F-043

精度保証付きオンライン型高速近似系列マイニング

Precision Guaranteed Online Type High-speed Approximation Algorithm

村田順平[1]　岩沼宏治[2]　石原龍一[23]　鍋島英知[2]
Junpei Murata[1]　Koji Iwanuma[2]　Ryuichi Ishihara[23]　Hidetomo Nabeshima[2]

1　はじめに

本論文では，巨大なストリームデータ中に頻出する部分系列の抽出を目的としたオンライン型高速近似アルゴリズムを提案する．また，提案アルゴリズムの近似精度を示し，その有用性を評価実験により示す．

近年では，通信技術の向上やストレージ容量の増加により，大量のデータが高速で流れるようになった．それにより，データベースのサイズが高速に巨大化しいる．例えばクレジットカードの利用履歴や通信のトラフィックなど，様々なものがある．このようなデータを**ストリームデータ**という．ストリームデータをマイニングする際，Apriori アルゴリズムなどのオフライン型アルゴリズムでは実行時間がデータの蓄積速度に追いつかない問題が発生する．そこで，近年ではデータベースを一度しか読まずに，高速に処理を行うオンライン型アルゴリズムの研究が盛んに行われている [1, 2, 3].

オンライン型アルゴリズムでは，メモリ空間消費を抑えることが大きな課題となる．厳密解を求めていてはこの問題の解決が困難なため，近似解を求めて効率のよい計算を実現する戦略が一般的である [2]．本研究でも空間消費抑制のため，冗長解の生成，つまり頻出でない部分系列の抽出を一部許す．本研究で提案するオンライン型アルゴリズムは，データベース中に頻出する部分系列すべてを抽出し，また，冗長解についても精度保証をする．

本論文では，データベース中に頻出する要素を抽出するオンライン型アルゴリズムである Lossy Counting 法 [1, 2, 3] を基礎とし，系列先頭頻度 [4] を用いることで，長大な系列データ上に頻出する部分系列の高速抽出を目指す．石原ら [5] が同様の手法を提案しているが，誤差の考慮がされておらず，またアルゴリズムも複雑なものであった．本論文では新たに誤差の精度保証を持つ簡潔なアルゴリズムを提案する．頻出部分系列抽出問題は，頻出アイテム抽出問題に比べ空間使用量が大幅に大きいので，空間使用量を制限する方法を提案し，その手法の誤差を示す．また，部分系列抽出問題においても，空間使用量は Lossy Counting 法と同様に系列長の対数程度のサイズに押さえ込むことを示し，実験により確認する．さらに，提案手法の時間計算量が読み込んだ系列長に比例することを示す．

2　準備

本論文で扱う表記法と用語の定義を以下に示す．

定義 1 すべてのアイテムの集合を $I = (i_1, i_2, \cdots, i_n)$ とする．**アイテム集合系列** (以下，系列と呼ぶ) とは，アイテム集合の並びであり，$S = \langle s_1 s_2 \cdots s_n \rangle$ と表記する．各 $s_i (s_i \subseteq I, 1 \leq i \leq n)$ を S の**要素**と呼び，$(a_1, a_2, \cdots, a_m)$ と略記する．S の**系列長**を $|S|$ で表す．系列 $\alpha = \langle s_1 \cdots s_m \rangle$ が系列 $\beta = \langle t_1 \cdots t_n \rangle$ の**部分系列**であるとは，$s_1 \subseteq t_{j_1}$，$s_2 \subseteq t_{j_2}$，$\cdots$，$s_m \subseteq t_{j_m}$ を満たす整数 $1 \leq j_1 < j_2 < \cdots < j_m < n$ が存在する場合をいい，$\alpha \subseteq \beta$ と表す．

定義 2 単一系列データベースとは，アイテム集合系列であり，マイニングの対象となるデータベースである．以下，系列データベースと呼ぶ．

以下，ストリームデータを系列データベースと呼ぶ．

定義 3 アイテム系列とは，アイテムの単一要素集合の系列 $\langle (a_1)(a_2)\cdots(a_n) \rangle$ であり，アイテムの系列 $\langle a_1 a_2 \cdots a_n \rangle$ と同一視する．本研究で抽出する部分系列は，すべてアイテム系列である．

定義 4 $S = \langle s_1 s_2 \cdots s_n \rangle$ を系列データベース，$\alpha = \langle t_1 t_2 \cdots t_m \rangle$ をアイテム系列とする．S 上の α の出現頻度関数 $F(S, \alpha)$ とは，整数値 $k (0 \leq k \leq |S|)$ を返す関数である．また，このとき α の**相対頻度** $R(S, \alpha)$ を以下のように定義する．

$$R(S, \alpha) = \frac{F(S, \alpha)}{|S|}$$

相対頻度 $R(S, \alpha)$ は，α が系列データベース S 中にどの程度の割合で出現するかを表し，1 以下の非負数を取る．

定義 5 N 個の要素からなる系列データベース $S = \langle s_1 s_2 \cdots s_N \rangle$ を考える．このとき，部分系列 α が S 中に**頻出**であるとは，与えられた最小サポート値 $\sigma (0 < \sigma < 1)$ に対し，$F(S, \alpha) \geq \sigma N$，もしくは $R(S, \alpha) \geq \sigma$ が成り立つことをいう．

アイテムの頻度を計算する場合，出現頻度関数はそのアイテムがデータベース中に単純に何回出現しているかを返せばよいだけだが，部分系列の場合，そのような単純な計算方法がない．そこで本論文では，出現頻度関数として**系列先頭頻度**と**系列全体頻度** [4] を導入する．系列先頭頻度は逆単調性を持たないが，重複数え上げが生じない頻度尺度である．系列全体頻度は逆単調性を持ち，重複数え上げが生じない尺度であるが，算出方法が複雑である．系列先頭頻度と系列全体頻度を以下に定義する．

定義 6 系列データベース $S = \langle s_1 s_2 \cdots s_n \rangle$ があるとき，S の i 番目の要素から始まる長さ k の部分系列を**ウィンドウ**と呼び，$\mathrm{win}(S, i, k)$ と表記し，以下で定義する．また，このときの k を**ウィンドウ幅**という．

$$\mathrm{win}(S, i, k) = \begin{cases} (s_i \cdots s_{i+(k-1)}) & \text{if } i + (k-1) \leq n, \\ (s_i \cdots s_n) & \text{otherwise.} \end{cases}$$

[1]山梨大学大学院コンピュータ・メディア工学専攻
[2]山梨大学大学院医学工学総合研究部
[3]現在 ニューメディア総研

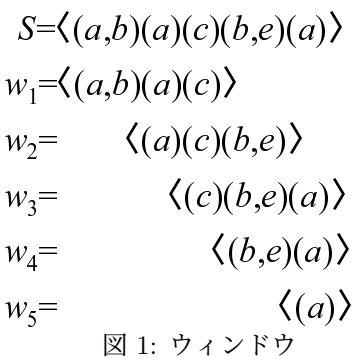

図 1: ウィンドウ

ウィンドウは，データベース中の部分系列を探すときの注目範囲にあたる．

定義 7 $\alpha = \langle a_1 \cdots a_m \rangle$ をアイテム系列，$\beta = \langle b_1 \cdots b_n \rangle$ をアイテム集合系列とするとき，$\alpha \lhd \beta$ を $a_1 \in b_1$ かつ $\alpha \subseteq \beta$ が成り立つ場合と定める．

定義 8 系列データベース $S = \langle s_1 s_2 \cdots s_n \rangle$，アイテム系列 $\alpha = \langle t_1 t_2 \cdots t_m \rangle$，ウィンドウ幅 k $(1 \leq m \leq k < n)$ に対し，S における α の**系列先頭頻度**H-freq(S, α, k) を以下で定義する．

$$\text{H-freq}(S, \alpha, k) = \sum_{i=1}^{n} \delta(\text{win}(S, i, k), \alpha)$$

ここで，δ は以下の関数と定める．

$$\delta(\langle s_i \cdots s_n \rangle, \langle t_1 \cdots t_m \rangle) = \begin{cases} 1 & \text{if } \langle t_1 \cdots t_m \rangle \lhd \langle s_1 \cdots s_n \rangle \\ 0 & \text{otherwise} \end{cases}$$

例 1 系列データベース $S = \langle (a,b)(a)(c)(b,e)(a) \rangle$，ウィンドウ幅 $k = 3$ であるとき，アイテム系列 $\langle a \rangle$，$\langle ac \rangle$ の系列先頭頻度を考える．

S のウィンドウは，図1にあるように先頭から順に $w_1 = \langle (a,b)(a)(c) \rangle$，$w_2 = \langle (a)(c)(b,e) \rangle$，$w_3 = \langle (c)(b,e)(a) \rangle$，$w_4 = \langle (b,e)(a) \rangle$，$w_5 = \langle (a) \rangle$ となる．S 中におけるある系列 α の系列先頭頻度は，先頭部分に α を部分系列として含むウィンドウの数である．この場合，$\langle a \rangle$ は w_1, w_2, w_5 に，$\langle ac \rangle$ は w_1, w_2 にそれぞれ含まれている．よって，系列 $\langle a \rangle$，$\langle ac \rangle$ の系列先頭頻度はそれぞれ 3, 2 となる．

定義 9 系列データベース $S = \langle s_1 s_2 \cdots s_n \rangle$，アイテム系列 $\alpha = \langle t_1 t_2 \cdots t_m \rangle$，ウィンドウ幅 k $(1 \leq m \leq k < n)$ に対し，S における α の**系列全体頻度**T-freq(S, α, k) を以下で定義する．

$$\text{T-freq}(S, \alpha, k) = \min_{\beta \subseteq \alpha}(\text{H-freq}(S, \beta, k))$$

オンライン型アルゴリズムでは Apriori などの逆単調性を利用した枝刈りを行わない．よって本論文では，逆単調性は無いが算出方法が単純な系列先頭頻度を出現頻度関数として用いる．系列全体頻度の適用については，応用として節 3.2.4 で述べる．以降，出現頻度関数 F は，系列先頭頻度を表すものと約束する．

3 オンライン型アルゴリズム

3.1 Lossy Counting 法 [Manku [1]]

本節では，先行研究である Lossy Counting 法 (以下 LC 法) について簡単に述べる．

LC 法は，最小サポート値 $\sigma (0 < \sigma < 1)$ と許容誤差 $\epsilon (0 < \epsilon < \sigma)$ を受け取り，単一アイテムを要素としたストリームデータ中に頻出する全アイテムと，幾らかの非頻出なアイテムを出力する．読み込んだストリームデータの長さが N である場合，抽出されるアイテムの出現頻度は少なくとも $(\sigma - \epsilon)N$ 以上である事が保証される．

LC 法では，ストリームデータを仮想的に $w = \epsilon^{-1}$ 個のアイテムが入ったバケットに区切る．各バケットには 1 番から始まる番号を付け，現在のバケット番号を b とする．読み込んだアイテムが N 個目である場合，$b = \lceil \frac{N}{w} \rceil$ となる．アルゴリズムが保持する**頻度表** D は，三つ組み $(e, C(e), \Delta(e))$ の集合である．e はアイテム，$C(e)$ は e が D に格納されてからの出現頻度，$\Delta(e)$ は e の真の出現頻度 $F(e)$ と $C(e)$ の誤差の最大値である．e に対応する三つ組みは高々一つである．相対頻度の小さいアイテムの情報は頻度表から削除し，空間使用量を削減する．

LC 法は，以下のステップをユーザから出力要請があるまで繰り返す．

1. ストリームデータからアイテム e を読む．
2. e の三つ組みが頻度表 D があれば，その $C(e)$ の値を 1 だけ増やす．
3. 三つ組みが無ければ，$(e, 1, b-1)$ を D に挿入．
4. 次のバケットに移る際に，D をチェックし，$C(e) \leq b - \Delta(e)$ となる三つ組みを削除する．

ユーザが出力要請を出したならば，$C(e) \geq (\sigma - \epsilon)N$ となるアイテム e を頻出であるとして出力する．

このアルゴリズムでは以下の定理が成り立つ．定理の証明は，参考文献 [1] を参照していただきたい．

定理 1 読み込んだ系列長 N，最小サポート σ，誤差 ϵ のとき，出力されたアイテム e の真の出現頻度 $F(S, e)$ は，$F(S, e) \geq (\sigma - \epsilon)N$ を満たす．

定理 2 読み込んだ系列長 N，誤差 ϵ のとき，このアルゴリズムのメモリ空間使用量は，$\epsilon^{-1} \log(\epsilon N)$ 個の三つ組みが格納できるサイズで十分である．

3.2 提案手法

本節では，LC 法を頻出部分系列問題に拡張した提案手法について述べる．

提案手法は，最小サポート値 $\sigma (0 < \sigma < 1)$，許容誤差 $\epsilon (0 < \epsilon < \sigma)$，ウィンドウ幅 $k (1 < k)$ を受け取り，マイニング対象であるストリームデータ中に頻出する部分系列全てを抽出する．幾つかの非頻出な部分系列も抽出するが，読み込んだストリームデータの長さが N である場合，抽出される部分系列の出現頻度は少なくとも $(\sigma - \epsilon)N$ 以上であることが保証される．また，LC 法と同様，相対頻度が必ず ϵ 以下であるといえる部分系列を保持しないことで，空間使用量の増加を抑制する．

提案手法では，アイテム集合系列から部分系列を抽出するので，アイテム系列からアイテムを抽出する LC 法と比べ，空間使用量が非常に大きい．そのため，空間使用量の抑制方法を用いる必要がある．しかし，使用できるメモリ空間が巨大であり，空間使用量の抑制が必要ないとするならば，LC 法とほぼ同様の処理が可能である．まず，そのアルゴリズムを解説する．空間使用量の抑制方法は節 3.2.5 で述べる．

3.2.1 アルゴリズム

提案手法では，図 1 のようにウィンドウを単位時刻ごとにスライドさせながら，ストリームデータ中に出現する部分系列の出現頻度を数える．アルゴリズムが保持する情報である頻度表 D は，三つ組み $(\alpha, C(\alpha), T(\alpha))$ の集合である．α はウィンドウから得られた系列，$C(\alpha)$ は α が D に格納されてからの出現頻度，$T(\alpha)$ は D に格納された時刻である．また，部分系列 α に対応する三つ組みは高々一つである．

ユーザからの出力要請があるまで，以下のステップを繰り返す．

1. 現時刻を t とする．ウィンドウ w を読み，$\alpha \lhd w$ となる α すべてに，以下の 2,3 の処理を行う
2. α に対応する三つ組みが頻度表 D があれば，その $C(\alpha)$ を一つ増やす．
3. α に対応する三つ組みが無ければ，$(\alpha, 1, t)$ を D に挿入．
4. D をチェックし，$C(\alpha) \leq \epsilon \times (t - (T(\alpha) - 1))$ となる三つ組みを削除する．
5. 時刻を 1 だけ進め，ウィンドウをスライドする．

出力要請を受けると，$C(\alpha) + \epsilon \times (T(\alpha) - 1) \geq \sigma t$ となる部分系列 α を頻出であるとし，出力する．

4. では，頻度表に挿入されてからの相対頻度が誤差 ϵ 以下である部分系列を全て，頻度表から削除する．現在の時刻を t としたとき，$t - (T(\alpha) - 1)$ とは，α が頻度表 D に挿入された時刻から，現時刻までに読んだウィンドウの数を表す．よって，$\epsilon \times (t - (T(\alpha) - 1))$ は，相対頻度が ϵ となるための出現頻度を表す．$C(\alpha)$ がこの値以下ならば，α が D に格納されてからの相対頻度が ϵ 以下であると言える．

続いて，アルゴリズムの正当性を説明する．

定理 3 ストリームデータ S，最小サポート値 σ，許容誤差 ϵ，頻度表 D に格納された三つ組み $(\alpha, C(\alpha), T(\alpha))$ において，以下の式が必ず成り立つ．

$$C(\alpha) \leq F(S, \alpha)$$

証明 α は部分系列，$C(\alpha)$ は D に格納されてからの出現頻度，$T(\alpha)$ は D に格納された時刻である．このアルゴリズムは，頻度表 D に格納されてからの相対頻度が誤差 ϵ 以下である部分系列を全て頻度表から削除するので，D 中の部分系列 α は，時刻 $T(\alpha)$ 以前に出現した可能性がある．よって，真の出現頻度 $F(S, \alpha)$ は必ず表に格納されてからの頻度 $C(\alpha)$ 以上である． □

定理 4 ストリームデータ S，その長さ N，最小サポート値 σ，許容誤差 ϵ，頻度表 D 中の三つ組み $(\alpha, C(\alpha), T(\alpha))$ において，このアルゴリズムで抽出される部分系列 α の真の出現頻度 $F(S, \alpha)$ は以下の式を満たす．

$$F(S, \alpha) \geq (\sigma - \epsilon)N$$

証明 まず，α が D に格納される以前にどれだけ出現していた可能性があるかを考える．単位時刻ごとにウィンドウを一つスライドさせるので，α が D に格納される以前に読み込んだウィンドウの数は $T(\alpha) - 1$ である．定義 8 より，各ウィンドウにおける α の出現回数は高々一回である．また，格納されてからの相対頻度が ϵ 以下ならば頻度表に格納されないので，時刻 $T(\alpha)$ 以前の α の出現回数は最大で $\epsilon \times (T(\alpha) - 1)$ である．この値を超えて出現する部分系列は頻度表からは決して削除されず，保持され続けられる．よって，α の真の出現頻度 $F(S, \alpha)$ は $C(\alpha)$ より最大で $\epsilon \times (T(\alpha) - 1)$ 大きいといえる．これらと定理 3 をあわせて以下の式が成り立つ．

$$C(\alpha) \leq F(S, \alpha) \leq C(\alpha) + \epsilon \times (T(\alpha) - 1)$$

よって，読み込んだ系列長が N のとき，部分系列 α が $C(\alpha) + \epsilon \times (T(\alpha) - 1) \geq \sigma N$ を満たすならば，$F(S, \alpha) \geq \sigma N$ も満たされる可能性があるので，α は頻出部分系列として抽出される．また，α が抽出される場合，$N \geq T(\alpha) - 1$ と定理 3 より，以下の式も成り立つ．

$$F(S, \alpha) + \epsilon N \geq \sigma N$$

よって

$$F(S, \alpha) \geq (\sigma - \epsilon)N$$

以上より，抽出される部分系列の真の出現頻度は必ず $(\sigma - \epsilon)N$ となる． □

3.2.2 提案手法の空間使用量

提案手法のアルゴリズムの空間使用量はどれほどに抑えられるかを示す．空間使用量は，LC 法 [1, 3] とほぼ同様の考え方で求められる．

定理 5 各ウィンドウで得られる部分系列の最大数を M，許容誤差 ϵ，読み込んだ系列長 N があるとき，アルゴリズムが利用する三つ組の数は，$\frac{M}{\epsilon} log N$ 個以下である．

証明 頻度表 D の三つ組み $(\alpha, C(\alpha), T(\alpha))$ において，$T(\alpha) - 1 = N - i$ であるものの数を d_i とする．このとき，頻度表 D 内の三つ組みの数を $|D|$ とすると，$|D| = \sum_{i=1}^{N} d_i$ である．$T(\alpha) - 1 = N - i$ である三つ組みにおいて i とは，α が D に格納されてから現時刻までに読み込まれた系列の長さに等しい．よって，この三つ組みの $C(\alpha)$ は ϵi 以上となる．このことから，$T(\alpha) - 1 = N - i$ である各三つ組みの $C(\alpha)$ の総和は，$\epsilon i d_i$ 以上となる．

データストリームの末尾から長さ j の区間で出現する部分系列の延数は，各ウィンドウで出現する部分系列を M 個としているので，jM となる．したがって，以下の式が成り立つ．

$$\sum_{i=1}^{j} \epsilon i d_i \leq jM \quad (j = 1, 2, \cdots, N)$$

$$\sum_{i=1}^{j} i d_i \leq \frac{jM}{\epsilon} \tag{1}$$

以上より，以下の式 (2) が j に関する帰納法により証明できる．証明は紙面の都合上省略させていただく．

$$\sum_{i=1}^{j} d_i \leq \frac{M}{\epsilon} \sum_{i=1}^{j} \frac{1}{i} \tag{2}$$

頻度表 D 内の三つ組みの数は $|D| = \sum_{i=1}^{N} d_i$ であるので，

$$|D| \leq \frac{M}{\epsilon} \sum_{i=1}^{N} \frac{1}{i} \leq \frac{M}{\epsilon} \log N$$

以上より，定理 5 は成り立つ． □

対象ストリームデータの各要素はアイテム集合であることに注意されたい．

3.2.3 Lossy Counting 法の単純拡張

LC 法を部分系列抽出問題に単純に拡張することを考える．LC 法の各ステップにおいて，アイテムではなく部分系列を読み込む動作に変更するだけでも，頻出部分系列の抽出は可能である．また，保持する必要がある三つ組みの数の見積もりは $\frac{M}{\epsilon} \log \epsilon N$ 以下となり，提案手法よりも小さくなる．しかし提案手法における頻度の見積もりは，バケット単位ではなく各ウィンドウで行うので，より精密になる．そのため，抽出した頻出部分系列それぞれの頻度の見積もり精度が提案手法の方が高くなる．

3.2.4 系列全体頻度の計算

ストリームデータ上で，定義 9 で定義した系列全体頻度の計算も容易に行える．

系列 α の系列全体頻度は，α の部分系列すべての系列先頭頻度を求める必要があるため，ストリームデータの走査中に系列全体頻度を計算することが困難である．しかし，系列先頭頻度を用いて頻出部分系列を抽出し，抽出した部分系列だけを用いてそれぞれの系列全体頻度を計算することは可能である．

定義 10 系列 $\alpha = \langle a_1 a_2 \cdots a_n \rangle$ があるとき，α の**接尾辞**とは，系列 $\langle a_i \cdots a_n \rangle$ であり，$suf(\alpha, i)$ で表す．

補題 1 系列データベース S，ウィンドウ幅 k，アイテム系列 $\alpha = \langle a_1 a_2 \cdots a_n \rangle$ があるとき，以下が成り立つ．

$$\text{T-freq}(S, \alpha, k) = \min_{i=1}^{n}(\text{H-freq}(S, suf(\alpha, i), k))$$

補題 1 は，アイテム系列 α の各接尾辞の系列先頭頻度だけで，α の系列全体頻度を計算できることを示す．証明は省略する．参考文献 [4] を参照されたい．

補題 1 より，系列 α の接尾辞の系列先頭頻度のうち，その最小値が α の系列全体頻度となる．よって，α 自身も含め，α の接尾辞全てが系列先頭頻度において頻出であるならば，α は系列全体頻度においても頻出である．逆に，α の接尾辞が一つでも系列先頭頻度において頻出でないならば，α は系列全体頻度において頻出でない．

3.2.5 頻度表サイズ拡大の制限とそれに関する誤差保証

提案手法では，アイテム集合の系列であるストリームデータも対象とするので，要素の大きさやウィンドウ幅によって，一つのウィンドウから得られる部分系列の数が膨大になる．そのため，保持すべき部分系列の情報がおおくなり，メモリ容量を超える可能性がある．その場合に考えられる対処法と，それによる誤差を考える．

処理の概要は，新たに頻度表に挿入する部分系列の数を制限することで，空間使用量を抑制する．その方法は以下の二つが考えられる．

1. 頻度表のサイズを制限し，そのサイズを超えて部分系列を挿入しない．
2. 頻度表に新たに格納する部分系列数を制限し，頻度表が急激に増大することを防ぐ．

どちらも，制限を超えた分の部分系列を**消去**する．消去する条件が異なるだけで，本質的には同じである．

アルゴリズムは以下の通りである．条件により挿入できない部分系列はそのまま消去し，消去が起きた回数だけを記録する．挿入できなかった部分系列の数を記録するわけではないことに注意していただきたい．例えば，あるひとつのウィンドウを読んだときに，10 種類の部分系列の情報が挿入できなかった場合，消去の回数は 10 回ではなく 1 回である．消去が起きた回数は，頻度表からの消去や抽出の際に見積もる頻度の計算に使用する．

実現には，保持する情報として，消去回数 K と，さらに，頻度表中の部分系列 α に対応する $K(\alpha)$ を追加する．$K(\alpha)$ は α が頻度表に追加される以前に起きた消去の回数を表す．そして，各ウィンドウの部分系列 α すべてに対し，以下を行う．

1. α が頻度表 D にあるならば，$C(\alpha)$ を 1 だけ増やす．

2-a. α が頻度表 D になく，消去が起きなければ D に $(\alpha, 1, t, K)$ を挿入．

2-b. 各条件により α が挿入できないならば，α を消去．

3. 消去が起きたならば，次のウィンドウに移る際に K を 1 だけ増やす．

また，節 3.2.5 のアルゴリズム 4. にある削除条件は，以下のように変更する．

$$C(\alpha) + M(\alpha) - \epsilon(t - (T(\alpha) - 1)) \leq 0$$

α は，過去に $K(\alpha)$ 回消去されている可能性があるので，頻度の見積もりに $K(\alpha)$ だけ足す．また，抽出条件も以下のように変更する．

$$C(\alpha) + K(\alpha) + \epsilon \times (T(\alpha) - 1) \geq \sigma t$$

この手法では，誤差が最大で K 回大きくなる．消去の回数が小さく，ストリームデータが長ければ大きな誤差にならないので，十分に信頼できる手法であるといえる．消去の回数が σN を超えると，出現する全ての部分系列が頻出であるとみなされる．よって，消去の回数が σN を超えないよう消去の条件を設定する必要がある．

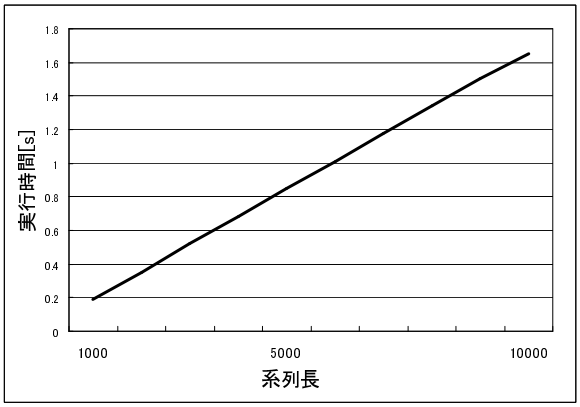

図 2: 実行速度の変化

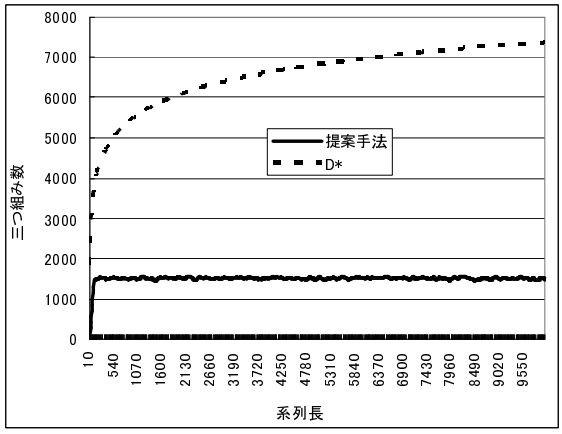

図 3: 系列長の増加に伴う空間使用量の変化

4 評価実験

提案手法を C 言語で実装し，単一アイテムを要素とした系列データベース S を対象に実験を行った[1]．実験パラメータは，ウィンドウ幅 5，最小サポート値 $\sigma = 0.020$，許容誤差 $\epsilon = 0.010$ とする．対象の系列は実験用に作成したものであり，出現するアイテムは 100 種類で，それぞれ 1〜100 の ID が付く．アイテムの出現確率は，平均 50，分散 19 で正規分布しており，ID50 が最も頻繁に出現し，50 から離れるほど出現確率が低くなる．

実験環境は，OS:Turbo Linux 10 Server，CPU: Pentium(R) 4 CPU 3.60GHz，メモリ: 3GB である．

4.1 実行速度

データベース S の系列長を 1000 から 10000 まで，1000 ずつ長く変化させ，データベースの拡張に伴う実行時間の変化を調べた．実験結果を図 2 に示す．

図 2 より，データベースの拡張に伴い実行時間が線形に増加していることが分かる．このことから，本手法はデータベースの系列長の拡張性に優れているといえる．

4.2 空間使用量

頻度表 D の三つ組みの数が $\frac{n}{\epsilon}logN$ 以下になることを，実験により示す．同時に，三つ組みを単位とし，実際の空間使用量を調べる．ウィンドウ幅が 5 のとき，要素が全て単一アイテムであるウィンドウから得られる部分系列の数は最大で 16 であるので，$n = 16$ として計算する．

実験結果を図 3 に示す．図中の破線 **D*** は $\frac{n}{\epsilon}logN$ の値である．実際の三つ組みの数は D*よりも大幅に少ない．三つ組み数の増加は非常に緩やかであり，系列長が長くなるにつれて，空間使用量の見積もり値との差は大きくなっていくことが分かる．

5 まとめ

本論文では，対象のストリームデータ中に頻出アイテムを近似的に抽出するオンライン型アルゴリズムである LC 法 [1] を基礎とし，先頭系列頻度 [4] を導入することで，頻出部分系列を高速近似抽出するオンライン型アルゴリズムを提案した．提案手法の正当性を示し，また，空間使用量の見積もりを行った．さらに，空間使用量の制限手法を提案し，その手法の誤差を示した．評価実験により本アルゴリズムの有用性を示した．

今後の課題として，人工データベースでなく，様々な実データを用いて，空間使用量に関する実験する必要がある．また提案手法は，対象系列の要素がアイテム集合である場合でも動作するようになっているが，まだ実装ができていないので，実際に実装し実験することで，アイテム集合の系列を対象としても提案手法が動作することを示す必要がある．

[1] アイテム集合を要素とした系列データベースは，時間の都合上実装できなかった．

謝辞

本研究の一部は文科省科学研究費補助金 (No.20240016) の支援を受けている．

参考文献

[1] G.S. Manku and R. Motwani: Approximate frequency counts over data streams. Proc. VLDM'02, pp.346-357, 2002.

[2] 有村博紀: 大規模データストリームのためのマイニング技術の動向．電子情報通信学会論文誌，Vol.J88-D-I, No.3, pp.563-575, 2005

[3] 徳山豪: オンラインアルゴリズムとストリームアルゴリズム，井立出版，2007

[4] K Iwanuma, R Ishihara, Y Takano, H Nabeshima: Extracting Frequent Subsequences from a Single Long Data Sequence: A Novel Anti-Monotonic Measure and a Simple On-Line Algorithm. Proc. off IEEE ICDM, pp.186-193, 2005

[5] 石原龍一，岩沼宏治，鍋島英知: 時系列データ中の頻出部分系列を高速抽出するオンライン近似計算法，山梨大学大学院医学工学教育部コンピュータ・メディア工学専攻修士論文，2006