半順序ストリームデータのサマリ構築

ϵ-Approximate Summary of Streaming Data with Partial-Order

山本 泰生 [1*]　岩沼 宏治 [1]　今井 友輝 [2]
Yoshitaka Yamamoto[1]　Koji Iwanuma[1]　Yuki Imai[2]

[1] 山梨大学大学院総合研究部
[1] Department of Computer Science, University of Yamanashi
[2] 株式会社コンピュータマインド
[2] Computermind Corporation

Abstract: In this paper, we propose an online algorithm to construct an ϵ-*approximate summary* of streaming data with partial-order. The proposed method has higher versatility that enables us to apply for various typical problems on streaming data. We focus on the problem of itemset mining or sequential pattern mining and show how they can be solved using the proposed method.

1　はじめに

クラウドサービスの普及と IoT の発展に伴い，多岐にわたる分野において，多種多様な**ストリームデータ** (*Streaming data*, SD と略す) が出現している．一般に SD は，観測系から生成され続ける無限長のデータ列と定義される．時間経過とともに蓄積されるデータ総量が急速に増加する点から，SD を扱う計算タスクでは，ボトルネックとなるデータ全体のスキャンを極力避けながら，新規データを逐次的にインメモリ処理することが求められる．このような“On-the-fly”なオンライン処理はビッグデータの管理・運用技術における重要な研究課題と位置づけられている [9].

この背景のなか，特に SD を扱うデータマイニング分野では，データの要約表現を求めるオンラインアルゴリズム [1, 10, 2] が活発に研究されている．さて，データの要約表現といったとき，平均，分散等の統計量や機械学習における生成モデルを指すことがあるが，ダイナミックな概念遷移を扱う SD では，データの出現分布を特徴づけるノンパラメトリックな要約表現を対象とすることが多い．その代表例として，任意の ϕ $(0 \leq \phi \leq 1)$ に対する ϕ-分位数を与える **ϕ-分位数サマリ** (ϕ-*quantile summary*, QS と略す) を求める問題がある．

実数 $\mathcal{R}$ 上の n 個の要素からなるストリームデータを考える．このとき昇順で $\lceil \phi n \rceil$ 番目 (ランクと呼ばれる) の要素を ϕ-分位数と呼ぶ．QS は任意の ϕ-分位数を与える要約表現であり，ヒストグラム生成やノンパラメトリック検定の要素技術として幅広く利用されている．誤差を許容する ϵ-近似アルゴリズムの枠組みでこれまで様々な手法が提案されており，文献 [10] ではそれらの包括的な比較検証がなされている．

さて本研究では，SD の要約表現を求める問題を，半順序関係を用いて形式化することを考える．例えば，QS 問題は，全順序集合 $\langle \mathcal{E}, \leq \rangle$ 上のストリームデータの要約表現を求める問題 (ただし $\mathcal{E}$ は実数全体，$\leq$ は実数上の大小関係) とみなすことができる．興味深いことに，ストリームデータマイニングの代表的な緒問題が，対象とする順序関係の種類に応じて，この要約表現を求める問題に帰着することができる．

本稿では，QS 問題，“Heavy hitter” (HH と略す) 問題として知られる**アイテムマイニング**，**アイテム集合マイニング**，**アイテム系列マイニング**の各問題と SD の要約表現を求める問題との関係を明らかにする．次に，QS 問題を解く**マージ可能** (*mergeable*) でかつ省メモリな決定性アルゴリズムを提案する．最後に，この提案法を拡張することで**束** (*lattice*) 関係上の要約表現を求める汎用のオンライン近似アルゴリズムを提案する．

アイテムの集合や系列を扱うマイニングタスクの本質的困難さは，解候補の組み合わせ爆発現象にある．特にオンライン処理が要求される SD においては，これを効果的に抑制することがより重要となる．低頻度候補の削除 [7] や飽和性による可逆圧縮 [11]，また飽和性を緩和した非可逆圧縮法 [8] などの技法がこれまでに提案されている．他方，データの要約表現という観点からの検討はこれまで十分になされてこなかった．本稿では，提案法がアイテム集合マイニングとアイテム系列マイニングのそれぞれのタスクにどのように適用されるか概説し，その有用性について前者タスクに関する予備実験の結果を示す．

*山梨大学大学院総合研究部
〒 400-8510 山梨県甲府市武田 4-4-37 A3-K212
E-mail: yyamamoto@yamanashi.ac.jp

2 用語定義と問題設定

観測系から生成されうるイベントの全体集合を $\mathcal{E}$ とする．$\leq$ を $\mathcal{E}$ 上の順序関係とする．ストリームデータ V_n は，順序集合 $\langle \mathcal{E}, \leq \rangle$ 上のイベント列 $\langle e_1, \ldots, e_n \rangle$ と定義される．ただし n は出力要請のあった時刻 (通常は未知の値) であり，$e_i \in \mathcal{E}$ $(1 \leq i \leq n)$ である．$\leq$ が束のとき，イベント $e, e' \in \mathcal{E}$ の**上限** (*greatest specialization* もしくは *supremum*) を $gs(e, e')$ と表す．

V_i を時刻 i までのストリームデータとし，e を $\mathcal{E}$ 中のイベントとする．このとき，$|\{e_j \mid e_j \in V_i, e_j \leq e\}|$ を V_i における e の**サポート** (*support*) と呼び，$sup(e, V_i)$ と表す．本稿では，任意の時刻 i とイベント e に対して，ある誤差の範囲内で $sup(e, V_i)$ を復元できる要約表現を考える．以降，各イベントに対し，現時刻のサポート見積を与えるデータ構造を**サマリ** (*summary*) と呼ぶ．時刻 i のサマリを S_i とし，イベント e に対して S_i が与えるサポート見積を $c(e, S_i)$ と表す．

定義 1 (Δ-近似サマリ) S_i が V_i に対する **Δ-近似サマリ** *(Δ-approximate summary)* であるとは，ある非負整数 Δ が存在して，任意の $e \in \mathcal{E}$ に対して，

$$c(e, S_i) - \Delta \leq sup(e, V_i) \leq c(e, S_i) + \Delta$$

を満たすときまたそのときに限る．

ストリームデータの特徴から，データ全体を何度も走査することは原理的に困難である．本研究では，各時刻のイベント e_i から逐次的にサマリ S_i を更新する**1-パス型近似アルゴリズム**（*one-pass approximation-algorithm*）を扱う．近似アルゴリズムには，許容誤差率 ϵ $(0 \leq \epsilon \leq 1)$ を限定するパラメータ制約型とメモリ使用量 k $(k > 0)$ を限定するリソース制約型の 2 種類が存在する [12] が，本稿では前者のパラメータ制約型近似アルゴリズムに焦点をあてる．このとき，各時刻 i において，$\Delta \leq \epsilon \times i$ を満たすような Δ-近似サマリを求めることが計算タスクとなる．

3 関連研究

3.1 アイテムマイニング

各時刻 i においてアイテムが到着するストリームデータを対象とする．計算タスクは各時刻 i での任意のアイテムの頻度を許容誤差 $\epsilon \times i$ の範囲内で与えることである．決定性アルゴリズムでは Space-Saving 法 [7] が有名である．この手法では，アイテムと頻度計数の組のエントリを $\lceil \frac{1}{\epsilon} \rceil$ 個だけ利用し，各時刻 i に到着するアイテムに対し，(1) そのアイテムのエントリが存在すればその計数を 1 だけ増やす; (2) 存在せずかつ未使用のエントリがあればそのエントリにアイテムを登録する; (3) エントリが一杯であれば計数が最も小さいエントリのアイテムと交換しその計数を 1 だけ増やす，という更新手続きをとる．これにより，$\epsilon \times i$ より大きい頻度をもつ任意のアイテムとその頻度を許容誤差 $\epsilon \times i$ の範囲内で与えることができる．

例 1 $V_8 = \langle a, b, c, d, a, c, b, e \rangle$, $\epsilon = 0.25$ のとき，Space-Saving 法では時刻 8 において 4 つのエントリの集合 $S_8 = \{(2, a), (2, c), (2, b), (2, e)\}$ を管理する．

アイテムの全体集合を $\mathcal{I}$ とすると，Space-Saving 法が管理するエントリ集合 S_n は，順序集合 $\langle \mathcal{I}, = \rangle$ 上のストリームデータの ϵn-近似サマリとなっている (エントリに未登録アイテム e は，$c(e, S_n) = 0$ と考えればよい)．すなわち，等価関係上の ϵn-近似サマリは Space-Saving 法により求めることが可能である．

3.2 ϕ-分位数サマリ

ϵn だけ誤差を許容する ϕ-分位数サマリは，全順序集合 $\langle \mathcal{R}, \leq \rangle$ 上の ϵn-近似サマリと等価である．定義から，ストリームデータ V_n における各 e のサポート $sup(e, V_n)$ は，e の V_n 上の**ランク** (*rank*) に相当する．すなわち，V_n の ϕ-分位数とは，サポートが $\lceil \phi n \rceil$ であるイベントに他ならない[1]．分位数サマリは，任意のイベントに対し，ϵn の範囲内でそのランク (すなわちサポート) を復元できる．よって，これは ϵn-近似サマリである．

さて，分位数サマリの重要な性質として**マージ可能性** (*mergeability*) [1] がある．ストリームデータ $V^{(1)}$ (*resp.* $V^{(2)}$) に対して，$\Delta^{(1)}$ (*resp.* $\Delta^{(2)}$)-近似サマリ $S^{(1)}$ (*resp.* $S^{(2)}$) が与えられたとする．このとき 2 つのストリームデータの連結 $V^{(1)} \cdot V^{(2)}$ に対する $\Delta^{(1)} + \Delta^{(2)}$-近似サマリ S を，$S^{(1)}$ と $S^{(2)}$ から構成できるとき，S はマージ可能であるという．空間計算量が最良の決定性アルゴリズムとして GK 法 [4] が知られているが，このデータ構造 GK サマリはマージ可能ではない [1]．マージ可能な決定性アルゴリズムとしては，陽には示されていないものの，階層構造を利用したサマリ構築法 [13] がある．他方，この手法の空間計算量は $O(\frac{1}{\epsilon}\log^2 \epsilon n)$ であり，GK サマリに劣っている．本稿では，マージ可能でかつ GK 法と同様の空間計算量をもつ近似アルゴリズムを示す．

3.3 アイテム集合マイニング

各時刻 i において**トランザクション** (*transaction*) と呼ばれるアイテム集合 $t_i \subseteq \mathcal{I}$ が到着するストリームデータを対象とする．計算タスクは，ストリームデー

[1] 厳密には重複要素のランクをタイブレークする必要がある．

タ中の全頻出アイテム集合とそれらの頻度を誤差 ϵn の範囲内で与えることである．この問題は，順序集合 $\langle 2^{\mathcal{I}}, \supseteq \rangle$ 上の ϵn-近似サマリを求める問題に帰着できる ($\supseteq$ は集合間の包含関係を表す)．この帰着は，V_n における e のサポートが e の頻度と一致することから導かれる．また $\langle 2^{\mathcal{I}}, \supseteq \rangle$ は束である点に注意されたい．

本研究では，全順序関係に対するサマリ構築法を束関係上のものへ拡張する．結果として提案法は，飽和性を緩和したアイテム集合族の非可逆圧縮表現 [8] を求める新しい 1 パス型近似アルゴリズムとなっている．

3.4 アイテム系列マイニング

トランザクション t_i ではなく，アイテムの系列 $s_i = (a_i^{(1)}, \ldots, a_i^{(m_i)})$ が各時刻 i に到着するストリームデータを対象とする (m_i は s_i の系列幅であり，$a_i^{(j)} \in \mathcal{I}$ を満たす)．アイテム集合マイニングに比べるとより複雑に見えるが，この問題も束上のサマリ構築アルゴリズムを用いて解くことができる．ただし扱う順序集合はアイテムを頂点とする有向グラフ全体の集合となる．また有向グラフ間の関係 $\leq$ を次のように与える．

定義 2 $G_1 = (N_1, E_1)$ と $G_2 = (N_2, E_2)$ を有向グラフ (G_1 と G_2 は孤立頂点を持たない) とし，グラフ上で到達可能な経路の集合をそれぞれ P_{G_1} と P_{G_2} とする．このとき，$P_{G_1} \supsetneq P_{G_2}$ もしくは $P_{G_1} = P_{G_2}$ かつ $E_{G_1} \supseteq E_{G_2}$ を満たすとき $G_1 \leq G_2$ と表す．

定義から $\leq$ は順序関係である．また G_1，G_2 には上限が存在し，$gs(G_1, G_2)$ の辺集合は $P_{G_1} \cap P_{G_2}$ となる．よって $\leq$ は束になっている．

例 2 図 1 におけるグラフ G_1，G_2 を考える (それぞれ系列 $s_1 = (a, b, c, d)$，$s_2 = (b, a, d, c)$ に相当する)．$P_{G_1} \cap P_{G_2}$ を辺集合にもつ右端のグラフを G_3 とすると，$G_1 \leq G_3$ と $G_2 \leq G_3$ が成り立つ．また $G_3' < G_3$ でありかつ $G_1 \leq G_3'$，$G_2 \leq G_3'$ を満たす G_3' は存在しない．存在すると仮定すると，少なくとも $P(G_3') = P(G_3)$ である．しかし，$G_3' < G_3$ なので，G_3 に存在しない辺を G_3' は持つこととなり，$P(G_3') = P(G_3)$ に矛盾する．よって G_3 は G_1 と G_2 の上限となっている．

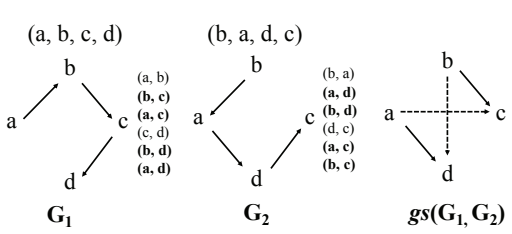

図 1: 2 つの有向グラフの上限

4 ϵn-近似サマリのオンライン計算法

4.1 全順序ストリームデータのサマリ構築

提案法の枠組みは先行研究 [13] と同じである．すなわちストリームデータ V_n をブロックに区切り，ブロック単位で**間引き** (*compress*) と**マージ** (*merge*) を適用し，階層構造のサマリを構築していく．許容誤差率を ϵ とし，以後，$k = \lceil \frac{1}{\epsilon} \rceil$ とおく (簡単のため k は偶数と仮定する)．V_n 上のある区間のイベント列 $b^l = \langle e_i, \ldots, e_{i+k\times 2^{l-1}-1} \rangle$ を **l-レベルブロック**と呼ぶ．簡単のため，V_n は 1-レベルブロックを用いて，$V_n = \langle b_1^1, \ldots, b_m^1 \rangle$ と書けるものとする (すなわち $n = k \times m$)．**l-レベルサマリ**はある l-レベルブロックの要約表現を表し，**エントリ**と呼ばれる組 (c, e) を高々$2k$ 個保持する．e はその l-レベルブロックに出現したイベントを表し，c は e のサポート見積を表す．**全体サマリ** S はいくつかの l_i-レベルサマリの集合 $\{s^{l_1}, \ldots, s^{l_u}\}$ である．ただし同じレベルのサマリは高々1 つしか含まない (すなわち $i \neq j$ ならば $l_i \neq l_j$ である)．

Algorithm 1 ϵn-近似サマリを求める提案法

Input: $V_n = \langle b_1^1, \ldots, b_m^1 \rangle$, k
Output Summary S of V_n

```
1:  i := 1
2:  initialize S
3:  while i ≤ m do read b_i^1
4:      s_i^1 := create(b_i^1)                    ▷ (1)(2)
5:      S := S ∪ {s_i^1}
6:      while there exist s_1^l, s_2^l ∈ S for some l do
7:          s^{l+1} := merge(s_1^l, s_2^l)        ▷ (3)
8:          drop(s^{l+1})                         ▷ (4)
9:          replace s_1^l, s_2^l with s^{l+1}
10:     end while
11:     i := i + 1
12: end while
```

提案法 (Algorithm 1) では，V_n 中の各 b_i^1 $(1 \leq i \leq m)$ に対し，(1) 全順序 $\leq$ を用いて昇順整列し，(2) 1-レベルサマリ s_i^1 を構築する．(1) で整列した配列を $\vec{b}_i^1$ と書くと，(2) では $\vec{b}_i^1$ 中の各 j 番目のイベント $e^{(j)}$ に対し，エントリ $(j, e^{(j)})$ を作成し s_i^1 に登録する．すべてのエントリを登録後，s_i^1 を S に追加する (5 行目)．もし S が同じレベルのサマリ s_1^l と s_2^l を含む場合，(3) マージと (4) 間引きを行い，s_1^l と s_2^l を $l+1$ レベルのサマリ s^{l+1} に置き換える．S に同じレベルのサマリが存在する限り，この処理を繰り返す．

(3) のマージ操作では，s_1^l と s_2^l に含まれる全エントリをイベント順に昇順整列する．その後，i 番目のエントリのサポート見積を $2^{l-1} \times i$ に更新する．(4) の間引

き操作では，s^{l+1} 内で昇順整列されている $2k$ 個のエントリのうち奇数番目のエントリをすべて削除する．

例 3 $\epsilon = 0.25$ とし $\langle \mathcal{R}, \leq \rangle$ 上の 6 つの 1-レベルブロックからなるストリームデータ V_{24} を考える:

$$b_1^1 = \langle 1.4, 2.5, 7.4, 11 \rangle,\ b_2^1 = \langle 8.9, 4.2, 11, 1.5 \rangle,$$
$$b_3^1 = \langle 6.1, 1.5, 2.7, 3.3 \rangle,\ b_4^1 = \langle 5.2, 6.9, 18, 12 \rangle,$$
$$b_5^1 = \langle 0.4, 4.9, 4.2, 4.1 \rangle,\ b_6^1 = \langle 2.4, 8.1, 12, 15 \rangle.$$

b_1^1，b_2^1 の 1-レベルサマリ s_1^1，s_2^1 を求めると，

$$s_1^1 = \{(1, 1.4), (2, 2.5), (3, 7.4), (4, 11)\},$$
$$s_2^1 = \{(1, 1.5), (2, 4.2), (3, 8.9), (4, 11)\}$$

となる．次にこれらにマージと間引き操作を適用して得られる $s_1^2 = \{(2, 1.5), (4, 4.2), (6, 8.9), (8, 11)\}$ が S に追加される．同様に b_3^1 と b_4^1 から 2-レベルサマリ $s_2^2 = \{(2, 2.7), (4, 5.2), (6, 6.9), (8, 18)\}$ が S に追加される．ここで S 中に同じレベルのサマリ s_1^2 と s_2^2 が含まれるので，マージと間引き操作を適用して得られる 3-レベルサマリ $s_1^3 = \{(4, 2.7), (8, 5.2), (12, 8.9), (16, 18)\}$ と置き換える．また同様に，b_5^1 と b_6^1 から 2-レベルサマリ $s_1^2 = \{(2, 2.4), (4, 4.2), (6, 8.1), (8, 15)\}$ が S に追加される．結果として，全体のサマリ S には s_1^3 と s_1^2 が含まれる．それぞれ対応する各レベルブロックのサマリとなっており，S から任意のイベント e のサポートを誤差 $\epsilon n = 6$ の範囲内で復元することが可能である．例えば，$e = 5$ を考えると，サポートは $sup(5, V_{24}) = sup(5, b_1^3) + sup(5, b_1^2) = 7 + 5 = 12$ であり，サポート見積は $c(5, S) = c(5, s_1^3) + c(5, s_1^2) = 4 + 4 = 8$ なので，確かに許容誤差内となっている．

補題 1 l-レベルブロック b^l に対し，Algorithm 1 により得られる l-レベルサマリを s^l と表す．このとき，s^l は，b^l の $(2^{l-1} - 1)$-近似サマリとなっている．

証明: 数学的帰納法により証明する．まず $l = 1$ のときは明らかに成り立つ．$l = k$ のとき，任意のイベント e に対して，$c(e^{(i)}, s^k) \leq sup(e, b^k) \leq c(e^{(i+1)}, s^k)$ が成り立つと仮定する．ただし，$e^{(i)}, e^{(i+1)}$ は s^k に登録されているイベントであり，$e^{(i)} \leq e \leq e^{(i+1)}$ を満たすものとする．$l = k+1$ の場合を考える．s^{k+1} は，ある k-レベルサマリ s_1^k と s_2^k から，マージ操作と間引き操作により得られる．仮定より，$c(e_1^{(i)}, s_1^k) + c(e_2^{(i')}, s_2^k) \leq sup(e, b^{k+1}) \leq c(e_1^{(i+1)}, s_1^k) + c(e_2^{(i'+1)}, s_2^k)$ が成り立つ．$e_1^{(i)}$ と $e_2^{(i')}$，$e_1^{(i+1)}$ と $e_2^{(i'+1)}$ はぞれぞれマージ後 (でかつ間引き前) の s^{k+1} 内で連続するエントリに登録されており，それぞれ少なくとも一つは間引き後に残る．残った 2 つのイベントは s^{k+1} 内で隣り合うので $c(e^{(u)}, s^{k+1}) \leq sup(e, b^{k+1}) \leq c(e^{(u)}, s^{k+1})$ と書ける．よって $l = k + 1$ の場合も成り立つ．□

定理 1 Algorithm 1 が出力するサマリ S は V_n の ϵn-近似サマリとなる．また S はマージ可能であり，そのサイズは $O(\frac{1}{\epsilon}\log\epsilon n)$ である．

証明: 補題 1 より S が ϵn-近似サマリであることは明らかである．また 2 つのサマリ S_1, S_2 が与えられたとき，各レベルのサマリ同士にマージと間引きを適用していくことで一つのサマリを構成できる．よって提案法のサマリはマージ可能である．レベルは最大でも $\log\frac{n}{k}$ であり，各レベルのサマリは高々一つである．また各レベルのサマリのエントリ数は間引き後 $k = \lceil \frac{1}{\epsilon} \rceil$ となる．よってサマリのサイズは $O(\frac{1}{\epsilon}\log\epsilon n)$ となる．□

4.2 半順序ストリームデータのサマリ構築

Algorithm 1 を束上のサマリ構築に拡張する．V_n を束 $\langle \mathcal{E}, \leq \rangle$ 上のストリームデータとすると，V_n 上の各 l-レベルサマリ s^l が $(\epsilon \times 2^{l-1} \times k)$-近似サマリとなるよう Algorithm 1 の操作 (1)(2)(3)(4) を修正すればよい．

1-レベルブロック b^1 から s^1 を構成する操作 (1)(2) を考える．全順序を扱った前節では，s^1 に対するイベント e のサポート見積 $c(e, s^1)$ は，s^1 において $e^{(i)} \leq e$ を満たすエントリ $(e^{(i)}, c^{(i)})$ の最大の $c^{(i)}$ としていた．全順序ではこのような $e^{(i)}$ が任意の e に対して必ず存在し同じサポートを持つ．他方，半順序ではそうなるとは限らない．ただし束の場合，任意のイベント e に対して，次の条件を満たす $\hat{e}$ が唯一存在する．

1. $sup(\hat{e}, b^1) = sup(e, b^1)$.
2. 上を満たす $\hat{e}'$ に対し，$\hat{e}' \leq \hat{e}$ ならば $\hat{e}' = \hat{e}$.

$b^1 = \langle e_1^1, \ldots, e_k^1 \rangle$ とすると，$\hat{e}$ は $\{e_i^1 \mid e_i^1 \in b^1, e_i^1 \leq e\}$ の上限に相当する．

いま U_k^i を $\{1, 2, \ldots, k\}$ の部分集合で i 個の要素を持つ集合とし，イベントの集合 $\{e_j^1 \mid e_j^1 \in b^1, j \in U_k^i\}$ の上限を $\hat{e}(U_k^i)$ と書く．この上限のサポートは，上限が空でない限り，i である．そこで，サポートが i であるような上限 $\hat{e}(U_k^i)$ の集合を $E^{(i)}$ と表し，組 $(i, E^{(i)})$ をエントリとして s^1 に登録することを考える．このとき s^1 におけるイベント e のサポート見積を次のように与えると，$c(e, s^1) = sup(e, b^1)$ が成り立つ．

$$c(e, s^1) = argmax_i(\exists \hat{e} \in E^{(i)}\ s.t.\ \hat{e} \leq e).$$

例 4 $\langle 2^{\{1,2,3,4\}}, \supseteq \rangle$ 上の V_4 を次のように与える:

$$V_4 = \langle \{2, 3, 4\}, \{1, 3, 4\}, \{1, 2, 4\}, \{1, 2, 3\} \rangle.$$

$\epsilon = 0.25$ (すなわち $k = 4$) のとき 1-レベルブロック b_1^1 のサマリ s_1^1 は $\{(1, E^{(1)}), (2, E^{(2)}), (3, E^{(3)})\}$，ただし

$$E^{(1)} = \{\{2, 3, 4\}, \{1, 3, 4\}, \{1, 2, 4\}, \{1, 2, 3\}\},$$
$$E^{(2)} = \{\{3, 4\}, \{2, 4\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{1, 2\}\},$$
$$E^{(3)} = \{\{1\}, \{2\}, \{3\}, \{4\}\}$$

となる．ここで，イベント $e = \{1, 4\}$ を考えると，$\hat{e} \supseteq e$ を満たすイベント $\hat{e}$ を含むエントリは $(1, E^{(1)})$ と $(2, E^{(2)})$ の 2 つである．よって $c(e, s^1) = 2$ となり，確かに e のサポートと一致する．

さて，この例の $E^{(1)}$，$E^{(2)}$，$E^{(3)}$ の和集合は，V_4 の全飽和アイテム集合族に相当することに気づく．すなわちトランザクションストリームの s^1 構築には，飽和アイテム集合を求める既存法を利用することができる．

他方，系列を含む一般束のサマリ構築では，束の閉包性に基づく**漸近交差法** (*incremental intersection*) [11, 3] が利用できる．1-レベルブロック $b^1 = \langle e_1^1, \ldots, e_k^1 \rangle$ において，e_1^1 から e_j^1 $(1 \leq j \leq k)$ までのイベントより構成される上限の全体集合を C_j と表す．このとき以下の漸化式が成り立つ．

$$C_{j+1} = C_j \cup \{gs(\hat{e}, e_{j+1}^1) \mid \hat{e} \in C_j, gs(\hat{e}, e_{j+1}^1) \neq \emptyset\}.$$

ただし $C_1 = \{e_1^1\}$ である．上式に基づき，逐次的かつ累積的に C_k を求めていく．その後，サポート単位で C_k を分割すれば各サポート i の $E^{(i)}$ が得られる．

次に，操作 (3) のマージを考える．半順序の場合，s_1^l と s_2^l の各エントリ中のイベント集合 $E_1^{(i)}$ と $E_2^{(j)}$ から，サポート見積が $i+j$ となる上限の集合 V を求める．この V は $\{gs(\hat{e}_1, \hat{e}_2) \mid \hat{e}_1 \in E_1^{(i)}, \hat{e}_2 \in E_2^{(j)}, gs(\hat{e}_1, \hat{e}_2) \neq \emptyset\}$ と表され，Algorithm 2 における関数 *merge* の 5 行目の関数 *gs* により求めている．

最後に，操作 (4) の間引きを考える．全順序の場合と同じく，マージ操作により得られる s^{l+1} には $2k$ 個のエントリが含まれ，その計数に応じて昇順整列されているものとする．このとき，奇数番目 ($2i+1$ とする) のエントリに登録されている各イベント e をすべて削除する．ただし，直前の $2i$ 番目のエントリに $e' \leq e$ となる e' が存在しない場合，e をその直前のエントリに移動する必要がある．この処理は，削除されるイベント e に必ず誤差 2^l の範囲内で e のサポートを復元できるイベント e' が残されることを保証するためである．なお全順序においても，一値一箇所の原則に従い，同じイベントは一つのエントリでのみ管理する場合，同様の処理が必要となる．

提案法の出力 S はマージ可能な ϵn-近似サマリとなり，サイズは $O(\frac{|E|}{\epsilon} \log \epsilon n)$ ($|E|$ は S の各エントリに保持されるイベント集合の最大の基数を表す) である．

Algorithm 2 マージと間引き

```
1: function MERGE(s_1^l, s_2^l)  ▷ s_1^l, s_2^l: l-レベルサマリ
2:     initialize s^{l+1}
3:     for each (i, E_1^(i)) ∈ s_1^l do
4:         for each (j, E_2^(j)) ∈ s_2^l do
5:             V := gs(E_1^(i), E_2^(j))
6:             if V ≠ ∅ & (i + j, E^(i+j)) ∈ s^{l+1} then
7:                 add V to E^(i+j)
8:             else if V ≠ ∅ then
9:                 add (i + j, V) to s^{l+1}
10:            end if
11:        end for
12:    end for
13:    return s^{l+1}                    ▷ l + 1-レベルサマリ
14: end function

15: function DROP(s^{l+1})              ▷ s^{l+1}: l + 1-サマリ
16:    delete the entry (2^l, E^(2^l)) from s^{l+1}
17:    for each i (1 ≤ i ≤ k − 1) do
18:        E := E^((2i+1)×2^l), E_p := E^((2i)×2^l)
19:        for each e ∈ E do
20:            if ∃e' ∈ E_p s.t. e' ≤ e then
21:                remove e from E
22:            else
23:                move e from E to E_p
24:            end if
25:        end for
26:        delete the entry ((2i+1)×2^l, E) from s^{l+1}
27:    end for
28: end function
```

5 提案法によるパターンマイニング

3 章で述べた通り，提案法はアイテム集合やアイテム系列を抽出対象とする頻出パターンマイニングに応用できる．頻出パターンマイニングでは，最小サポート率 σ が与えられ，$\sigma \times n$ より大きな頻度を持つ頻出パターン全体 $\mathcal{F}$ を求める．ここで σ より小さな ϵ を与えるとき，提案法の出力 S は，$\mathcal{F}$ の任意のパターン (集合もしくは系列) とその頻度を誤差 ϵn の範囲内で復元することができる．このような復元能力は **ϵ-劣性** (ϵ-*deficiency*) と呼ばれる．すなわち，提案法は $\mathcal{F}$ の ϵ-劣な非可逆圧縮表現 S を求めるオンライン近似アルゴリズムとみることができる．予備実験として，山梨大学への Web アクセスログデータ (2009 年 9 月の 2 週間分) をベンチマークとして用いて，ϵ-劣性を保証する先行研究 [5] のオンライン近似手法との比較を行った．図 2 に示す通り，先行研究に比べ，管理されるアイテム集合の個数が大幅に減少している．これは提案法の間引き操作による圧縮効果によるためと考えられる．

6 むすび

本稿では，半順序関係に基づくストリームデータの要約表現を求める近似アルゴリズムを提案している．ス

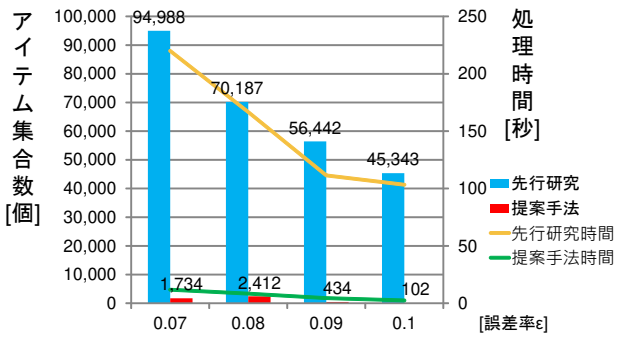

図 2: Web アクセスログでの実験結果

トリームデータマイニング分野では，HH 問題や QS 問題など，サマリと呼ばれるデータ要約表現をオンライン抽出する研究が進められている．本研究では，これら諸問題の一般クラスとして，半順序集合上のストリームデータから ϵn-近似サマリを求める問題を定式化し，その中で全順序と束関係に関するサマリ構築法を提案した．前者の提案法は，ϕ-分位数サマリを与える決定性アルゴリズムとして利用でき，理論的に最も優れた GK 法と同じ空間計算量で動作する．ただし GK サマリとは異なり，提案法はマージ可能なサマリを提供する．後者の束関係に関する提案法は，アイテム集合や系列を扱うパターンマイニングに適用することができる．サマリ構築の観点からパターンマイニングにアプローチする研究は我々の知る限り最初の試みであり，予備実験の結果，省メモリ化の効果が確認されている．今後，より規模の大きいベンチマークにおいて検証を行うとともに，系列マイニングにおいても性能評価を行う予定としている．

謝辞

本研究は一部，文科省科研費補助金 (基盤 C:17K00301 および 16K00298） の援助を受けている．

参考文献

[1] P. K. Agarwal *et al.*: Mergeable summaries, *Proc. on PODS*, 23–34 (2012)

[2] V. Braverman *et al.*: BPTree: an L_2 heavy hitters algorithm using constant memory, *Proc. on PODS*, 1–15 (2016)

[3] C. Borgelt, X. Yang, R. N. Cadenas, P. C. Saez and A. P. Montano: Finding closed item sets by intersecting transactions, *Proc. of the 14th Int. Conf. on Extending Database Technology* (EDBT2011), 367–376 (2011)

[4] M. Greenwald and S. Khanna: Space-efficient online computation of quantile summaries, *Proc. of Int. Conf. on Management of Data* (SIGMOD'01), 58–66 (2001)

[5] 岩沼宏治, 山本泰生, 福田翔士: ストリーム中の頻出飽和集合を抽出するオンライン型 ϵ-近似アルゴリズムの完全性, 人工知能学会論文誌, 31, 1–10 (2016)

[6] G. S. Manku and S. Rajagopalan and B. G. Lindsay: Approximate medians and other quantiles in one pass and with limited memory, *Proc. of Int. Conf. on Management of Data* (SIGMOD'98), 426–435 (1998)

[7] A. Metwally, D. Agrawal and A. E. Abbadi: Efficient computation of frequent and top-k elements in data streams, *Proc. of the 10th Int. Conf. on Database Theory* (ICDT2005), 398–412 (2005)

[8] G. Song *et al.*: CLAIM: an efficient method for related frequent closed itemsets mining over stream data, *DASFAA2007*, LNCS 4443, 664–675 (2007)

[9] D. Vesset *et al.*: IDC FutureScape: world wide big data andanalytics 2016 predictions, *IDC FutureScape* (2015)

[10] L. Wang, G. Luo, K. Yi and G. Cormode: Quantiles over data streams: an experimental study, *Proc. of Int. Conf. on Management of Data* (SIGMOD'13), 737–748 (2013)

[11] S.-J. Yen, C.-W. Wu, Y.-S. Lee, V. S. Tseng and C.-H. Hsieh: A fast algorithm for mining frequent closed itemsets overstream sliding window, *Proc. of the 2011 IEEE Int. Conf. on Fuzzy Systems*, 996–1002 (2011)

[12] Y. Yamamoto and K. Iwanuma: Online pattern mining for high-dimensional data streams, *Proc. of IEEE BigData2015*, 2615–2617 (2015)

[13] Q. Zhang and W. Wang: A fast algorithm for approximate quantilesin high speed data streams, *Proc. of IEEE 19th Int. Conf. on Scientific and Statistical Database Management* (SSDBM'07), p. 29 (2007)