「文字列の方程式を解く」

「協調と制御」領域：篠原　歩

要　旨

文字列は、情報の格納・伝達のための最も基本的な型の一つであり、効率のよい文字列処理は、大量のテキストデータを扱うために必須の要素技術である。本研究では、パターン照合やデータ圧縮、複数のテキストに共通に現れるパターンの発見など、さまざまな文字列処理に対して、数値演算やフーリエ変換などの数値的な手法を活用した高速化の研究を行った。また、文字列方程式の解の上限を数学的に厳密に証明した。

1．研究のねらい

インターネットの発展を背景として膨大なデータが蓄積されているが、特にXMLフォーマットの導入は、あらゆる情報を計算機と人間の双方に可読な文字列として表現することによって協調的な知的生産性を高めようとするものであり、大量のテキストデータを効率よく処理するための技術はますます重要になっている。パターン照合をはじめとした文字列処理に関する研究は、これまでも深く研究されてきたが、そこで用いられる手法は文字列としての性質を巧みに利用して文字列処理に特化されたものであり、他方で膨大な研究の蓄積がなされている数値演算とはほとんど無関係である。テキストデータの分類や、大量のテキストデータに潜む規則の発見など、知的な文字列処理を行うための操作の大部分は計算量的に困難な問題を内包しており、これらを現実的な時間で近似していくためには、「文字列」に対する新しい視点が必要であると考える。

本研究は、文字列として与えられる情報を、そのままの離散的な値としてではなく、連続的な数値としてとらえるための枠組みを開発し、その上でさまざまな文字列操作を数値演算として実装できるようにするための基礎理論の構築を目的としている。

2．研究経緯と成果

2.1 高速フーリエ変換による近似文字列照合の高速化

文字列照合問題は、$T=t_1, \cdots, t_n$と$P=p_1, \cdots, p_m$をそれぞれテキストとパターンと呼ばれるアルファベットΣ上の文字列として、テキストTに現れるパターンPの出現位置を全て見つける問題として定式化できる。パターンとテキストの間の少数の不一致を許した近似文字列照合問題は、実用的な観点から極めて重要であり、本質的にはTとPの間のスコアベクトル$C(T, P)=(c_1, \cdots, c_{n-m+1})$を求める問題と見なすことができる。ここに

■ **篠原　歩**　（九州大学大学院システム情報科学研究院）
グループメンバー：馬場謙介、稲永俊介、Hyyro, Heikki Matti
リサーチスタッフ：馬場謙介、稲永俊介、井上　淳
（研究者プロフィール）　1965年福岡県生まれ。90年九州大学大学院総合理工学研究科情報システム学専攻修士課程修了。94年博士（理学）取得。90年九州大学理学部附属基礎情報学研究施設助手、94年同助教授、96年より九州大学大学院システム情報科学研究科助教授、現在に至る。機械学習、発見科学、文字列処理、アルゴリズムと計算量理論、バイオインフォマティクスの研究に従事。

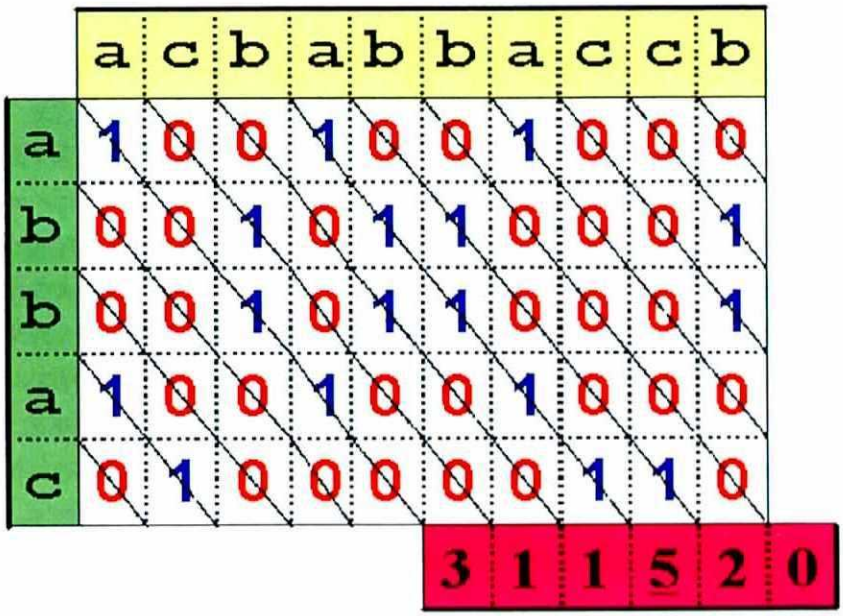

図１　テキスト acbabbaccb とパターン abbac のスコアベクトル(3, 1, 1, 5, 2, 0).

c_iは、Tの部分文字列 $t_i, \cdots, t_{i+m-1}$と Pの間の一致の数であり、$c_i=m$のとき、テキスト中の i 番目の位置にパターンそのものが現れることを意味する。図１に示すとおり、これはテキストとパターンの一致する文字の個数を対角線上に集計していけば得られるものであり、したがって素朴なアルゴリズムを用いると計算時間は O(mn)となる。通常の厳密な文字列照合は O($m+n$)時間で行えるため、この計算量を落とすことが課題となる。スコアベクトルの計算は、テキストとパターンに対応する２つの関数の畳み込みとみなせるので、デジタル信号処理の分野で行われているように、高速フーリエ変換(FFT)を応用した高速化が見込まれる。実際、文字列のアルファベットがΣ={a, b}のとき、それぞれを{-1, 1}に対応させ、まさに文字列を数値列かのように取り扱うことで、基本的には全く同じように FFT を用いてスコアベクトルを求めることができる。しかしながら、アルファベットサイズσ=|Σ|が３以上のときは、文字列と数値列との本質的な差異が顕在化し、適用が不可能となる。つまり、数値としての演算と、文字比較としての演算に、うまい対応関係が取れなくなってしまう。この困難を克服するために、Atallah は 2001 年にモンテカルロ型確率アルゴリズムを提案した。これは次の２つのアイディアに基づく。(1) σ種類の文字を、１のσ乗根である複素数に対応させるすべての写像を考え、それぞれの対応のもとで得られるスコアベクトルの平均を考える。個々の写像における不具合が、全体の平均をとることでちょうど相殺されるようになっており、こうして得られる平均は、求めるべきスコアベクトルと正確に一致する。(2)しかしながら、すべての写像を実際に計算するには、σ^n通りの可能性を考えなければならない。そこで、正確な値を計算する代わりに、ランダムに選んだ k 個の写像のみを考え、その平均によってスコアベクトルの推定を行う。この期待値は求めるスコアベクトルに等しく、分散が $(m-c_i)^2/k$で抑えられることが示されており、FFT を用いることで計算時間は$O(kn\log m)$ となる。本研究では、このアルゴリズムをより洗練させ、σ種類の複素数の代わりに、{-1, 1}を考えるだけで同等の性能を得られることを示し、さらに推定値の分散を厳密に解析することに成功した。つまり、文字を数値に置き換える際の写像の数をσ^nから 2^nに減らせることを示した。このように、数値処理でよく用いられる FFT の応用によって、近似文字列照合を効率よく行うことが原理的に可能であることを証明した。

2.2 一変数の文字列方程式の解の上限

数値的な手法において、与えられた制約を等式や不等式の形で表現し、それを満たす解を探すというのは最も基本的である。また、観測データをうまく説明する法則を微分方程式という形で見つけ出すこともよく行われている。同様に、文字列処理の最も基本的な操作である、文字列照合問題を文字列に関する方程式とみなすこともできる。例えば、テキスト文字列 abababaabbababa の中に出現するパターン aab の位置を探す問題は、 XaabY = abababaabbababa という文字列方程式を満たす変数代入 (X:=abab, Y:=bababa)を見つける問題とみなせる。より一般的には、X aabY = aYabbaba のように、両辺に変数を含むような方程式を解くことを考える（この場合は X:=ababa, Y:=baba という解がある）。このような問題を解く作業は、一見、容易そうに見えるが、実は決してそうではない。歴史的にみても、文字列方程式を解く問題が計算可能であることが知られたのは 1977 年になってからであり、その計算量を落とす努力が実を結び始めたのは 1990 年代に入ってからである。難しさの本質は、解となる変数代入の長さを、文字列方程式の形から見積もるのが容易ではないところにある。例えば、文字列方程式 Xba = abX を満たす解は、X=a, aba, ababa, abababa,... のように無限個存在する。長さが最小になるような解だけを考えるとしても、例えば方程式 YYXbaaaaZ = ZYYXaaaab の最小解は、

X := aaaabaaaa,

Y := aaaabaaaaaaaabaaaabaaaa,

Z := aaaabaaaaaaaabaaaabaaaaaaaabaaaaaaaab

であり、問題の長さに比べて極めて長い。一方、文字列方程式 XabYY = bZZYX は解なしであり、文字列方程式を解くアルゴリズムを開発する際には、どの長さまで調べればその問題には解がないと言えるのかを知る必要がある。このように、解が存在するときにその最小解の上限を求めることが、そのまま最悪時の計算時間の評価に直結する。一般の文字列方程式に対する最小解の上限について、まだ十分にその挙動が解析されていない。そこで、本研究では、変数が１種類だけ使われているような文字列方程式（例えば XabXbaaaaaa=aaaaaabaXbX）に限定して、最小解の長さの上限を求めることにした。既存研究においては、方程式の長さの４倍程度でおさえられるだろうという予測はなされていたが、その厳密な値は知られていなかった。我々は、網羅的に１変数文字列方程式を枚挙しながらその解を求めるプログラムを作成して長時間動作させ、その結果を分類しながら問題の分析を行った。その過程の中で、文字列の繰り返し構造のもつ性質に着目することによって、最終的には解の上限を厳密に、しかも極めて単純な式によって表すことに成功し、数学的な証明を与えた。また逆に、任意に指定された長さに対して、この上限と一致する問題を構成的に与えるアルゴリズムも示した。これらの論理的帰結として、１変数文字列方程式は、解をもつとすれば、その最小解は高々、問題の長さで抑えられることになる。このことにより、実用的な観点から、１変数文字列方程式を解くアルゴリズムの高速化が行えることがわかった。

2.3 高速パターン発見アルゴリズム

数値的に得られた観測データから、そこに内在する関係を関数として抽出する手法は、補間や内挿、外挿として知られ、数値データ処理の基本として、さまざまな関数族に対するアルゴリズムが開発されて、実用に供している。近年、HTML や XML のような半構造を持ったテキストデータや、DNA 配列、アミノ酸配列のようにほとんど構造の知られていないテキストデータが大量に蓄積され、利用可能になってきたことをうけて、これらのデータベースから、そこに内在する関係を文字列パターンとして発見するための手法の開発が望まれている。本研究では、この問題を、２つの文字列集合を与えられたときに、それを高精度で分離する文字列パターンを高速に見つける最適化問題として定式化し、そのために有用なデータ構造とアルゴリズムを与

えた。

まず、探索アルゴリズムに関しては、部分文字列パターン、部分列パターン、変数を含むパターン等、さまざまなパターン族に対して、その形式言語としての特徴を生かした探索空間の枝刈り技法を取り入れることにより、ユーザの指定するスコア関数を最適化するパターンを実用的な時間で発見できるアルゴリズムの開発に成功した。

また、この中で頻繁に用いられるパターン照合を高速に行うための索引構造として、有効無閉路文字列グラフ(DAWG)、有向無閉路部分列グラフ (DASG) 等を対象として、その性質や高速な構築アルゴリズムを開発した。さらに、文字列から索引構造を構築する代わりに、索引構造から文字列を再構成する新しい問題について詳しい考察を行った。すなわち、既存の問題の逆問題として、有効無閉路文字列グラフ、有向無閉路部分列グラフ、および接尾辞配列の３つの有用な索引構造に対して、そのグラフ構造から文字列を線形時間で推定するアルゴリズムを開発することに成功した。このことにより、文字列とその索引構造の間の関係に関するより深い知見を得た。

３． 今後の展望

以上に述べた研究成果をもとにして、現在、下記のような研究を進めている。

- 文字列方程式に関して、現時点では１変数の文字列方程式に対する線形時間のアルゴリズムは知られていない。また、２変数の文字列方程式に対する既存の最良のアルゴリズムは$O(n^3)$時間である。本稿で述べた、解の長さの上限を利用することによって、これらを改良することを検討している。また実用面からの高速化のために、パターン発見アルゴリズムで用いたような探索空間の枝刈り手法の開発を行う。
- パターン発見アルゴリズムに関して、さらに複数のパターンの組み合わせや、不一致を許した近似パターンの発見等への一般化を行う。また、ここで開発してきた手法を逆に数値列パターンの発見へ応用する可能性を探る。
- ロボットの協調制御への応用を行う。ここでは、さまざまな抽象レベルにおいて数値列パターンの発見と高速処理が重要な要素技術となる。例えば、４足ロボットの制御に関しては、12 個の関節に対する角度を表す数値列を系統的に発生させる必要があり、高速な歩行のためには、実機を用いた長時間の試行錯誤が必要となる。その作業を効率化するために、蓄積した数値列データから、よい歩行のパターンを見つけ出し、それを改良していくという手法が可能であると考えている。また、複数のロボットの協調的な動作のために、各ロボットの動作の列をデータベースに蓄積し、それを解析することによって、目標達成に有効な動作パターンを抽出することを目指す。
- 文字列を対象として得られた本研究のさまざまな手法を、木構造やグラフ構造などへさらに一般化する。

発表リスト（論文、口頭発表、出版物）

国際・口頭発表

[1] Shunsuke Inenaga, Masayuki Takeda, Ayumi Shinohara, Hiromasa Hoshino, and Setsuo Arikawa, "The Minimum DAWG for All Suffixes of a String and its Applications", Proc. 13th Ann. Symp. on Combinatorial Pattern Matching (CPM2002), Lecture Notes in Computer Science 2373, pp. 153-167, Springer-Verlag, July 2002.

[2] Kensuke Baba, Ayumi Shinohara, Masayuki Takeda, Shunsuke Inenaga, and Setsuo Arikawa,
"A Note on Randomized Algorithm for String Matching with Mismatches",
Proc. The Prague Stringology Conference '02 (PSC'02), pp. 9-17, Czech Technical University, September 2002.

[3] Shunsuke Inenaga, Ayumi Shinohara, Masayuki Takeda, and Setsuo Arikawa,
"Compact Directed Acyclic Word Graphs for a Sliding Window",
Proc. 9th International Symposium on String Processing and Information Retrieval (SPIRE2002), Lecture Notes in Computer Science 2476, pp. 310-324, Springer-Verlag, September 2002.

[4] Shunsuke Inenaga, Ayumi Shinohara, Masayuki Takeda, Hideo Bannai, and Setsuo Arikawa,
"Space-Economical Construction of Index Structures for All Suffixes of a String",
Proc. 27th Inter. Symp. on Mathematical Foundation of Computer Science (MFCS2002), Lecture Notes in Computer Science 2420, pp. 341-352, Springer-Verlag, August 2002.

[5] Shunsuke Inenaga, Ayumi Shinohara,
"Bidirectional Construction of Suffix Trees"
Prague Stringology Conference 2002 (PSC'02) pp.75-87, September 2002.

[6] Shunsuke Inenaga, Hideo Bannai, Ayumi Shinohara, Masayuki Takeda, and Setsuo Arikawa,
"Discovering Best Variable-Length- Don't-Care Patterns",
Proc. 5th International Conference on Discovery Science (DS2002), Lecture Notes in Computer Science 2534, pp. 86-97, Springer-Verlag, November 2002.

[7] Satoru Miyamoto, Shunsuke Inenaga, Masayuki Takeda, and Ayumi Shinohara,
"Ternary Directed Acyclic Word Graphs",
Proc. Eighth International Conference on Implementation and Application of Automata (CIAA2003), Lecture Notes in Computer Science 2759, pp. 120-130, Springer-Verlag, July 2003.

[8] Hideo Bannai, Shunsuke Inenaga, Ayumi Shinohara, and Masayuki Takeda,
"Inferring Strings from Graphs and Arrays",
Proc. 28th International Symposium on Mathematical Foundations of Computer Science (MFCS2003), Lecture Notes in Computer Science 2747, pp. 208-217, Springer-Verlag, August 2003.

[9] Kensuke Baba, Satoshi Tsuruta, Ayumi Shinohara, and Masayuki Takeda
"On the Length of the Minimum Solution of Word Equations in One Variable",
Proc. 28th International Symposium on Mathematical Foundations of Computer Science (MFCS2003), Lecture Notes in Computer Science 2747, pp. 189-197, Springer-Verlag, August 2003.

[10] Tetsuya Nakatoh, Kensuke Baba, Daisuke Ikeda, Yasuhiro Yamada, and Sachio Hirokawa
"An Efficient Mapping for Scores of String Matching"
Proc. Prague Stringology Conference '03 (PSC '03), pp.127-136, Czech Technical University, 2003.

[11] Shunsuke Inenaga, Takashi Funamoto, Masayuki Takeda, and Ayumi Shinohara,
"Linear-Time Off-Line Text Compression by Longest-First Substitution",
Proc. 10th International Symposium on String Processing and Information Retrieval (SPIRE 2003), Lecture Notes in Computer Science 2857, pp. 137-152, Springer-Verlag, October 2003.

[12] Zdenek Tronicek and Ayumi Shinohara,
"The Size of Subsequence Automaton",
Proc. 10th International Symposium on String Processing and Information Retrieval (SPIRE 2003), Lecture Notes in Computer Science 2857, pp. 304-310, Springer-Verlag, October 2003.

[13] Kensuke Baba, Yoshihito Tanaka, Tetsuya Nakatoh, and Ayumi Shinohara
"A Generalization of FFT Algorithms for String Matching",

Proc. International Symposium on Information Science and Electrical Engineering 2003, pp. 191-194, November 2003.

[14] Heikki Hyyrö, Kimmo Fredriksson, Gonzalo Navarro,
"Increased Bit-Parallelism for Approximate String Matching"
Proc. The Third International Workshop on Experimental and Efficient Algorithms (WEA2004), Lecture Notes in Computer Science 3059, Springer-Verlag, May, 2004.

[15] Heikki Hyyrö, Ayumi Shinohara
"Bit-Parallel LCS-length Computation Revisited"
Proc. 15th Australasian Workshop on Combinatorial Algorithms (AWOCA2004), July, 2004.

[16] Hideo Bannai, Heikki Hyyrö, Ayumi Shinohara, Masayuki Takeda, Kenta Nakai, Satoru Miyano
"Finding Optimal Pairs of Patterns"
Proc. 12th International Conference on Intelligent Systems for Molecular Biology ／3rd European Conference on Computational Biology (ISMB/ECCB2004), July, 2004

[17] Heikki Hyyrö,
"A note on bit-parallel alignment computation"
Proc. Prague Stringology Conference '04 (PSC'04), pp.79-87, August, 2004.

[18] Hideo Bannai, Heikki Hyyrö, Ayumi Shinohara, Masayuki Takeda, Kenta Nakai, Satoru Miyano
"Finding Optimal Pairs of Patterns"
Proc. 4th Workshop on Algorithms in Bioinformatics (WABI2004), pp. 450-462, September, 2004.

[19] Shunsuke Inenaga, Hideo Bannai, Heikki Hyyrö, Ayumi Shinohara, Masayuki Takeda, Kenta Nakai, Satoru Miyano
"Finding Optimal Pairs of Cooperative and Competing Patterns with Bounded Distance"
Proc. The 7th International Conference on Discovery Science (DS2004), pp.32-46, October, 2004.

[20] Heikki Hyyrö,
"An Improvement and an Extension on the Hybrid Index for Approximate String Matching"
Proc. The Eleventh Symposium on String Processing and Information Retrieval (SPIRE2004), pp.208-209, October, 2004.

[21] Heikki Hyyrö, Jun Takaba, Ayumi Shinohara, Masayuki Takeda
"On Bit-Parallel Processing of Multi-byte Strings"
Proc. The first Asia Information Retrieval Symposium (AIRS2004), October, 2004.

[22] Shunsuke Inenaga, Ayumi Shinohara and Masayuki Takeda.
"An efficient pattern matching algorithm on a subclass of context free grammars",
Proc. Eighth International Conference on Developments in Language Theory (DLT'04), Lecture Notes in Computer Science 3340, pp. 225-236, Springer-Verlag, December, 2004.

[23] Heikki Hyyrö, Yoan Pinzon and Ayumi Shinohara.
"Fast Bit-Vector Algorithms for Approximate String Matching under Indel-Distance"
Proc. The 31st Annual Conference on Current Trends in Theory and Practice of Informatics (SOFSEM 2005), Lecture Notes in Computer Science, Springer-Verlag, January, 2005.

国際・論文発表

[1] Kensuke Baba, Ayumi Shinohara, Masayuki Takeda, Shunsuke Inenaga, and Setsuo Arikawa
"A Note on Randomized Algorithm for String Matching with Mismatches",
Nordic Journal of Computing, Vol. 10, pp. 2-10, 2003.

国内・口頭発表

[1] 馬場謙介、篠原歩、竹田正幸、稲永俊介、有川節夫
"近似文字列照合のための確率アルゴリズム", LA シンポジウム, 2002.

[2] 中藤哲也、馬場謙介、池田大輔、山田泰寛、廣川佐千男、篠原歩
"近似文字列照合のための決定性アルゴリズム"
第１４回データ工学ワークショップ, 2003.

国内・論文発表

[1] 中藤哲也, 馬場謙介
"近似文字列照合のための効率的なアルゴリズム"
日本データベース学会 Letters, vol.2(1), pp.87-90, 日本データベース学会, 2003.