大量データストリームの類似探索手法

A Similarity Search Method for Multiple Data Streams

藤原 靖宏♦ 櫻井 保志♦
山室 雅司♦

Yasuhiro FUJIWARA Yasushi SAKURAI
Masashi YAMAMURO

現在データストリームを利用したアプリケーションへの注目が金融，環境，モバイル，ウェブアプリケーション，製造等の分野で集まっている．本研究では流入する複数のシーケンスのうち，探索時刻から任意の長さの問い合わせに対して類似したシーケンスの組み合わせを探索する問題を対象とする．本研究では DAPSS(DAta stream Processing for Store and Search) を提案する．DAPSS はこの問題を高速，正確，省メモリに解くことができる．DAPSS について性能評価を行った結果，ナイーブな手法と比較して数十倍高速に処理が行えることを確認した．

There is much interest in the processing of data streams for applications in the fields such as finance, environment, mobile communications, webservices, and manufacturing. This paper focuses on the problem to search, exactly, similar pairs of streaming data sequences of arbitrary length. We propose DAPSS(DAta stream Processing for Store and Search). DAPSS can address this problem fast, accurate, and with small memory. Experiments show that DAPSS is dozens of times faster than the naive method while outputs are exact.

1. はじめに

現在データストリームを利用したアプリケーションへの注目が金融，環境，モバイル，ウェブアプリケーション，製造等の分野で集まっており，またその分野の研究も盛んである[4][6][12]．本研究では特に複数のシーケンスから構成されるデータストリームを対象とする．一般的にデータストリームは高いビットレートで長い期間にわたって流入するため，処理するべきデータ量は膨大になる．そのためデータストリームの処理に対しては，限られたメモリ量でかつ高速に行うことが要求される [8][2][3]．

多くの既存研究では上記の要求を満たすため，データは処理後に廃棄され，また精度は犠牲にされてきた．しかしデータを廃棄することはデータストリームが処理後の分析などに用いられることがありえるため好ましくない．そしてまた精度を犠牲にすることもアプリケーションの応用を考えたときに好ましくない．そのため本研究ではデータストリーム処理の要求として先に挙げたものの他に，データストリームの処理結果が厳密に正確であるという要求を加える．

本研究では流入する複数のシーケンスのうち，探索時刻から任意の長さの問い合わせシーケンスに対して類似したシーケンスの組み合わせを探索する問題を対象とする．この問題は多くの分野で応用が可能である．

データストリームのデータ量は膨大であるため，従来この問題を解くには多くの計算量とメモリ量が必要であった．本研究では DAPSS(DAta stream Processing for Store and Search) を提案する．DAPSS はこの問題を高速，正確，省メモリで処理できる．

DAPSS では処理結果の厳密性を達成するため，データシーケンスの探索処理においてメモリ内に格納したデータシーケンスの特徴量とディスクに格納したオリジナルのデータシーケンスを用いる．

本研究では DAPSS について人工データと実測データを用いて性能評価を行った．検証した結果ナイーブな手法と比較して高速に処理が行えることを確認した．

2. 関連研究

本研究は流入するシーケンスを対象とするが，蓄積されたシーケンスを対象とした研究は過去から多くされている．Agrawal らは whole matchig (等しい長さのシーケンスが対象)について研究した[1]．Agrawal らの手法は Faloutsos らによって subsequence matching (異なる長さのシーケンスが対象)へと拡張された[7]．

現在流入するシーケンスを対象とした研究が盛んである．Zhu らは流入するシーケンスの相関関係を高速に計算する手法について研究した[10]．Bulut らは流入するシーケンスの相関関係などを様々な長さの window に対して計算する手法について研究した[5]．

3. 問題設定

本研究では流入する m 個のシーケンスのうち，探索時刻からユーザが希望する任意の長さの問い合わせシーケンスに対して類似したシーケンスの組み合わせを探索する問題を扱う．この問題の解をシーケンス $X=(x_1,x_2,\cdots,x_n)$ と $Y=(y_1,y_2,\cdots,y_n)$ のユークリッド距離 $D(X,Y)$ を用いて以下のように定義する．

問題 問い合わせシーケンスの長さ l と閾値 ε が与えられたとき，類似したシーケンスを検知する問題の解は以下の条件を満たすシーケンスの組み合わせ X_l と Y_l のすべての集合とする．

$$D(X_l,Y_l)=\sqrt{\sum_{t=n-l+1}^{n}(x_t-y_t)^2}\le\varepsilon \quad (1)$$

この問題をナイーブに解く場合はすべてのシーケンスをメモリ上に保持しておいて，すべてのシーケンスの組み合わせについて距離計算を行う．ナイーブな手法の問題点として，多くのメモリ量が必要になることと，多くの計算量が必要になることが挙げられる．

♦ 正会員 日本電信電話株式会社 NTTサイバースペース研究所
{fujiwara.yasuhiro,sakurai.yasushi,yamamuro.masashi}@lab.ntt.co.jp

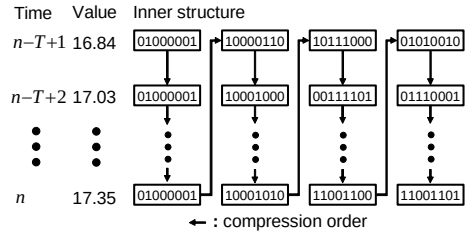

図 1 可逆圧縮
Fig.1 Lossless compression

4．提案手法

4.1 DAPSS で用いる手法

可逆圧縮 可逆圧縮は高速にシーケンスを圧縮するための手法である．可逆圧縮によりディスクにアクセスする I0 コストを低減できる．

PSA PSA(Piecewise Statistical Approximation) はシーケンスの平均と標準偏差を特徴量として用いてシーケンスの距離を高速に近似計算するための手法である． PSA では設定する距離関数によりシーケンス間の距離の下限値または上限値を計算できるので，ほとんどディクスにアクセスせずに類似シーケンスを求めることができる．

マトリクス マトリクスは多次元空間内の複数の基準点からの距離を用いて類似シーケンスを絞り込むための手法である．マトリクスにより膨大な組み合わせの中から類似データシーケンスの候補を高速に絞り込むことができる．

4.2 可逆圧縮

4.2.1 前処理

本手法の前処理ではデータを分割して並び替える．図１に示すように計算機の内部においてはシーケンスのデータ値は複数バイトの集合として表現される．図１においてTは保存するシーケンスの長さである．ひとつのデータ値においては隣り合うバイトは似ていないため，データ値をそのまま圧縮してもそれほど効果は期待できない．しかしシーケンスにおいてはデータ値が漸次的に変化する特徴があるため，複数データ値の符号部と仮数部は似ている特徴がある．そのため本手法ではT時間ごとにシーケンスをバイト単位に分割し，並び替えたバイト列に対して圧縮を行う．

4.2.2 圧縮・格納方法

符号化にはランレングス符号化を用いる．符号化には他にハフマン符号化[9]や LZ77 符号化[11]などがあるが，ランレングス符号化はワンパスで実行できるので高速処理に向いているためである．

更新処理においてはシーケンスごとにシーケンシャルファイルの形で格納する．これは探索処理においてデータシーケンスを参照するときに高速なシーケンシャルアクセスができるようにするためである．

4.3 PSA

4.3.1 PSA における特徴量

PSA ではシーケンスを特定の長さのセグメントに分割し，セグメントにおける平均と標準偏差を特徴量として距離の近似計算を行う．PSA を以下のように定義する．

定義1 (PSA) s_iをシーケンスX（長さn）をN個の特定の長さのセグメントに分割したときのi番目のセグメントとする．l_iをs_iの長さとし，ϕ_iをs_iの平均とし，σ_iをs_iの標準偏差としたとき，シーケンスXの PSA における特徴量を$\hat{X}=\left(\langle l_1,\phi_1,\sigma_1\rangle,\langle l_2,\phi_2,\sigma_2\rangle,\cdots,\langle l_N,\phi_N,\sigma_N\rangle\right)$と３つの係数のタプルとして定義する．

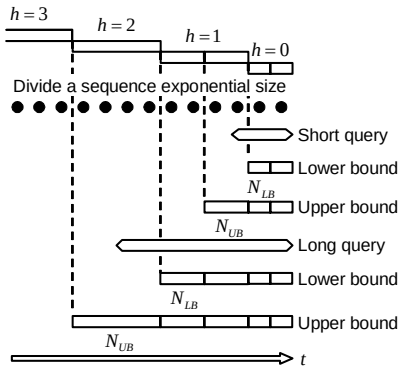

図 2 シーケンスの分割
Fig.2 Sequence segmentation

ここで$n=\sum_{i=1}^{N} l_i$である．シーケンスXにおいてセグメントs_iの開始点を$p_i(1\le i\le n)$とすると，$\phi_i = 1/l_i \cdot \sum_{j=p_i}^{p_{i+1}-1} x_j$，$\sigma_i = \sqrt{1/l_i \cdot \sum_{j=p_i}^{p_{i+1}-1} x_j^2 - \phi_i^2}$ と計算する．

4.3.2 PSA の性質

PSA では設定する距離関数により距離の下限値と上限値を計算できるので，探索漏れも過剰探索も発生しない．

探索漏れが発生しないことを保証する補助定理として lower bounding lemma [1]が知られている．lower bounding lemma とは近似後のシーケンスの距離を$L(\hat{X}_l,\hat{Y}_l)$としたときに，$L(\hat{X}_l,\hat{Y}_l)\le D(X_l,Y_l)$が成り立てば探索漏れが発生しないという補助定理である．すなわち距離の下限値を用いることにより探索漏れなくシーケンスの類似判断ができる．

また過剰探索が発生しないことを保証する補助定理として新たに upper bounding lemma を示す．

補助定理1 (upper bounding lemma) 近似後のシーケンスの距離を$U(\hat{X}_l,\hat{Y}_l)$としたときに，$U(\hat{X}_l,\hat{Y}_l)\ge D(X_l,Y_l)$の条件が成り立つことが過剰探索が発生しないことの十分条件である．

すなわち距離の上限値を用いることにより過剰探索なくシーケンスの類似判断ができる．

4.3.3 下限値と上限値の計算方法

PSA では図2に示す通り，下限値は問い合わせシーケンスより短いシーケンスから計算し，上限値は問い合わせシーケンスより長いシーケンスから計算する．シーケンスは複数のセグメントを集約して構成する．下限値の計算に用いるセグメントのなかで最も番号の小さいものをN_{LB}，上限値の計算に用いるセグメントのなかで最も番号の小さいものをN_{UB}とすると，$N_{LB}=\min\left(j\,|\,\sum_{i=j}^{N} l_i \le l\right)$，$N_{UB}=\max\left(j\,|\,\sum_{i=j}^{N} l_i \ge l\right)$として求める．

下限値を計算するときの距離関数$L(\hat{X}_l,\hat{Y}_l)$と上限値を計算するときの$U(\hat{X}_l,\hat{Y}_l)$は以下のように定義する．

定義2 (距離関数 lower bound) $L(\hat{X}_l,\hat{Y}_l)$を以下のように定義する．

$$L\left(\hat{X}_l,\hat{Y}_l\right)=\sqrt{\sum_{i=N_{LB}}^{N} l_i\left\{(\phi_i^X-\phi_i^Y)^2+(\sigma_i^X-\sigma_i^Y)^2\right\}} \quad (2)$$

定義3 (距離関数 upper bound) $U(\hat{X}_l,\hat{Y}_l)$を以下のように定

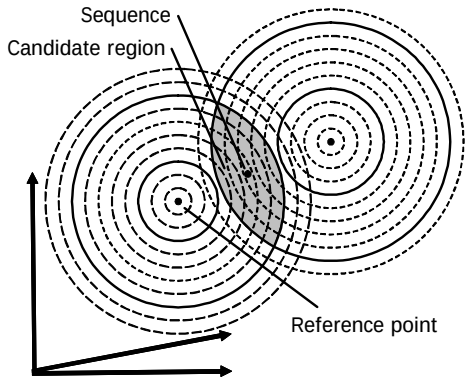

図3 マトリクス
Fig.3 Matrix

義する．

$$U(\hat{X}_l,\hat{Y}_l)=\sqrt{\sum_{i=N_{UB}}^{N} l_i\{(\phi_i^X-\phi_i^Y)^2+(\sigma_i^X+\sigma_i^Y)^2\}} \quad (3)$$

4.3.4 セグメント長さ

セグメントの長さを決定するために処理時刻からの経過時間によってセグメントのレベルわけを行う．図2に示すようにセグメントs_iの長さl_iはレベルhに依存し，hが大きくなるに従ってl_iも大きくなる．レベルhのセグメントの長さは2^hである．このようにするのは任意の長さの問い合わせシーケンスに対応するためである．すなわち問い合わせシーケンスが短い場合も長い場合も PSA で下限値と上限値を計算したときの相対的な誤差は同等になる．

特徴量の更新は各レベルhのセグメントの個数がCより大きくなるときに結合することでインクリメンタルに行う．$C=2$であるときの具体的な更新方法を説明する．c_iをレベルiにおけるセグメントの個数とする．更新前のセグメントは$c_0=2$，$c_1=2$，$c_2=1$である．ここでシーケンスデータx_{n+1}が流入してくると$c_0=3$となる．$C=2$であるので，レベル0のはじめの2つのセグメントを結合してレベル1のセグメントを作成する．すると$c_1=3$となるので，同様に処理する．結果として更新後のセグメントは$c_0=1$，$c_1=1$，$c_2=2$となる．セグメントs_iとs_{i+1}を結合してs_i'するときは，$l_i'=2l_i$，$\phi_i'=(\phi_i+\phi_{i+1})/2$，$\sigma_i'=\sqrt{(\sigma_i^2+\sigma_{i+1}^2)/2+(\phi_i-\phi_{i+1})^2/4}$のように更新する．

4.4 マトリクス

4.4.1 データ構造

マトリクスは近似後の多次元空間を同心超球構造に分割し，シーケンスに対して領域番号を付与する手法である．本手法では図 3 に示すように，PSA で近似後の多次元空間上に複数の基準点$O_i(1\le i\le K)$を設定する．各シーケンスには領域番号$R(\hat{X}_l,O_i)$を付与する．領域情報はシーケンスと基準点の距離を刻み幅$\varepsilon/\omega(\omega>0)$で分割したものであり，$R(\hat{X}_l,O_i)=\lfloor\omega\cdot L(\hat{X}_l,O_i)/\varepsilon\rfloor$として計算する．マトリクスのデータ構造は基準点$O_i$とシーケンス$\hat{X}_l$の行列として表現される．

4.4.2 探索処理

探索処理ではまずマトリクスの要素の計算をする．はじめに基準点をランダムに決定する．そして各シーケンスと基準点の距離を計算してマトリクスの要素を計算する．次に探索処理ではマトリクスを用いて類似シーケンスの候補を探索する．具体的にはデータシーケンスごとにすべての基準点において$|R(\hat{X}_l,O_i)-R(\hat{Y}_l,O_i)|<\omega+1$が成り立つか調べる．ひとつの基準点においてにでも成立しなければ非類似とする．本手法では探索漏れなく類似するシーケンスを求めることができる．

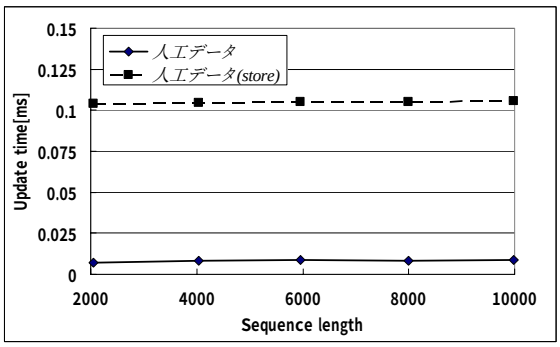

図4 更新時間
Fig.4 Wall clock time for update processing

4.5 処理概要

4.5.1 更新処理

更新処理ではデータを受信する度に PSA における特徴量を更新する．シーケンスデータは一定時間Tごとに可逆圧縮を用いてディスクに圧縮格納する．

4.5.2 探索処理

探索処理ではその都度マトリクスを構築する．次にマトリクスを用いて類似シーケンスの候補を絞り込む．絞り込まれたシーケンスの組み合わせに対して PSA の特徴量から距離の下限値と上限値を計算して類似判断をする．下限値と上限値を用いて類似判断ができない場合，すなわち距離の下限値が閾値以下でありかつ距離の上限値が閾値より大きい場合はディスクに格納されたシーケンスを解凍参照し，正確な距離を計算し類似判断を行う．

4.6 メモリ使用量

DAPSS におけるメモリ量は$O(m\cdot\log(n)+l)$になる．これはナイーブな手法におけるメモリ使用量$O(m\cdot n)$と比較して省メモリであることがわかる．

5. 評価実験

実験には人工データを用いた．また実験におけるパラメータは可逆圧縮において保存するシーケンスの長さ$T=64$，PSA におけるセグメントのキャパシティ$C=25$，マトリクスにおける同心超球数$K=10$，分割幅$\omega=10$とした．また閾値εは問い合わせシーケンスの長さlによって変化させ，$\varepsilon=0.02\cdot l$とした．実験は CPU が Pentium4 の 3.2GHz，メインメモリが 1GB のマシンで行った．

5.1 更新時間

図4を見ると更新時間はシーケンスの長さnに対してほとんど変わらないことがわかる．これはシーケンスが長くなっても PSA の特徴量の更新において結合するセグメントの個数がほとんど変わらないためである．また図4より格納処理を行うと大幅に更新時間がかかることがわかる．これはディクスのアクセスには非常に時間がかかるためである．しかしディスクのアクセスは一定時間に一度のみしか行わな

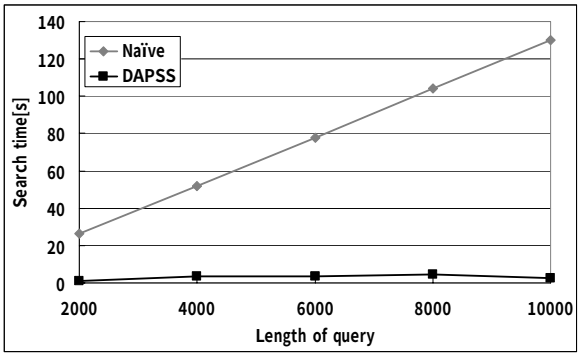

図 5　シーケンス長を変化させたときの探索時間
Fig.5 Wall clock time versus sequence length

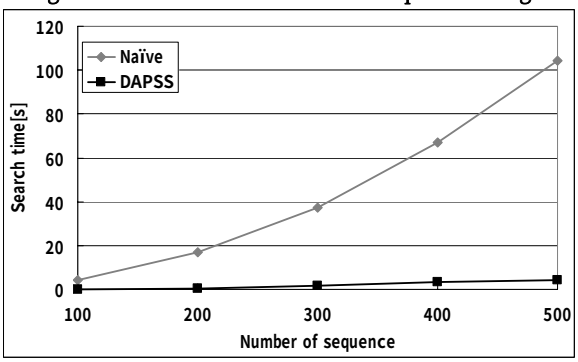

図 6　シーケンス数を変化させたときの探索時間
Fig.6 Wall clock time versus the number of sequences

いので問題にならない.

5.2　探索時間

5.2.1　シーケンスの長さを変えた場合

探索時間の実験結果を図5に示す．なお実験ではシーケンスの数 $m = 500$ とした.

結果を見ると DAPSS はナイーブな手法と比較して 15～55 倍と高速な探索が行えることがわかる．ナイーブな手法では探索時間はシーケンスの長さ n が増加すると $O(n)$ で増加する．しかし DAPSS ではシーケンスの長さが増加しても $O\log(n)$ 個の特徴量を用いて計算を行うため，シーケンスの長さの増加の影響は抑えられ，探索時間の大幅な低減を達成している.

5.2.2　シーケンスの数を変えた場合

実験結果を図 6 に示す. なお実験では問い合わせシーケンスの長さ $l = 8000$ とした.

結果を見ると DAPSS はナイーブな手法と比較して 20～45 倍と高速な探索が行えることがわかる．ナイーブな手法では探索時間はシーケンスの数 m が増加すると $O(m^2)$ で増加する．しかし DAPSS ではマトリクスによって類似シーケンスの候補を絞り込んでいるためシーケンスの数の増加の影響は抑えられ，探索時間の大幅な低減を達成している.

6．まとめ

本研究では流入する複数の任意長のシーケンスのうち，類似した組み合わせを探索する問題について取り組んだ．本研究では DAPSS を用いることによってこの処理を高速，正確，省メモリで行えることを示した.

本研究では３つの手法を提案した. 可逆圧縮は流入するシーケンスを高速かつ小容量で格納できる．PSA はシーケンスが長くなっても高速にシーケンスの距離を計算できる．マトリクスはシーケンスが多くなっても高速に類似シーケンスの候補を絞り込める.

本研究では提案手法を用いて検証し，ナイーブな手法より数十倍高速に処理が行えることを確認した.

[文献]

[1] R.Agrawal, C. Faloutsos and A. N. Swami.: “Efficient Similarity Search In Sequence Databases”, In FODO, 1993.
[2] B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom.: ” Models and Issues in Data Stream Systems”, In PODS, 2002.
[3] S. Babu, J. Widom.: “Continuous Queries over Data Streams”, SIGMOD Record, 2001
[4] H. Balakrishnan, M. Balazinska, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, E. F. Galvez, J, Salz, M. Stonebraker, N. Tatbul, R. Tibbetts, S. B. Zdonik.: ”Retrospective on Aurora”, VLDB J., 2004.
[5] A. Bulut , A. K. Singh.: “A Unified Framework for Monitoring Data Streams in Real Time”, In ICDE, 2005
[6] J. Chen, D. J. DeWitt, F. Tian, Y. Wang.: “NiagaraCQ: A Scalable Continuous Query System for Internet Databases”. In SIGMOD, 2000.
[7] C. Faloutsos, M. Ranganathan and Y. Manolopoulos.: “Fast Subsequence Matching in Time-Series Databases”, In SIGMOD, 1994.
[8] L. Golab, M. Tamer Ozsu.: “Issues in data stream management”. SIGMOD Record, 2003.
[9] D.A.Huffman.: “A method for the construction of minimum redundancy codes”. In IRE, 1952.
[10] Y. Zhu, D. Shasha.: “StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time”. In VLDB, 2002.
[11] J. Ziv, A. Lempel.: “A Universal Algorithm for Sequential Data Compression”. IEEE Transactions on Information Theory, 1977.
[12] 藤原靖宏，櫻井保志，山室雅司.: “大量なデータストリームの類似検索手法”. DBWeb, 2005.

藤原 靖宏　Yasuhiro FUJIWARA
2003 年 早稲田大学大学院理工学研究科電気工学専攻修了. 日本電信電話（株）入社．時系列データ処理の研究開発に従事．電子情報通信学会，日本データベース学会各会員.

櫻井 保志　Yasushi SAKURAI
1991 年 同志社大学工学部電気工学科卒業．日本電信電話（株）入社．1996 年 奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．2004 年～2005 年カーネギーメロン大学客員研究員. 索引技術, 情報検索の研究開発に従事. 博士（工学）．ACM，情報処理学会，日本データベース学会各会員.

山室 雅司　Masashi Yamamuro
1987 年 早稲田大学大学院理工学研究科数学科専攻修了. 日本電信電話（株）入社．1990 年コロンビア大学大学院電気工学専攻修了．ディジタル情報流通，データベース設計法の研究開発に従事．博士（工学）．IEEE-CS，電子情報通信学会，日本ソフトウェア学会，情報処理学会，日本データベース学会各会員.