確率的データストリームにおける 情報利得を用いたTop-kパターン照合手法

杉浦　健人†　　石川　佳治†

† 名古屋大学大学院情報科学研究科　〒464–8601 愛知県名古屋市千種区不老町

E-mail: sugiura@db.ss.is.nagoya-u.ac.jp, ishikawa@is.nagoya-u.ac.jp

あらまし　センシングデータと教師あり学習を用いた行動認識は近年活発に研究されており，複合イベント処理などの更なる応用が期待されている．しかし，センサの精度や周囲の環境しだいで認識の正確さは容易に落ちてしまうため，単純に最尤の認識結果を選ぶだけではユーザが検出したいイベントを見逃してしまう可能性がある．そこで本稿では，認識結果が確率的データストリームとして表されることを想定し，確率的データストリーム上でのパターン照合による複合イベント処理について述べる．まず，非確率的データストリームにおけるパターン照合のセマンティクスを紹介し，確率的データストリームにおける適切なセマンティクスについて議論する．次に，生起確率の大きい照合結果を単純に選ぶだけでは適切な照合結果が得られないことを示し，情報利得に基づく照合結果の選出を提案する．最後に，Top-k の情報利得を持つ照合結果を効率的に得るアルゴリズムを提案し，実データ及び人工データを用いた実験により提案手法の有効性と効率性を示す．

キーワード　確率的データストリーム, 複合イベント処理, パターン照合, 正規表現, Top-k 問合せ

1. はじめに

センサ技術及び無線通信技術の発展により，スマートフォンやウェアラブルデバイスなどを用いた行動認識が活発に行われている [1–3]．これらの行動認識は主に機械学習，特に教師あり学習を基に行われており，「歩く」や「ジョギングする」などの単純なイベントの検出が対象となっている．しかし，それらのイベントはある時点における行動のみを表すため，情報としては単純過ぎる．したがって，**複合イベント処理** [4] を用いて「ある期間歩き続けている」などのより複雑なイベントを検出することが求められている．

行動認識のような教師あり学習の結果に対して複合イベント処理を行うとき，分類結果が不確実性を含む**確率的データストリーム**である点に注意する必要がある．教師あり学習を用いて分類を行うとき，用いるセンサの精度や分類の境界の曖昧さなどにより，分類結果の信頼度は容易に落ちてしまう．例えば，スマートフォンの加速度センサを用いて「歩く（`walk`）」,「階段を上る（`stUp`）」,「階段を下る（`stDown`）」などの行動を認識する場合を考える．これらの行動の動作は似ているため，加速度センサのみで正確に分類するのは難しい．加えて，現実的な使用環境では測定値のノイズなど，分類モデルの学習時には想定されていない要素も入りうるため，分類は更に難しくなる．そのため図 1 のように，各分類結果が生起した可能性を持つデータストリーム，つまり確率的データストリームが得られる．

本稿では確率的データストリームに対する複合イベント処理として**パターン照合**を適用し，適切な照合結果（マッチ）の検出を目指す．確率的データストリームに対する既存研究では，選択や射影，集約 [5,6]，Top-k 問合せ [7]，頻出アイテムの検出 [8]，クラスタリング [9] などが行われている．しかしこれらの研究では，複合イベントを表す問合せを記述することも，複合イベント処理を効率的に行うことも難しい．そこで本稿では，**正規表現**に基づくパターン照合を用いることで，柔軟な問合せの記述と効率的な複合イベント処理の実現を目指す．

event symbol	time step						
	1	2	3	4	5	6	7
`stay (s)`	0.15	0.25	0.80	0.05	0.05	0.05	0.05
`walk (w)`	0.20	0.20	0.05	0.40	0.40	0.40	0.40
`jog (j)`	0.25	0.15	0.05	0.05	0.05	0.05	0.05
`stUp (u)`	0.20	0.20	0.05	0.25	0.25	0.25	0.25
`stDown (d)`	0.20	0.20	0.05	0.25	0.25	0.25	0.25

図 1　確率的データストリーム

確率的データストリームにパターン照合を適用したとき，マッチの適切さをどのように表すかが問題となる．データの不確実性により，確率的データストリームにおけるパターン照合では多数のマッチが得られる．全てのマッチをユーザの手で確認するのは非現実的であるため，何かしらの尺度に基づき適切なマッチを選別する必要がある．既存研究 [10] では適切さの尺度としてマッチの生起確率を用いているが，生起確率にはクリーネ閉包を含むパターンを適切に評価できない，機械学習の精度を考慮できないという二つの問題がある．例として，図 1 にパターン $p = \langle \mathtt{w}^+ \rangle$ を照合することで得られる以下の二つのマッチについて考える．なお，イベントの添字は時刻を表し，これ以降データストリームにおける時刻 t_s から t_e までの時区間を $[t_s : t_e]$ で表す．

$m_1 = \langle \mathtt{w}_1, \mathtt{w}_2 \rangle$,　　$P(m_1) = 4.0 \times 10^{-2}$

$m_2 = \langle \mathtt{w}_4, \mathtt{w}_5, \mathtt{w}_6, \mathtt{w}_7 \rangle$,　　$P(m_2) \simeq 2.6 \times 10^{-2}$

生起確率のみを見ると，m_1 の方が m_2 よりも優れている．しかし，これは m_2 が多くのイベントからなる複雑なイベントを表しているためであり，一概に m_2 が劣っているとは言えない．むしろ，連続的な生起を表すクリーネ閉包（$^+$）が付与されていることを考慮すると，m_2 の方が適切であるとも考えられる．このように生起確率には，複雑なイベントを示すマッチを低く評価する傾向がある．更に，図 1 の時区間 [1 : 2] と [4 : 7] を比べると，時区間 [1 : 2] はそもそも分類が上手くいっていないことがわかる．したがって時区間 [1 : 2] からのマッチの検出は適切でないと考えられるが，生起確率の絶対値を単純に比べる場合，そのような元々の分類の精度を考慮することもできない．

本稿では，マッチの適切さを表すための指標として，マッチの相対的な生起確率の大きさを表す**情報利得**を導入する．上述の問題の原因は，生起確率の絶対値に基づきマッチを比較している点である．そこで，同じ時区間に存在する他の系列に比べて，検出したマッチの生起確率が相対的に大きいかどうかに注目する．例として，上述の m_1 及び m_2 について考える．時区間 [1 : 2] には $\langle \mathtt{s}_1, \mathtt{s}_2 \rangle, \langle \mathtt{s}_1, \mathtt{w}_2 \rangle$ など，合計 5^2 個の系列 $S_{[1:2]}$ が存在する．系列の生起確率の幾何平均 $GM(S_{[1:2]})$ は 3.9×10^{-2} であり，m_1 の生起確率とほぼ等しい．つまり，m_1 は同じ時区間の系列に比べて，特別生起確率が大きいマッチではないことがわかる．一方で，時区間 [4 : 7] には $\langle \mathtt{s}_4, \mathtt{s}_5, \mathtt{s}_6, \mathtt{s}_7 \rangle$ など合計 5^4 個の系列 $S_{[4:7]}$ が存在し，その幾何平均 $GM(S_{[4:7]})$ は 4.3×10^{-4} である．したがって，m_2 は同じ時区間の系列に比べて，生起確率が非常に大きいことがわかる．この関係をマッチの生起確率と系列の幾何平均の比で表し，更に対数をとることで，以下のように情報利得の式が得られる．なお，$m.t_s$ 及び $m.t_e$ はそれぞれマッチの開始・終了時刻を表す．

$$IG(m) = \log_2 \frac{P(m)}{GM(S_{[m.t_s:m.t_e]})} \tag{1}$$

マッチの情報利得 $IG(m)$ は，マッチ m が同じ時区間に存在する他の系列に比べて生起確率が大きいとき正の，小さいとき負の値をとるマッチの評価関数となる．

本稿では，上述したマッチの情報利得に基づく，パターン照合結果の *Top-k* **問合せ**について述べる．本稿の構成は以下のとおりである．まず，2. で関連研究を紹介し，3. で入力となるパターンなどの定義を行う．その後，4. で本稿で扱うパターン照合の Top-k 問合せについて定義し，5. で照合アルゴリズムを具体的に述べる．最後に，6. で提案手法を実験により評価し，7. でまとめを述べる．

2. 関 連 研 究

2.1 確率的データストリーム

確率的データストリームに対してパターン照合を行う研究として，イベント生起の時間的な相関を考慮したものがある [11]．例えば，ある時刻で階段を上っていた人物が，次の時刻で急に階段を下り始める可能性は小さいと考えられる．文献 [11] では，このような時間的な相関を一階マルコフ過程を用いて表している．しかしこの研究では，パターン照合の結果として各時刻でマッチが検出されるかされないかのみを考えており，本稿のようにマッチの系列としての適切さは評価していない．そのため，本稿のような具体的なイベント系列に注目するパターン照合には適用できない．また，各イベントの生起確率の大きさにのみ注目しているため，本稿で提案する情報利得のように入力された確率的データストリームの分類精度を考慮することもできない．

一方で，ある時間窓内で Top-k の生起確率を持つマッチを検出する研究も提案されている [10]．この研究では生起確率を指標としていることに加え，パターン照合におけるイベントのスキップを前提としている点が問題となる．例えば，図 1 でパターン $p = \langle \mathtt{j}^+ \ \mathtt{w}^+ \rangle$ を照合した場合を考える．このとき，確率の小さいイベントは無視され $\langle \mathtt{j}_1, \mathtt{w}_4 \rangle, \langle \mathtt{j}_1, \mathtt{w}_5 \rangle$ などが Top-1 の生起確率のマッチとして検出される．しかし，行動認識などのようにイベントの連続的な生起に注目する場合，$\langle \mathtt{j}_1, \mathtt{w}_2 \rangle$ の方が適切なマッチだと考えられる．したがって，イベントの連続的な生起に注目するシナリオではこの研究は適用できない．

2.2 非確率的データストリーム

入力となるデータに不確実性が存在しない非確率的データストリームに対してパターン照合を行う研究は数多く提案されている．例えば，パターン照合に詳細な条件の指定を許したもの [12,13] や，イベントが時刻通りに入力されないもの [14,15]，分散環境での使用を想定したもの [16]，イベントの生起頻度を利用して効率的な照合を行うもの [17]，FPGA（field-programmable gate array）を利用したもの [18] などがある．しかし，確率的データストリームに対するパターン照合はまだ成熟していないため，本稿ではまず基礎的なパターン照合について議論する．

また，非確率的データに対するパターン照合として文字列照合が古くから研究されている．特に，正規表現から NFA（非決定性有限オートマトン）を生成するトンプソンの構築法 [19] は強力で，Knuth-Morris-Pratt 法 [20] や Aho-Corasick 法 [21] など，効率的なパターン照合を実行する NFA を容易に生成できることが示されている [22]．本稿でも正規表現とトンプソンの構築法を活用し，マッチの検出を行う．

3. 準　　備

パターン照合の入力となる確率的データストリーム *PDS* 及び問合せパターン p を定めた後，マッチの情報利得を定める．

3.1 確率的データストリーム

まず，確率的データストリームの要素となる**確率的イベント**を以下に示す．

［定義 1］ **確率的イベント** e_t は，各イベントシンボル $\alpha \in \Sigma$ に対して生起確率 $P(e_t = \alpha)$ を持つ時刻 t のイベントである．ただし，Σ はイベントシンボルの全集合を示す．また，イベントの生起確率 $P(e_t = \alpha)$ は以下の式を満たす．

$$\forall \alpha \in \Sigma,\ 0 \leqq P(e_t = \alpha) \leqq 1 \tag{2}$$

$$\forall \alpha, \beta \in \Sigma,\ \alpha \neq \beta \rightarrow P(e_t = \alpha \wedge e_t = \beta) = 0 \tag{3}$$

$$P(\bigvee_{\alpha \in \Sigma} e_t = \alpha) = \sum_{\alpha \in \Sigma} P(e_t = \alpha) = 1 \tag{4}$$

□

簡略化のため，これ以降 $e_t = \alpha$ を α_t で表す．なお，提案手法は文献 [11] のように確率的イベントがマルコフ性を持つ場合にも拡張できるが，議論を簡潔にするために，本稿では時間に関して確率的な独立性を想定する．

確率的イベントを用いて，**確率的データストリーム**を以下のように定義する．

［定義 2］ **確率的データストリーム** $PDS = \langle e_1, e_2, ..., e_n \rangle$ は，確率的イベントの系列である． □

例えば，図 1 の確率的データストリームは，$\Sigma = \{\mathtt{s}, \mathtt{w}, \mathtt{j}, \mathtt{u}, \mathtt{d}\}$ である確率的イベント e_t の系列 $PDS = \langle e_1, e_2, ..., e_7 \rangle$ として表せる．

3.2 問合せパターンとマッチ

問合せパターンの記述方法を以下のように定める．

［定義 3］ 入力となるパターン p は以下の文法から生成される．なお，$\alpha \in \Sigma$ であり，ϵ は空イベントを示す．

$$p ::= \alpha \mid \epsilon \mid p\ p \mid p \vee p \mid p^* \mid p^+ \mid (p) \tag{5}$$

□

つまり，本稿ではパターンが一般的な正規表現によって記述されることを想定する．

パターンに適合するマッチとその生起確率を定める．

［定義 4］ マッチ m は，ユーザが指定したパターン p に適合するイベントシンボル α_t の系列である．マッチ m の開始・終了時刻をそれぞれ $m.t_s, m.t_e$ で表すとし，マッチの生起確率 $P(m)$ は以下の式で求める．

$$P(m) = \prod_{t=m.t_s}^{m.t_e} P(\alpha_t) \tag{6}$$

□

3.3 マッチの情報利得

マッチの情報利得を定める．

［定義 5］ マッチ m の開始・終了時刻をそれぞれ $m.t_s, m.t_e$ とする．また，ある時区間 $[t_s : t_e]$ に存在する系列の全集合を $S_{[t_s:t_e]}$ で表し，$GM(S)$ で集合 S の各要素の生起確率の幾何平均を表す．このとき，マッチの情報利得 $IG(m)$ を式 (1) で求める． □

マッチの情報利得は，各時刻で選択したイベントシンボルの情報利得の総和として計算できる．まず，イベントの生起確率が時間的に独立であることを想定しているため，幾何平均 $GM(S_{[t_s:t_e]})$ は以下のように書き換えられる．なお，E_t は $P(\alpha_t) > 0$ を満たすイベントシンボル $\alpha \in \Sigma$ の集合とする．

$$GM(S_{[t_s:t_e]}) = \prod_{t=t_s}^{t_e} GM(E_t) \tag{7}$$

式 (6), (7) を式 (1) に代入する．

$$\begin{aligned} IG(m) &= \log_2 \frac{\prod_{t=m.t_s}^{m.t_e} P(\alpha_t)}{\prod_{t=m.t_s}^{m.t_e} GM(E_t)} \\ &= \sum_{t=m.t_s}^{m.t_e} \log_2 \frac{P(\alpha_t)}{GM(E_t)} \end{aligned} \tag{8}$$

マッチの情報利得を各時刻におけるイベントの情報利得の総和として表すことで，計算を高速化できる．例として，図 1 におけるマッチ $\langle \mathtt{w}_1, \mathtt{w}_2 \rangle$ について考える．時区間 [1 : 2] における系列の幾何平均は約 3.9×10^{-2} であるため，式 (1) に基づく場合は以下のように計算できる．

$$IG(\langle \mathtt{w}_1, \mathtt{w}_2 \rangle) = \log_2 \frac{P(\langle \mathtt{w}_1, \mathtt{w}_2 \rangle)}{GM(S_{[1:2]})} \simeq 3.7 \times 10^{-2}$$

式 (1) では，幾何平均を求めるために全系列 $S_{[1:2]}$ の生起確率を計算しなければならない点が問題となる．系列数は時区間が大きくなるにつれ指数関数的に増加するため，全ての生起確率を計算するのは効率が悪い．一方，式 (8) に基づく場合は，各時刻における情報利得の和として以下のように計算できる．

$$\begin{aligned} IG(\langle \mathtt{w}_1, \mathtt{w}_2 \rangle) &= \log_2 \frac{P(\mathtt{w}_1)}{GM(E_1)} + \log_2 \frac{P(\mathtt{w}_2)}{GM(E_2)} \\ &\simeq 1.86 \times 10^{-2} + 1.86 \times 10^{-2} \\ &\simeq 3.7 \times 10^{-2} \end{aligned}$$

式 (8) では各時刻で E_t の幾何平均を求めるだけでよいため，時区間の大きさに対する計算量は線形で抑えられる．

4. 情報利得に基づくパターン照合

適切なマッチを検出するために，まずパターン照合のセマンティクスを考える．その後，本稿で扱うパターン照合の問題定義を示す．

4.1 パターン照合のセマンティクス

存在する全てのマッチに対して単純に Top-k 問合せを行ったとき，不要なマッチをいくつも出力してしまう可能性がある．例えば，図 1 に対してパターン $p = \langle \mathtt{w}^+ \rangle$ を照合し，Top-2 の情報利得を持つマッチを検出する場合を考える．このとき図 2 の m_2 が Top-1 のマッチ，m_3 が Top-2 のマッチとなる．しかし，m_3 は m_2 の部分系列であり，検出することで得られる情報は重複している．そもそも実世界で実際に生起したイベントを考えると，m_2 及び m_3 は「時区間 [4 : 7] 付近で歩いていた」という同じイベントに対応しており，二つとも検出する意義は小さい．むしろ，m_1 などの他の時区間に存在するマッチを検出する方が有意義である．

そこで，非確率的データストリームで用いられたセマンティクスを参考に，確率的データストリームにおいて適切なセマンティクスを考える．既存研究 [23] では，正規表現を用いたパターン照合のセマンティクスを以下の三つにまとめている．

（1） all-path セマンティクス

（2） no-overlap セマンティクス

match	time step							$IG(m)$
	1	2	3	4	5	6	7	
m_1	w	w						0.04
m_2				w	w	w	w	5.88
m_3				w	w	w		4.41

図 2 $p = \langle \mathtt{w}^+ \rangle$ のマッチと情報利得

（3） use-and-throw セマンティクス

all-path セマンティクスでは，その名の通り存在する全てのマッチを出力の候補とする．しかし，前述のように，不要なマッチを多数検出する可能性があるためこのセマンティクスは不適切だと考える．

no-overlap セマンティクスは，時間的に重複しないマッチの組合せを出力する．図 2 の場合，$\{m_1, m_2\}$ や $\{m_1, m_3\}$ の組合せは出力できるが，$\{m_2, m_3\}$ は出力できない．no-overlap セマンティクスを用いるとき，任意のイベントシンボルを示す “.” を用いることで，出力されるマッチの組合せを正規表現 $\langle(. \vee p)^*\rangle$ で記述できる．つまり，マッチの各組合せを図 3 のように一つの系列として表せる．正規表現 $\langle(. \vee p)^*\rangle$ から照合用の有限オートマトンを生成することもできるため，Viterbi アルゴリズム [24] などの既存アルゴリズムも利用できる．しかし，no-overlap セマンティクスには，選言を含むパターンが与えられたとき実用的に同時に検出したいマッチを同時に出力できないという問題がある．例として，パターン $p = \langle \mathrm{u}^+ \vee \mathrm{d}^+\rangle$ が与えられた場合を考える．“u” と “d” はそれぞれ異なるイベントを表すため，ユーザとしては $\langle \mathrm{u}_1, \mathrm{u}_2\rangle$ 及び $\langle \mathrm{d}_1, \mathrm{d}_2\rangle$ などのマッチを同時に検出したい場合も考えられる．しかし，これらのマッチは時間的に重複しているため，no-overlap セマンティクスでは出力できない．

use-and-throw セマンティクスは，上述した no-overlap セマンティクスの問題を解決するために文献 [23] で提案されたものであり，入力されたイベントを一度だけ照合に使用する．例として，図 1 にパターン $p = \langle \mathrm{u}^+ \vee \mathrm{d}^+\rangle$ を照合した場合を考える．$\{\langle \mathrm{u}_1, \mathrm{u}_2\rangle, \langle \mathrm{d}_2, \mathrm{d}_3\rangle\}$ の組合せは，共有するイベントがないため出力の候補となる．一方 $\{\langle \mathrm{u}_1, \mathrm{u}_2\rangle, \langle \mathrm{u}_2, \mathrm{u}_3\rangle\}$ は，“u_2” を 2 回使用しているため候補とならない．use-and-throw セマンティクスは柔軟にマッチを検出できるが，no-overlap セマンティクスのように解の候補を正規表現で表せない．つまり，有限オートマトンによる解の受理ができないため，既存アルゴリズムを利用した効率的な照合は難しい．加えて，単一の問合せ内で複雑な制限を課しているため，YFilter [25] などで行われている複数問合せの同時処理による効率化も難しい．

以上の点を考慮し，本稿では no-overlap セマンティクスに基づくパターン照合を行う．no-overlap セマンティクスには，$p = \langle \mathrm{u}^+ \vee \mathrm{d}^+\rangle$ などの選言を含むパターンに対し異なるイベントからなるマッチを同時に出力できないという問題点があるが，これは $\langle \mathrm{u}^+\rangle$ 及び $\langle \mathrm{d}^+\rangle$ を別々の問合せとして扱うことで解決できる．no-overlap セマンティクスは正規表現 $\langle(. \vee p)^*\rangle$ から生成される有限オートマトンによって解を検出できるため，YFilter [25] などのような複数問合せの同時処理も可能である．なお，複数問合せの具体的な処理方法は今後の課題とし，本稿では単一の問合せに対する処理の効率化のみを述べる．

sequence	time step							matches
	1	2	3	4	5	6	7	
s_1	.	.	.	.	.	.	.	$\emptyset$
s_2	.	.	.	w	w	w	w	$\{m_2\}$
s_3	.	.	.	w	w	w	.	$\{m_3\}$
s_4	w	w	.	.	.	.	.	$\{m_1\}$
s_5	w	w	.	w	w	w	w	$\{m_1, m_2\}$
s_6	w	w	.	w	w	w	.	$\{m_1, m_3\}$

図 3　図 2 におけるマッチの組合せと対応する系列

4.2 問 題 定 義

no-overlap セマンティクスを適用した，パターン照合結果の Top-k 問合せを考える．no-overlap セマンティクスに基づく場合，出力できるマッチの組合せが制限されるため，単純に情報利得の大きいマッチを順に選択することはできない．したがって，マッチ集合の要素数が k 以下であるという条件下で，情報利得の総和を最大化することを考える．通常の Top-k 問合せであれば，スコアの大きい順に解を選ぶため，解のスコアの総和も最大となる．そこで，no-overlap セマンティクスを満たすマッチ集合のうち，情報利得の総和が最大となる集合を選ぶことで Top-k 問合せを実行する．なお，要素数が k「以下」であることを許すのは，k の値が大きいときにマッチの不要な分割が行われないようにするためである．例えば，図 1 にパターン $p = \langle \mathrm{w}^+\rangle$ を照合し，Top-3 問合せを行う場合を考える．Top-2 であれば図 2 の $\{m_1, m_2\}$ の情報利得が最大となる．しかし Top-3 の場合，この集合に $\langle \mathrm{w}_3\rangle$ を新しいマッチとして加えるよりも，m_2 を $\langle \mathrm{w}_4, \mathrm{w}_5\rangle$ と $\langle \mathrm{w}_6, \mathrm{w}_7\rangle$ などに分割する方が情報利得の総和は大きくなる．このような分割を避けるため，要素数が k 以下であることを条件にし，Top-3 問合せであっても $\{m_1, m_2\}$ を解として出力できるようにする．

本稿におけるパターン照合の問題定義を以下にまとめる．

［定義 6］ 確率的データストリーム PDS，パターン p，出力するマッチの最大数 k が入力されたとき，以下の式を満たすマッチ集合 M を検出する．なお，$\mathrm{ts\_overlap}(\mathrm{m}_1, \mathrm{m}_2)$ は m_1 と m_2 の時間的なオーバラップを示す述語である．

$$\begin{aligned} &\text{maximize} && \sum_{m \in M} IG(m) \\ &\text{subject to} && |M| \leqq k \\ & && \forall m_1, m_2 \in M, \neg \mathrm{ts\_overlap}(\mathrm{m}_1, \mathrm{m}_2) \end{aligned} \tag{9}$$

□

5. 照 合 方 法

全てのマッチを一度検出し情報利得が最大となる組合せを探すことで解は得られるが，この方法は明らかに効率が悪い．クリーネ閉包を含む場合マッチの数は多項式関数的に増えるため，入力されるデータストリームのサイズによってはマッチの検出だけで時間がかかる．加えて，候補となるマッチの組合せは指数関数的に増えるため，全組合せの列挙は困難である．

そこで，情報利得が最大となる集合 M を検出するという問題を，情報利得が最大となる系列 $\langle(. \vee p)^*\rangle$ を検出する問題として考える．no-overlap セマンティクスに基づくとき，4.1 で述べたように，集合 M は図 3 のような一つの系列として表せる．つまり，情報利得が最大となる系列を見つけ，その系列に含ま

れるマッチを集合にまとめることで，解が得られる．なお，系列の情報利得は系列に含まれるマッチの情報利得の総和とし，任意のイベント “.” が持つ情報利得は 0 とする．

5.1 動的計画法に基づく照合

正規表現 $\langle(. \vee p)^*\rangle$ に対応する ϵ-NFA（非決定性有限オートマトン）を利用する．ϵ-NFA の生成は以下の 4 ステップで行う．

（1） パターン p に対応する ϵ-NFA をトンプソンの構築法を用いて生成する [19].

（2） ϵ-NFA を等価な DFA（決定性有限オートマトン）に変換，最小化する [19].

（3） 全最終状態から初期状態に向け ϵ-遷移を追加し，最終状態を初期状態のみとする．

（4） “.” で初期状態自身に移る遷移を加える．

つまり，パターン p に対応する DFA に対して，照合を繰り返すために初期状態に移る ϵ-遷移と，初期状態で “.” を繰り返す遷移を加えることで生成する．例えば，パターン $p = \langle \mathtt{w}^+ \rangle$ が入力されたとき，対応する ϵ-NFA は図 4 となる．

本稿では，Viterbi アルゴリズム [24] などと同様に，動的計画法に基づく照合アルゴリズムを提案する．式 (8) に示したマッチの情報利得と同様に，系列 $\langle(. \vee p)^*\rangle$ の情報利得も各イベントシンボルの情報利得の総和で求められる．つまり，ある時刻ある状態で情報利得の大小が定まったとき，その大小関係はそれ以降も変化しない．例として，図 3 の系列を考える．時刻 3 までの時点では $\langle ._1, ._2, ._3\rangle$ 及び $\langle \mathtt{w}_1, \mathtt{w}_2, ._3\rangle$ の二つの系列が考えられる．これらの系列は共に図 4 の初期状態に到達しているため，時刻 4 以降に追加されるイベントは同じである．つまり，時刻 4 以降これらの系列に加算される情報利得の値は等しい．$\langle ._1, ._2, ._3\rangle$ の情報利得は 0，$\langle \mathtt{w}_1, \mathtt{w}_2, ._3\rangle$ の情報利得は約 0.04 であるため，情報利得を最大化する場合，時刻 4 以降は $\langle \mathtt{w}_1, \mathtt{w}_2, ._3\rangle$ のみを処理すればよい．このように，データストリームを頭から処理し，各時刻各状態で情報利得が最大の系列を保持することで，情報利得が最大の系列を検出できる．

ただし，定義 6 のようにマッチ集合の要素数 $|M|$ を k 以下に制限する場合，上述の方法では解が検出できない．上述の例で $k = 1$ である場合を考える．$k = 1$ の場合，図 3 の $s_2 = \langle ._1, ._2, ._3, \mathtt{w}_4, \mathtt{w}_5, \mathtt{w}_6, \mathtt{w}_7\rangle$ が最適解となる．しかし，各時刻各状態で情報利得が最大の系列を保持するだけでは，時刻 3 の時点で $\langle ._1, ._2, ._3\rangle$ が破棄されるため，s_2 は検出できない．

そこで，図 5 のように系列内に含まれるマッチの数に応じて照合の段階を区別することで，Top-k 問合せを実行する．なお，図 5 はパターン $p = \langle \mathtt{w}^+ \rangle$ の照合における状態遷移図である．この状態遷移図において，系列は図 4 に示す ϵ-NFA の ϵ-遷移を行うたびに，つまり系列に含まれるマッチの数が増えるたびに次の階層に移る．例えば時刻 3 の時点では，階層 1 の初期状態に $\langle ._1, ._2, ._3\rangle$ が，階層 2 の初期状態に $\langle \mathtt{w}_1, \mathtt{w}_2, ._3\rangle$ が保持される．マッチの数に応じて系列を保持しているため，時刻 7 の階層 1 で s_2，階層 2 で s_5 というように，k の値に応じた最適解が検出できる．

具体的に，図 1 の確率的データストリーム，$p = \langle \mathtt{w}^+ \rangle$，$k = 2$ を入力した例を用いて説明する．まず，図 6 に時刻 2 までに

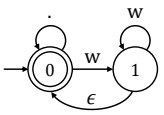

図 4 $\langle(. \vee \mathtt{w}^+)^*\rangle$ に対応する ϵ-NFA

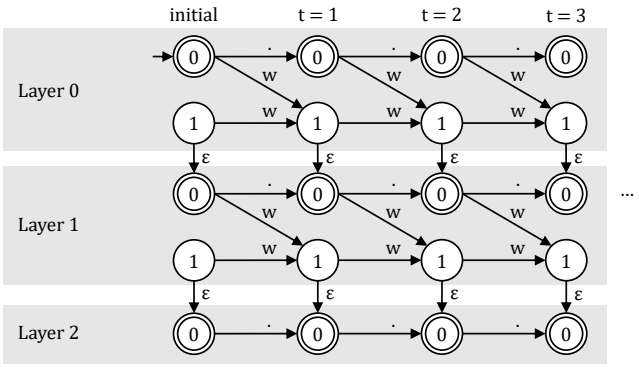

図 5 マッチ数で階層分けした状態遷移図

t	layer	state	
		0	1
initial	0	$\{\langle\rangle\}$	$\emptyset$
	1	$\emptyset$	$\emptyset$
	2	$\emptyset$	
1	0	$\{\underline{\langle ._1\rangle}\}$	$\{\underline{\langle \mathtt{w}_1\rangle}\}$
	1	$\{\underline{\langle [\mathtt{w}_1]\rangle}\}$	$\emptyset$
	2	$\emptyset$	
2	0	$\{\underline{\langle ._1, ._2\rangle}\}$	$\{\langle ._1, \mathtt{w}_2\rangle, \underline{\langle \mathtt{w}_1, \mathtt{w}_2\rangle}\}$
	1	$\{\langle ._1, [\mathtt{w}_2]\rangle, \underline{\langle [\mathtt{w}_1, \mathtt{w}_2]\rangle}, \langle [\mathtt{w}_1], ._2\rangle\}$	$\{\underline{\langle [\mathtt{w}_1], \mathtt{w}_2\rangle}\}$
	2	$\{\underline{\langle [\mathtt{w}_1], [\mathtt{w}_2]\rangle}\}$	

図 6 系列の生成過程

各状態に到達した系列を示す．なお，下線は各状態で情報利得が最大の系列を示し，各系列内では検出されたマッチを大括弧（[ ]）により区別している．処理の開始時点では階層 0 の初期状態のみが空系列を持つよう初期化する．時刻 1 では，初期状態からの遷移及び ϵ-遷移で到達できる状態に系列を拡張する．この例あれば，階層 0 の初期状態及び状態 1，階層 1 の初期状態に系列が到達する．この時点では各状態に唯一の系列しか到達しないため，各状態でその系列が保持される．時刻 2 でも同様に，時刻 1 に保持された系列から到達可能な状態に系列を拡張する．階層 0 の状態 1 及び階層 1 の初期状態に複数の系列が到達するため，それぞれで情報利得が最大となる系列 $\langle \mathtt{w}_1, \mathtt{w}_2\rangle$ 及び $\langle [\mathtt{w}_1, \mathtt{w}_2]\rangle$ のみを保持する．以降，最終時刻である時刻 7 まで同様の処理を続ける．そして，各階層の最終状態に到達した系列の中で最も情報利得が大きい系列を選択し，その中に含まれるマッチ集合を解として出力する．この例では，階層 2 の初期状態に到達する系列 $\langle [\mathtt{w}_1, \mathtt{w}_2], ._3, [\mathtt{w}_4, \mathtt{w}_5, \mathtt{w}_6, \mathtt{w}_7]\rangle$ の情報利得が最大であるため，マッチ集合 $\{\langle \mathtt{w}_1, \mathtt{w}_2\rangle, \langle \mathtt{w}_4, \mathtt{w}_5, \mathtt{w}_6, \mathtt{w}_7\rangle\}$ を解として出力する．

5.2 計 算 量

提案アルゴリズムの時間・空間計算量について順に述べる．

5.2.1 時間計算量

まず，提案アルゴリズムの時間計算量について述べる．上述したアルゴリズムでは系列の拡張と最尤系列の選択を繰り返し行う．最尤系列の選択は系列の生成と同時並行で行えるため，このアルゴリズムの計算量は系列をいくつ生成するかに依存する．なお計算のために，これ以降確率的データストリームに含まれるイベントの数を n で，パターンの長さ，つまりパターンに含まれるイベントシンボルの数を l で表す．

生成される系列数について，まず各タイムステップで生成される数を考え，次に全体で生成される数を考える．1 タイムステップ前に保持された系列を状態遷移図に従って伸ばすことで系列は拡張されるため，状態遷移図の各タイムステップにおける遷移数がそのまま系列の生成数となる．前節のステップ (1) でパターンから ϵ-NFA を生成するとき，その状態数及び遷移数は $O(l)$ である [19]．次にステップ (2) で DFA に変換するとき，DFA の各状態は NFA の状態の組合せで表されるため，状態数及び遷移数は $O(2^l)$ となる [19]．ステップ (3) と (4) で追加される遷移の数は定数で表せるため，状態遷移図の元となる ϵ-NFA の遷移数は変わらず $O(2^l)$ である．状態遷移図はこの ϵ-NFA を k 階層に拡張したものであるため，その遷移数，つまり各タイムステップにおける系列の生成数は $O(k2^l)$ となる．最後に，系列の生成はイベント数 n と同じ回数繰り返されるため，提案アルゴリズムの時間計算量は以下の式で表せる．

$$O(nk2^l) \tag{10}$$

つまり，DFA の状態数爆発問題 [19] により計算量が大きく増える可能性がある．

ただし現実的には，XML 文書などに対する照合とは違い，本稿で想定する行動認識などでそのような最悪の計算量をとることは少ないと考えられる．状態数が爆発する主な正規表現は，クリーネ閉包が付与されたイベントと付与されていないイベントを複数個組み合わせたものである．つまり，あるイベントの任意回数の生起と定数回の生起を厳密に区別しなければならないとき，状態数は爆発する．しかし，行動認識などにおいてイベントの生起回数は未知であるため，基本的にクリーネ閉包の付与されたイベントのみが使用されると考えられる．したがって，定数回の生起と区別する必要のあるケースも少なく，実際に状態数が爆発するのは稀だと考えられる．

また，状態数が爆発する場合も，アルゴリズムの実装を工夫することで対策がたてられる．一つの対策として，代表的な DFA 型エンジンである GNU Grep [26] でも用いられている遅延評価（lazy evaluation）について述べる．遅延評価は必要となるまで実際の値などを計算しないという戦略で，提案手法においてはいずれかの経路が到達するまで状態遷移図の状態及び遷移を計算しないことに相当する．つまり，DFA の状態数が非常に大きいとしても，その大部分は実際の処理では使用されないという点を利用する．遅延評価を用いるとき生成される状態数はデータストリームのイベント数に比例するため，状態数が爆発するようなパターンであっても，実行時には状態数及び遷移数を抑制できる．

5.2.2 空間計算量

空間計算量について述べる．提案手法は各タイムステップで生成された系列を保持する．系列の大きさは照合が終わった時点で最も大きくなるため，照合終了時の系列の数及び大きさから空間計算量を求める．提案手法で保持される系列数は状態遷移図の状態数と等しいため，時間計算量の場合と同様に $O(k2^l)$ となる．一方，照合終了時の系列の大きさは確率的データストリームの大きさと同じになるため，$O(n)$ である．つまり，空間計算量も式 (10) で表せる．ただし，時間計算量と同様に，状態数の爆発による計算量の増加は稀だと考えられる．

また，実際の空間計算量は更に小さくなることが予想される．式 (10) の計算量は全ての系列が別々に保持されることを想定しているが，実際には系列の多くは状態遷移図において同じ経路を共有する．したがって，経路が共有する部分で系列も同様に要素を共有することで，空間計算量を削減できる．

6. 評価実験

実データを用いて検出性能を，人工データを用いて処理速度を評価する．実データには Lahar プロジェクト [11] で公開されている屋内位置の確率的データストリームを使用する．実データは図 7 のように屋内構造をグラフで表しており，被験者が各ノードにいる確率と実際にその被験者がいた正解ノードの情報を一秒毎に持つ．つまり，パターン p を与えたとき，正解ノードの情報を用いることで正解となるマッチも検出できる．人工データには図 8 に示すマルコフモデルを用いてイベント数 10 万の確率的データストリームを生成した．生成は初期状態から始まり，各遷移確率に応じて状態を遷移する．各状態では五つのイベントシンボルを持つ確率的イベントを出力する．なお，イベントシンボルとしてアルファベット $\Sigma = \{a, b, ..., z\}$ を使用したため，シンボルの総数は 26 個である．

6.1 検出性能の評価

6.1.1 評価指標

適合率・再現率・F 値を用いて検出性能を評価する．ただし，確率的なパターン照合において検出結果と正解データが完全

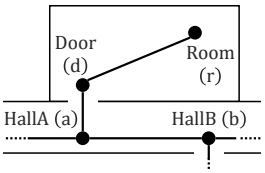

図 7　実データが持つグラフ構造の一部

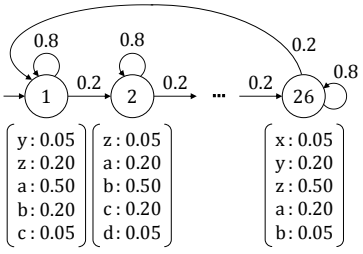

図 8　人工データ生成のためのマルコフモデル

一致することは稀であるため，二つのマッチがどれだけオーバラップするかを示す重複度を基に各指標を定める．

$$\begin{aligned}&\mathrm{overlap}(m,c)\\&=\frac{\min(m.t_e,c.t_e)-\max(m.t_s,c.t_s)+1}{\max(m.t_e,c.t_e)-\min(m.t_s,c.t_s)+1}\end{aligned}\tag{11}$$

適合率は検出の正確性，つまり検出したマッチが正解データとどれだけ一致するかを表す．本稿では，適合率を正解データに対するマッチの重複度の平均として定義する．

$$\mathrm{precision}(M,C)=\frac{1}{|M|}\sum_{m\in M}\max_{c\in C}(\mathrm{overlap}(m,c))\tag{12}$$

再現率は検出された正解データの割合であり，検出の網羅性を表す．適合率と同様に，再現率はマッチに対する正解データの重複度の平均として定める．

$$\mathrm{recall}(M,C)=\frac{1}{|C|}\sum_{c\in C}\max_{m\in M}(\mathrm{overlap}(m,c))\tag{13}$$

F 値は適合率と再現率の調和平均であり，検出の全体的な性能を表す．本稿では適合率と再現率を一対一の割合で評価し，F 値を以下の式で計算する．

$$\mathrm{F}(M,C)=\frac{2\cdot\mathrm{precision}(M,C)\cdot\mathrm{recall}(M,C)}{\mathrm{precision}(M,C)+\mathrm{recall}(M,C)}\tag{14}$$

6.1.2 生起確率を用いる場合との比較

比較手法として，マッチの生起確率に基づく Top-k 問合せを用いる．つまり，式 (9) において以下の式を目的関数とした場合と比較する．

$$\text{maximize}\quad\sum_{m\in M}P(m)\tag{15}$$

なお，情報利得を用いる場合と同様に，生起確率に基づく Top-k 問合せの解も提案手法を拡張することで求められる．

提案手法のように情報利得を用いることで，生起確率を用いる場合に比べて適合率・再現率・F 値を向上できる．実データに対して $p=\langle \mathtt{Door}^+\ \mathtt{Room}^+\ \mathtt{Door}^+\rangle$ を照合し，k の値を 1 から 30 まで変化させた際の結果を図 9，図 10，図 11 に示す．図からわかるとおり，生起確率を用いる場合いずれの指標でも 0.1 未満の値しか得られていない．これは，1. で述べたとおり，長い時間に渡って生起するマッチの生起確率が非常に小さいためである．各正解マッチは 20 から 40 秒ほどの時区間にわたって生起しており，その生起確率は 1.0×10^{-3} にも満たない．そのため，単純に生起確率の大きいマッチを出力するだけでは，正解マッチは検出できない．一方で，情報利得を用いる場合，マッチの生起確率が同時区間の系列の生起確率に比べて大きいかどうかに注目する．つまり，直感的に言えば，図 7 の “Door” や “Room” にいる確率が “HallA” や “HallB” などにいる確率よりも大きい時区間でマッチを検出する．機械学習による分類の時点で大きな誤りがない限り，そのような時区間に存在するマッチは当然正解マッチとも大きくオーバラップする．そのため，提案手法では全ての評価指標において生起確率を大きく上回る結果を残している．

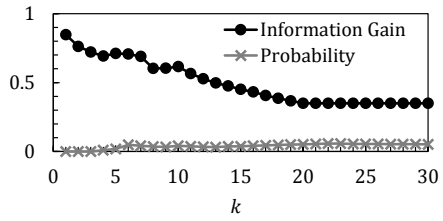

図 9　k の変化に対する適合率

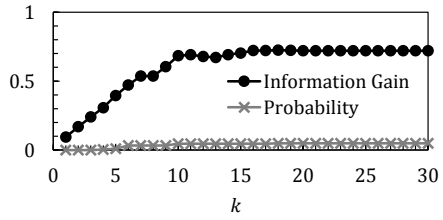

図 10　k の変化に対する再現率

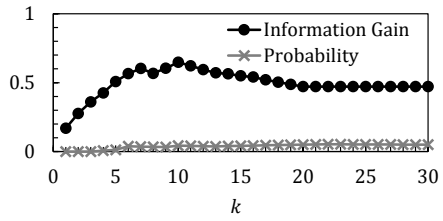

図 11　k の変化に対する F 値

また，提案手法を用いることで，適切なマッチを無駄なく検出できる．図 9 に示すとおり，適合率は k の値を大きくするほど小さくなる．適合率は検出の正確性を表す指標であるため，k の値の増加により適合率が減少することは，提案手法が適切なマッチを優先的に出力できていることを示す．なお各評価指標は $k=20$ 以降変化していないが，これは $k=20$ の時点で情報利得の総和が最大化し，それ以降同じマッチ集合が出力されているためである．一方で図 10 に示すとおり，再現率は k の値を大きくするほど大きくなり，その増加は $k=10$ 付近で止まっている．これは $p=\langle \mathtt{Door}^+\ \mathtt{Room}^+\ \mathtt{Door}^+\rangle$ に対する正解マッチの数が 9 個であるためであり，提案手法が正解に対応するマッチを無駄なく検出できていることを示す．また，図 11 に示すとおり，正解マッチの数である $k=9$ 前後で F 値が最大になっていることからも提案手法の有効性は確認できる．

6.2 処理速度の評価

入力となる確率的データストリームのイベント数 n，マッチの出力数 k，パターンの長さ l を変化させた際の実行時間を計測し，提案手法の時間計算量を確認する．実験では，確率的データストリームとして人工データを使用し，$\langle \mathtt{a}^+\rangle$，$\langle \mathtt{a}^+\ \mathtt{b}^+\rangle$ のようにクリーネ閉包を付与したイベントを一つずつ増やすことでパターンの長さ l を変化させた．なお，このとき状態遷移図の状態・遷移数はパターンの長さ l に対して線形に増加する．デフォルトの入力には $n=100{,}000$，$k=10$，$l=3$（$p=\langle \mathtt{a}^+\ \mathtt{b}^+\ \mathtt{c}^+\rangle$）を設定した．

実験結果から，提案手法の効率性が確認できた．図 12, 13, 14 にそれぞれ n，k，p を変化させた際の実行時間を示す．結果からわかるとおり，全ての場合で実行時間の増加は線形である．

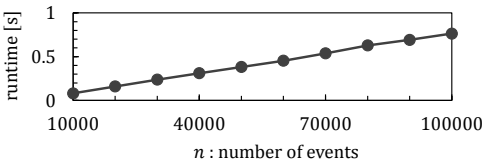

図 12　イベント数 n を変化させた際の実行時間

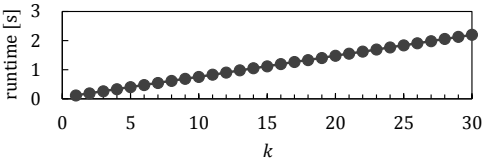

図 13　k を変化させた際の実行時間

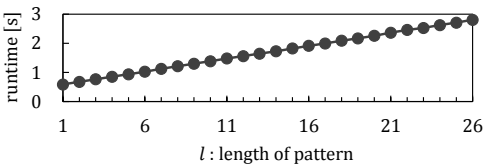

図 14　パターンの長さ l を変化させた際の実行時間

式 (10) で示したとおり，n 及び k に対する実行時間の増加は線形であり，実験結果でもそれが確認できた．また，今回の実験では上述のとおりパターンの変化に対して遷移数が線形に増加するため，l の変化に対する実行時間の増加も線形であった．

7. おわりに

確率的データストリーム上でのパターン照合を対象とし，マッチの新たな評価指標として，相対的な生起確率の大きさを表す情報利得を提案した．マッチの情報利得を定め，no-overlap セマンティクスに基づく Top-k パターン照合の問題定義を定めた．また，マッチの個数で階層分けした状態遷移図を用いることで，動的計画法により効率的に解が求められることを示した．実データを用いた実験により提案した情報利得が既存の評価指標よりも優れていることを示し，人工データを用いた実験によって実行時間においても優れていることを示した．

今後の課題としては，提案手法のリアルタイム処理への拡張，複数問合せの同時処理が挙げられる．本稿では与えられたデータストリームの全区間に対して処理を行ったが，過去 t 時刻におけるパターン照合をリアルタイムに行いたいという要求も存在する．そのため，既存研究でよく用いられる滑り窓などを適用し，本手法をリアルタイム処理に拡張することが考えられる．また，複数のユーザから同時に問合せが来ることも想定すると，複数問合せの同時処理も重要となる，したがって，YFilter [25] などのように，複数問合せの状態遷移図を共有することで処理を効率化することも必要となってくる．

謝辞　本研究は，JST COI（Center of Innovation）プログラムおよび科研費（16H01722, 26540043）による．

文　献

[1] J. Yin, Q. Yang, and J.J. Pan, “Sensor-based abnormal human-activity detection,” IEEE TKDE, vol.20, no.8, pp.1082–1090, 2008.
[2] L. Chen, C. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” IEEE TKDE, vol.24, no.6, pp.961–974, 2012.
[3] O.D. Lara and M.A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Commun. Surveys Tutorials, vol.15, no.3, pp.1192–1209, 2013.
[4] G. Cugola and A. Margara, “Processing flows of information: From data stream to complex event processing,” ACM Comput. Surv., vol.44, no.3, pp.15:1–15:62, 2012.
[5] T.T.L. Tran, L. Peng, Y. Diao, A. McGregor, and A. Liu, “CLARO: Modeling and processing uncertain data streams,” VLDB J., vol.21, no.5, pp.651–676, 2012.
[6] G. Cormode and M. Garofalakis, “Sketching probabilistic data streams,” Proc. 2007 ACM SIGMOD, pp.281–292, 2007.
[7] C. Jin, K. Yi, L. Chen, J.X. Yu, and X. Lin, “Sliding-window top-k queries on uncertain streams,” Proc. VLDB Endow., vol.1, no.1, pp.301–312, 2008.
[8] Q. Zhang, F. Li, and K. Yi, “Finding frequent items in probabilistic data,” Proc. 2008 ACM SIGMOD, pp.819–832, 2008.
[9] C.C. Aggarwal and P.S. Yu, “A framework for clustering uncertain data streams,” 2008 IEEE 24th ICDE, pp.150–159, 2008.
[10] Z. Li, T. Ge, and C.X. Chen, “ε-matching: Event processing over noisy sequences in real time,” Proc. 2013 ACM SIGMOD, pp.601–612, 2013.
[11] C. Ré, J. Letchner, M. Balazinska, and D. Suciu, “Event queries on correlated probabilistic streams,” Proc. 2008 ACM SIGMOD, pp.715–728, 2008.
[12] E. Wu, Y. Diao, and S. Rizvi, “High-performance complex event processing over streams,” Proc. 2006 ACM SIGMOD, pp.407–418, 2006.
[13] H. Zhang, Y. Diao, and N. Immerman, “On complexity and optimization of expensive queries in complex event processing,” Proc. 2014 ACM SIGMOD, pp.217–228, 2014.
[14] M. Liu, D. Golovnya, E.A. Rundensteiner, and K.T. Claypool, “Sequence pattern query processing over out-of-order event streams,” 2009 IEEE 25th ICDE, pp.784–795, 2009.
[15] B. Chandramouli, J. Goldstein, and D. Maier, “High-performance dynamic pattern matching over disordered streams,” Proc. VLDB Endow., vol.3, no.1-2, pp.220–231, 2010.
[16] M. Akdere, U. Çetintemel, and N. Tatbul, “Plan-based complex event detection across distributed sources,” Proc. VLDB Endow., vol.1, no.1, pp.66–77, 2008.
[17] Y. Mei and S. Madden, “ZStream: A cost-based query processor for adaptively detecting composite events,” Proc. 2009 ACM SIGMOD, pp.193–206, 2009.
[18] L. Woods, J. Teubner, and G. Alonso, “Complex event detection at wire speed with FPGAs,” Proc. VLDB Endow., vol.3, no.1-2, pp.660–669, 2010.
[19] J.E. Hopcroft, R. Motwani, and J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison Wesley, 2000.
[20] D.E. Knuth, J.H. Morris, Jr., and V.R. Pratt, “Fast pattern matching in strings,” SIAM J. Comput., vol.6, no.2, pp.323–350, 1977.
[21] A.V. Aho and M.J. Corasick, “Efficient string matching: An aid to bibliographic search,” Commun. ACM, vol.18, no.6, pp.333–340, 1975.
[22] I. Nakata, “Generation of pattern-matching algorithms by extended regular expressions,” Japan Soc. Softw. Sci. Tech., vol.5, pp.1–9, 1993.
[23] S. Santini, “Querying streams using regular expressions: Some semantics, decidability, and efficiency issues,” VLDB J., vol.24, no.6, pp.801–821, 2015.
[24] G.D. Forney, Jr., “The Viterbi algorithm,” Proc. IEEE, vol.61, no.3, pp.268–278, 1973.
[25] Y. Diao, P. Fischer, M.J. Franklin, and R. To, “YFilter: Efficient and scalable filtering of XML documents,” Proc. 18th ICDE, pp.341–342, 2002.
[26] Grep - GNU Project - Free Software Foundation: `https://www.gnu.org/software/grep/`.