2B-3

ストリーム上の頻出時系列とその近似発見アルゴリズムについて

岡本 敦† 正代 隆義‡

九州大学システム情報科学府情報学専攻† 九州大学システム情報科学研究院情報学部門‡

1. はじめに

本論文ではストリームデータの頻出時系列パタンを列挙する問題を考える．Karp ら[2]は，ストリームデータ中で出現割合があらかじめ定められたγ以上の頻出イベントを出力するストリームアルゴリズムを提案した．Manku と Motwani[3]は，頻出データの誤り具合の基準であるε-劣シノプス(ε-deficient synopsis)を定義し，誤り許容カウント法とよばれる精度保証付き頻出アイテム発見アルゴリズムを提案した．

ストリームデータの頻出時系列パタンを列挙するとき，時系列パタン出現数をどのようにカウントするかが重要な鍵となる．例えば，インターネットログデータに現れる頻出アクセスパタンの発見では，ひとつのパタンの出現数はそのパタンの 2 つの出現が重複しないようカウントするほうが良い．本論文では，先入れ先出しによる頻出度カウント(First-In First-Count)のもとで，ε-劣シノプスを保持する定数長時系列パタン発見ストリームアルゴリズムを提案する．さらに，実データ上で提案アルゴリズムの有効性を示す．

2. 時系列パタンと先入れ先出しカウント

E を有限個のイベントの集合とする．ストリームデータをイベントの有限列 $S=(a_1,a_2,...,a_N)$ $(a_i\in E, 1\le i\le N)$と定義する．自然数 k に対して，k 個のイベント $e_1,e_2,\ldots,e_k$ と k-1 個の自然数 $T_1,T_2,\ldots,T_{k-1}$ の交互列$\pi=(e_1,T_1,e_2,T_2,\ldots,T_{k-1},e_k)$を k-時系列パタンとよぶ．特に，全ての $T_1,T_2,\ldots,T_{k-1}$ がある自然数 T に等しいとき，πを (k,T) -時系列パタンとよぶ．$o=(i(1),i(2),\ldots,i(k))$を自然数の真の増加列とする．ストリームデータ $S=(a_1,a_2,\ldots,a_N)$と k-時系列パタン$\pi=(e_1,T_1,e_2,T_2,\ldots,T_{k-1},e_k)$に対し，$o$ が次の条件を満たすとき，o を S におけるπの出現とよぶ: 全てのτ $(\tau=1,\ldots,k)$に対して，$a_{i(\tau)}=e_\tau$かつ $i(\tau+1)-i(\tau)\le T_\tau$ である．$o=(i(1),\ldots,i(k))$と $o'=(i'(1),\ldots,i'(k))$を S におけるπの出現とする．o と o'が重複する出現であるとは，o と o'に同じ自然数が現れるときをいう．このとき，$a_{i(\tau)}$と $a_{i'(\tau')}$は同じイベントである．出現 o と o'が重複していなければ，o と o'は独立しているという．

定義 1. S をストリームデータとし，πを k-時系列パタンとする．$\mathrm{Occ}_S(\pi)$で S におけるπの出現全体を表す．$\mathrm{Occ}_S(\pi)$の部分集合 O が互いに独立な出現のみからなるとき，O を独立出現集合とよぶ．独立出現集合 O が極大であるとは，O に属さない任意の出現 o に対して，O に属す出現 o'で o と重複するものが存在するときをいう．

例 1. $S=(a,a,a,b,b,b,b,b,b,b)$, $\pi=(a,3,b,3,b)$とするとき，$O_1=\{(1,4,7),(2,5,8),(3,6,9)\}$,$O_2=\{(1,4,5),(3,6,7)\}$,$O_3=\{(3,4,5)\}$はいずれも極大独立出現集合である．

定義 2. Sにおけるπの 2 つの出現 $o=(i(1),i(2),\ldots,i(k))$と $o'=(i'(1),i'(2),\ldots,i'(k))$に対して，次が成り立つとき $o<o'$と書く: 自然数τ $(1\le\tau\le k)$が存在して，τ未満の自然数 r $(1\le r<\tau)$に対して，$i(r)=i'(r)$かつ $i(\tau)<i'(\tau)$である．$o<o'$または $o=o'$であるとき，$o\le o'$と書く．O を$\mathrm{Occ}_S(\pi)$の部分集合とする．$\min_{\le O}$ で≤に関して最小な O の出現を表す．S におけるπの先入れ先出しカウント極大独立出現集合(FIFC 極大独立出現集合)を入力 $\mathrm{Occ}_S(\pi)$に対する手続き FIFC-MOS (図 1)の出力と定める．Sにおけるπの FIFC 極大独立出現集合を $\mathrm{Occ}^{\mathrm{FIFC}}{}_S(\pi)$と書く．例 2 の S とπでは，O_2 が $\mathrm{Occ}^{\mathrm{FIFC}}{}_S(\pi)$である．

```
手続き FIFC-MOS(O: 出現集合);
begin
    O':=φ;
    while O≠φ do begin
        o:=min≤ O; O':=O'∪{o};
        O:=O-({o}∪{o'∈O | o と o'は重複する})
    end;
    output O'
end.
```

図 1: $\mathrm{Occ}^{\mathrm{FIFC}}{}_S(\pi)$は入力 $\mathrm{Occ}_S(\pi)$に対する出力．

3. FIFC 頻出時系列パタン発見アルゴリズム

長さ N のストリームデータ S と k-時系列パタンπに対して，$\mathrm{Supp}^{\mathrm{FIFC}}{}_S(\pi)=|\mathrm{Occ}^{\mathrm{FIFC}}{}_S(\pi)|/N$ と定め，FIFC 頻出度とよぶ．例えば，例 1 の S とπに対して，$\mathrm{Supp}^{\mathrm{FIFC}}{}_S(\pi)=2/9$ となる．γを正の実数$(0<\gamma\le 1)$とする．k-時系列パタンπが S に関してγ-FIFC 頻出であるとは，$\mathrm{Supp}^{\mathrm{FIFC}}{}_S(\pi)\ge\gamma$であるときをいう．次の計算問題を考える．

Finding Approximate Frequent Time Series Sequence Patterns in Streaming Data
†Atsushi Okamoto, ‡Takayoshi Shoudai
†‡Department of Informatics, Kyushu University, Japan

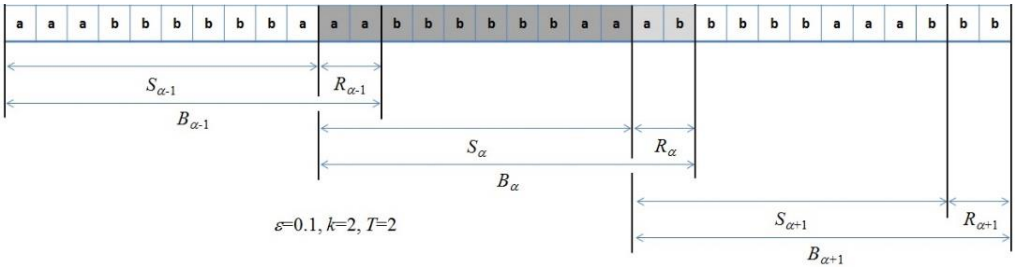

図 3: ストリームデータのバケットへの分割

アルゴリズム ROUND_ROBIN-FIFC-MOS(*S*, γ, ε);
begin
 D:=φ;
 for α:=1 to⌈ε*N*⌉ do begin
  for β:=1 to 1/ε do begin
   $W_{\alpha,\beta}$:=*S*[(α-1)/ε+β, (α-1)/ε+β+(*k*-1)*T*];
   foreach *o*=(*i*(1),…,*i* (*k*)) s.t.
    i(1)=(α-1)/ε+β<…<*i*(*k*)≤(α-1)/ε+β+(*k*-1)*T* を
    定義 2 の順序"≤"で小さい順に列挙 do begin
     π=($a_{i(1)}$,*T*,…,*T*,$a_{i\,(k)}$);
     if *used*(π,*i*(τ))=*false* for ∀τ (1≤τ≤*k*) then begin
      if ∃(π,*f*,δ)∈*D* s.t. *o* はπの出現 then *f*:=*f*+1
      else *D*:=*D*∪{(π,1,α)};
      forall τ (1≤τ≤*k*) do *used*(π,*i*(τ)):=*true*
     end
    end
   end;
   forall (π,*f*,δ)∈*D* s.t. *f*≤α-δ do *D*:=*D*-{(π,*f*,δ)}
  end;
  output all π s.t. ∃(π,*f*,δ)∈*D* かつ *f*≥(γ-ε)*N*
end.

図 2: FIFC 頻出誤り許容カウント法

FIFC 頻出 (*k*,*T*)-時系列パタン発見問題
入力: ストリームデータ *S* と正の実数γ (0<γ≤1).
問題: *S* に関してγ-FIFC 頻出である(*k*,*T*)-時系列パタンを全て列挙せよ.

総当たり法による誤り許容法を図 2 に示す．εは誤り許容値(0<ε≤1)である．*used*(π,*i*)は時系列パタンπと自然数 *i* に対して *false* に初期化されているとする．*S*=(a_1,a_2,...,a_N)を長さ 1/ε+(*k*-1)*T* のバケットに分割する．具体的には，自然数α (1≤α≤⌈ε*N*⌉) に対して，S_α=*S*[(α-1)/ε+1,α/ε], R_α=*S*[α/ε+1,α/ε+(*k*-1)*T*]とし，α番目のバケットを B_α=S_α·R_αと定める(図 3)．ここで，*i*≤*j* に対して，*S*[*i*,*j*]=(a_i,…,a_j)を表す．"·"はイベント列の連結を表す．アルゴリズムは，3 つ組(π,*f*,δ)をシノプスとよばれるデータ構造*D*に蓄えながらバケットを一つずつ処理する．

定理 1. アルゴリズム ROUND_ROBIN-FIFC-MOS は，ε-劣シノプスを保持する全てのγ-FIFC 頻出(*k*,*T*)-時系列パタンを出力する．

4. 評価実験と今後の課題

頻出アイテム発見ではアプリオリ法が有名である．FIFC 頻出度の場合，任意の *k*≥4 に対して，逆単調性がないことから，アプリオリ法の適用は難しい．例えば，*S*=(*a*,*b*,*a*,*a*,*b*,*a*,*b*,*b*,*a*,*a*,*b*,*b*)をストリームデータとし，π=(*a*,3,*b*,3,*a*,3,*b*)を 4-時系列パタンとする．このとき，$|\mathrm{Occ}^{\mathrm{FIFC}}{}_S(\pi)|$=3 であるが，πの最初の 3 イベントからなる部分パタンπ'=(*a*,3,*b*,3,*a*)に対して，$|\mathrm{Occ}^{\mathrm{FIFC}}{}_S(\pi')|$=2 である．一方，*k*≤3 では逆単調性が成り立つ．そこで，以下では *k*=3 のアプリオリ法によるアルゴリズム APRIORI-FIFC-MOS の実験結果も報告する．

評価実験用ストリームデータは某サイトの DNS ログデータである．実験環境は Xeon E5620 (2.4GHz,12MB キャッシュ)×2，メモリ 24GB で，実装は Red Hat Linux 上の GCC 4.4.6 を用いた．以降，*k*=3 である．アルゴリズム ROUND_ROBIN-FIFC-MOS は，*N*=3,000, ε=0.01 に対して，計算時間 5,268(s)であった．一方，APRIORI-FIFC-MOS では 1(s)未満である．*N*=2,000,000 (記録時間 8 時間 48 分) のデータでは，γ=0.0015, ε=0.0005 に対して，ROUND_ROBIN-FIFC-MOS はその処理を現実的な時間で終えることが出来なかった．一方 APRIORI-FIFC-MOS では，計算時間 6 時間 17 分で，11 個の 3-時系列パタンを出力した．

k≥4 で精度が保証された解を得るためには，現状では総当たり法しかない．今後の課題は，実時間処理が可能な精度保証付き頻出時系列パタン発見ストリームアルゴリズムの開発である．

謝辞
本研究の一部は国際連携によるサイバー攻撃の予知技術の研究開発 (総務省) の支援を受けたものである．

参考文献

1. R. M. Karp, C. H. Papadimitriou, and S. Shenker, "Simple Algorithm for Finding Frequent Elements in Streams and Bags", ACM Trans. Database Systems 28(1), pp.51-55, 2003.
2. G. S. Manku and R. Motwani, "Approximate Frequency Counts over Data Streams", Proc. VLDB 2002, pages 346-357, 2002.