たとえば，照合パターン $p = \mathrm{ab^+c}$ に対して図 8 左の確率的データストリームを考える．ここで現時刻を 4，問合せ閾値を 0.2 とする．現時刻におけるグループの一致確率は，$0.9 \times 1.0 \times (0.2 + 0.8) \times 0.1 = 0.09$ であり，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ができない．

	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 8: 不適切な処理となる例
Figure 8: Problematic cases of processing

先の例では，アクティブランを除いたグループで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，それでも不適切な状況がある．先と同じ照合パターンに対し図 8 右の確率的データストリームを考える．現時刻を 3，問合せ閾値を 0.2 とする．現時刻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で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．

6. 評価実験

6.1 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 9 に示す，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

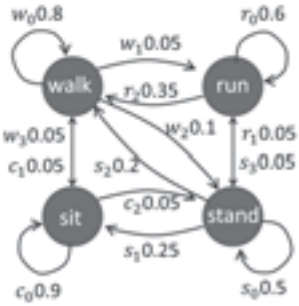


図 9: イベントストリーム生成のモデル
Figure 9: Event stream generation model

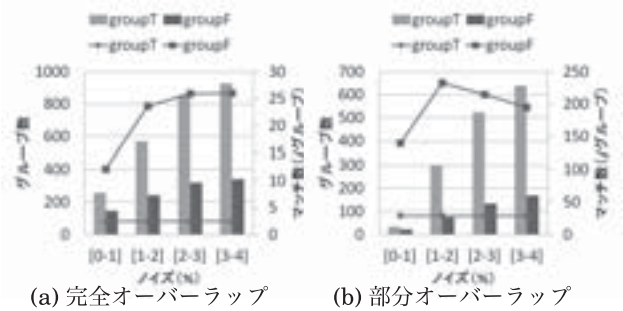
の元となった決定的イベントストリームに対して単純照合のセマンティクスで得られるマッチ集合を，検出されるべき 解（solution）であるとし，解に含まれるマッチを 解マッチ（solution match）として評価に用いた．なお，解マッチの数は 81 である．実験で用いる人工データのパラメータの基本情報を表 2 に記す．

表 2: 人工データのパラメータ
Table 2: Parameters for synthetic data

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%]

6.2 実験 1：グループ化能力の評価（閾値設定なし）

まず，閾値を設定せずに実験を行った．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となる．決定的データ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 $[0,1],[1,2],\ldots,[3,4]$ と最大・最小ともに 1%ずつ変化させて，四つの確率的データ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 walk sit$^+$ 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，完全オーバラップと部分オーバ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 10 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．



(a) 完全オーバーラップ　　(b) 部分オーバーラップ
図 10: グループ化能力の評価
Figure 10: Performance of grouping power

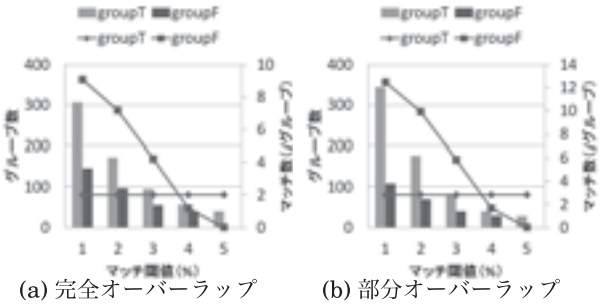
groupT は，出力されたグループ集合のうち，解マッチとオーバラップしているグループの集合を表す．解マッチの数は 81 であるため，groupT のグループ数は 81 が上限である．groupF は，出力されたグループ集合のうち，解マッチにオーバラップがないグループの集合であり，groupF のグループ数は，間違った結果の出力数である．図を見ると，どちらのセマンティクスでも groupT のグループ数は常に 81 であり，最高の結果である．一方，groupF のグループ数についてはどちらのセマンティクスでも groupT の数倍以上ある．これは，閾値を設定しない問合せでは，問合せ結果における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．間違った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，単純照合の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 も同様である．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．

完全オーバラップと部分オーバ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，部分オーバラップは平均して完全オーバラップの 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．

6.3 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 11 に示す．

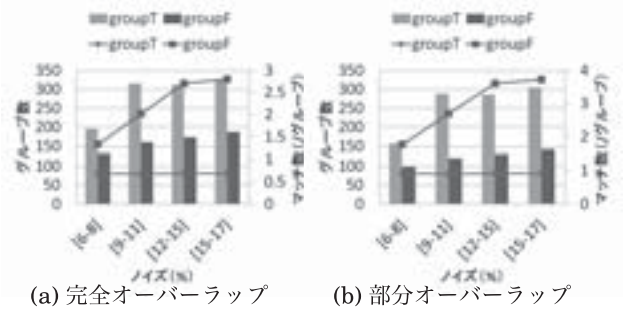


(a) 完全オーバーラップ　　(b) 部分オーバーラップ
図 11: ノイズ [3, 5] 固定でマッチ閾値を変化
Figure 11: Changing match threshold with noise [3, 5]

図の groupF を見ると，マッチ閾値を増やすにつれて，間違ったグループ数が減少し，マッチ閾値が 5%の時点で完全に 0 になる．一方，groupT を見ると，解マッチとオーバ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．よって，ノイズの存在のもとで，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．これは，提案手法のひとまとめにする機能がほとんど機能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なさない．

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 12 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 と大きく異る点は，部分オーバラップにおける groupT のグループ数が完全オーバラップのそれとほぼ同じである点である．部分オーバラップは定義上どこまでも連鎖が続く可能性があり，その現象が実験 1 で見られた．今回の結果から，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と示された．

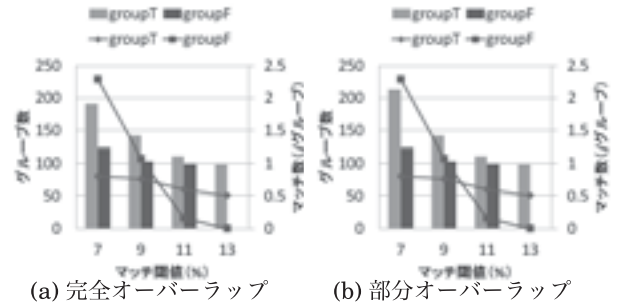


(a) 完全オーバーラップ　　(b) 部分オーバーラップ

図 12: マッチ閾値 0.05 固定でノイズを変化

Figure 12: Changing noise with match threshold 0.05

実験 2 の最初の実験では，間違ったマッチを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 13 に示す．図 11 と同様，マッチ閾値の増加に伴って groupF のグループ数が減り，マッチ閾値が 13%の時点で完全に 0 になる．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，提案手法のひとまとめにする機能がほとんど意味をなしていない．一方で，groupT のグループ数は図 11 とは異なり，徐々に下がり，マッチ閾値 13%では最高値の 81 から 30 下がって 51 となる．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ラップするマッチが削除されたことを意味する．よって，マッチ閾値を適切に設定しても，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．



(a) 完全オーバーラップ　　(b) 部分オーバーラップ

図 13: ノイズ [15, 17] 固定でマッチ閾値を変化

Figure 13: Changing match threshold with noise [15, 17]

7. 考察

提案手法では，ある程度大きな一致確率を持つマッチにオーバラップする小さな一致確率のマッチをどのようにまとめることができるかが重要なポイントである．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，マッチ閾値を用いないと，解マッチにオーバラップのないノイズのようなマッチが削除されず，部分オーバラップの過度な連鎖を引き起こす．また，ノイズのようなマッチが削除されないことで完全オーバラップにおいてもグループの一致確率が過度に下がっていた．これらのことから，マッチ閾値では以外の手法で，ノイズであるマッチを削除すべきかもしれない．

グループの一致確率の計算方法には，改良の余地があると考えている．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いため，グループに含まれるマッチの数の比較から意味をなさないグループを削除できるかもしれない．そのためには，どのような条件の下でグループを削除するかの指標を開発する必要があるが，より質の高い結果が得られる可能性がある．

8. 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+ [1] における問合せ処理方式（特に NFA^b）をベースとしている．SASE+ 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 [6, 8] 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．

9. 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，本稿ではストリーム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 [7] などが考えられる．

[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] J. Agrawal, Y. Diao, D. Gyllstrom, and N. Immerman. Efficient pattern matching over event streams. In *Proc. ACM SIGMOD*, pp. 147–159, 2008.

[2] G. Cugola and A. Margara. Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, 44(3), 2012.

[3] HASC (Human Activity Sencing Consortium) ホームページ. http://www.hasc.jp/.

[4] Y. Hattori and S. Inoue. A large scale gathering system for activity data using mobile devices. *Journal of Information Processing*, 20(1):177–184, Jan. 2012.

[5] 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):112–119, Mar. 2012.

[6] J. Letchner, C. Ré, M. Balazinska, and M. Philipose. Challenges for event queries over Markovian streams. *IEEE Internet Computing*, 12(6):30–36, 2008.

[7] J. Letchner, C. Ré, M. Balazinska, and M. Philipose. Access methods for Markovian streams. In *Proc. ICDE*, pp. 246–257, 2009.

[8] C. Ré, J. Letchner, M. Balazinksa, and D. Suciu. Event queries on correlated probabilistic streams. In *Proc. ACM SIGMOD*, pp. 715–728, 2008.

加藤 翔　**Sho KATO**

2013 年名古屋大学大学院情報科学研究科修了．大学院においては確率的データストリーム処理に関する研究を行う．現在，古河機械金属 (株) に勤務．

石川 佳治　**Yoshiharu ISHIKAWA**

名古屋大学大学院情報科学研究科教授．データベース，データ工学，情報検索等に興味を持つ．日本データベース学会，情報処理学会，電子情報通信学会，人工知能学会，ACM，IEEE 各会員．