情報処理学会論文誌　データベース　Vol. 2　No. 4　24–31 (Dec. 2009)

不一致を許す文字列照合のための FFT を用いた確率的アルゴリズムの精度評価

中藤哲也[†1]　馬場謙介[†2]　池田大輔[†3]
森　雅生[†4]　廣川佐千男[†1]

テキスト中から与えられたパターンを見つけ出す文字列照合問題は，Web の情報検索や DNA 配列の特定パターンの検索に用いられるなど，幅広い応用範囲を持つ．パターンの編集に置換のみを許した近似文字列照合は，不一致を許す文字列照合と呼ばれ，テキスト全域での一致スコアを求めるために，正確な一致場所を求める文字列照合よりも計算量が大きい．この問題の解法として，高速フーリエ変換（FFT）を利用した高速な確率的アルゴリズムがいくつか提案されており，それらは文字から数値への写像の生成方法により，写像の総数と，得られる推定値の精度が異なる．我々の提案するアルゴリズム[10] は写像の総数が理論上での最小であり，精度も提案されているアルゴリズム中で最も高い．本稿では，Atallah らのアルゴリズム[1] による推定値の精度と実験的な比較を行い，提案アルゴリズムの推定値の精度がより高いことを確認した．

Accuracy Evaluation of FFT-based Randomized Algorithms for String Matching with Mismatches

Tetsuya Nakatoh,[†1] Kensuke Baba,[†2]
Daisuke Ikeda,[†3] Masao Mori[†4]
and Sachio Hirokawa[†1]

String matching is the problem of finding all occurrences of a given pattern string in a given text string. It is applicable to a wide range of fields, such as Web information retrieval and pattern discovery of DNA sequences. The string matching with mismatches allows inexact match with substitution and has high complexity. In order to solve the problem several fast randomized algorithms have been proposed. They use the fast Fourier transformation (FFT). All of these algorithms introduce a certain number of mappings that convert symbols into numbers. The total number of such mappings and variance of estimates depends on the method to generate the mappings. This paper proposes an algorithm that achieves the theoretically minimum number of mappings and yields accurate estimates. Empirical evaluation is conducted to compare the accuracy of estimates of the proposed algorithm with that of Atallah et al. It is confirmed that the accuracy of the proposed algorithm is better.

1. はじめに

文字列照合問題[4),5),7)] は，与えられた長い文字列 $T = t_1 \cdots t_n$（テキストと呼ぶ）と短い文字列 $P = p_1 \cdots p_m$（パターンと呼ぶ）をアルファベット Σ 上の文字列とするとき，テキスト T に現れるパターン P の出現位置をすべて見つける問題である．この問題の解は $O(n)$ で得られることが知られている．これに対し，編集に置換のみを許した**不一致を許す文字列照合問題**があり，その距離はハミング距離として定義される．パターン長とハミング距離の差がマッチングのスコアとなる．スコアはテキスト T 上のすべての位置で得られるので，この問題はテキスト T 上のすべての位置におけるパターン P とのマッチングのスコアのベクトル $C(T, P) = (c_1, \ldots, c_{n-m+1})$ を求める問題と見なすことができる．ここで各 c_i は，T の部分文字列 $t_i \cdots t_{i+m-1}$ と P の間のシンボルの一致の数であり，$c_i = m$ は，テキスト中の i 番目の位置にパターンそのものが現れている場合である．

図 1 はスコアベクトルの例である．このスコアベクトルを求めるには，たとえば単純にシンボルの比較を m 回ずつ $n - m + 1$ 個の i について行えばよく，この素朴なアルゴリズムの計算量は $O(mn)$ であることが容易に分かる．Fischer ら[6] は，文字列の比較に畳み込みが使えることを見いだした．その原理に基づき高速フーリエ変換（FFT）を用いた計算量 $O(|\Sigma| n \log m)$ のアルゴリズム[7] が示されている．また，このアルゴリズムの改良として，解の推定値を求める確率的アルゴリズム[1)–3),8),9)] が提案されている．それらのアルゴリズムでは，文字を数値に変換する写像集合から，k 個のサンプルを取り出して計算することで計算量を $O(kn \log m)$ に抑えている．

†1 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University
†2 九州大学附属図書館
Kyushu University Library
†3 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University
†4 九州大学大学評価情報室
Institutional Research Office, Kyushu University

位置	1	2	3	4	5	6	7	8	9	10
テキスト	a	c	b	a	b	b	a	c	c	b
パターン	a	b	b	a	c					
		a	b	b	a	c				
			a	b	b	a	c			
				a	b	b	a	c		
					a	b	b	a	c	
						a	b	b	a	c
スコアベクトル	3	1	1	5	2	0				

図 1　スコアベクトルの例
Fig. 1　An example of score vector.

本稿では，我々の提案する最適な写像生成を用いた確率的アルゴリズム[10] を実装し，その推定値の精度について従来アルゴリズムとの実験的な比較を行う．

2. 準　　備

Σ を有限の**アルファベット**とし，Σ^* の要素を**文字列**と呼ぶ．文字列 w の長さを $|w|$ で表す．また，集合 S の要素数を $|S|$ で表す．$\sigma = |\Sigma|$ とする．

関数 $\delta : \Sigma \times \Sigma \to \{0, 1\}$ を

$$\delta(a,b) = \begin{cases} 1 & (a = b) \\ 0 & (a \neq b) \end{cases}$$

で定義する．

$m \leq n$ として，文字列 $T = t_1 t_2 \cdots t_n \in \Sigma^*$ と $P = p_1 p_2 \cdots p_m \in \Sigma^*$ について，

$$c_i = \sum_{j=1}^{m} \delta(t_{i+j-1}, p_j) \quad (1 \leq i \leq n-m+1)$$

を要素とするベクトル $C(T,P)$ を T と P の間の**スコアベクトル**と呼ぶ．**不一致を許す文字列照合問題**とは，それぞれ**テキスト**と**パターン**と呼ばれる 2 つの文字列が与えられて，その間のスコアベクトルを求めるものである．

$w = (w_0, w_1, \ldots, w_{n-1})$ を，n 次ベクトル $u = (u_0, u_1, \ldots, u_{n-1})$ と $v = (v_0, v_1, \ldots, v_{n-1})$ の畳み込みとする，つまり，$-n+1 \leq i \leq -1$ について $v_i = v_{i+n}$ として，

$$w_i = \sum_{j=0}^{n-1} u_j \cdot v_{i-j} \quad (0 \leq i \leq n-1)$$

が成り立つ．このとき，U，V，W をそれぞれ u，v，w の離散フーリエ変換とすると，$W = U \cdot V$ が成り立つので，w は u と v から FFT により $O(n \log n)$ 時間で求めることができる．

3. 関 連 研 究

3.1 FFT を用いた基本アルゴリズム

スコアベクトルを畳み込み演算により求めるアイデアは Fischer ら[6] によって提案された．このアイデアに基づき，スコアベクトルを高速フーリエ変換（FFT）によって求める基本的なアルゴリズムが文献 7) にまとめられている．

Σ から $\{0,1\}$ への写像の集合

$$\Psi = \{\psi_a \mid \psi_a(b) = \delta(a,b), a \in \Sigma\}$$

について，任意の a，$b \in \Sigma$ について

$$\sum_{\psi \in \Psi} \psi(a) \cdot \psi(b) = \delta(a,b)$$

である．よって，スコアベクトルは

$$c_i = \sum_{\psi \in \Psi} \sum_{j=1}^{m} \psi(t_{i+j-1}) \cdot \psi(p_j) \quad (1 \leq i \leq n-m+1)$$

である．

基本アルゴリズムは，上の式によりスコアベクトルを計算するものである．ここで，ベクトル $(\sum_{j=1}^{m} \psi(t_j) \cdot \psi(p_j), \sum_{j=1}^{m} \psi(t_{j+1}) \cdot \psi(p_j), \ldots, \sum_{j=1}^{m} \psi(t_n) \cdot \psi(p_j))$ は，2 つの n 次ベクトル $(\psi(t_1), \psi(t_2), \ldots, \psi(t_n))$ と $(\psi(p_m), \psi(p_{m-1}), \ldots, \psi(p_1), 0, \ldots, 0)$ の畳み込みであり，$O(n \log n)$ 時間で計算することができる．さらに，スコアベクトルは，テキストを重複を持たせて分割し，各部分文字列とパターンとのスコアベクトルから求めることができるので，各部分文字列の長さを $\Theta(m)$ とすれば，$\Theta(n/m)$ 回の $O(m \log m)$ 時間の計算で求めることができる[4]．よって，このベクトルは $O(n \log m)$ 時間で求めることができる．

$|\Psi| = \sigma$ より，基本アルゴリズムの計算時間は $O(\sigma n \log m)$ である．

3.2　確率的アルゴリズム

FFT による $O(n \log m)$ 時間の計算を σ 回繰り返す基本アルゴリズムは，σ が大きい場合は有効でない．Atallah ら[1] は，スコアベクトルの推定値を出力するモンテカルロ型の確率的アルゴリズムを提案した．k 個のサンプルにより $O(kn \log m)$ 時間で計算される推定値の期待値はスコアベクトルに等しく，分散の上限は $(m - c_i)^2/k$ である．この分散が小さいほどスコアベクトルに近い推定値を得やすいので，これを推定値の精度と見なすことができる．ω を 1 の原始 σ 乗根とする．Φ を Σ から $\{0, 1, \ldots, \sigma - 1\}$ への写像全体からなる集合とする．このとき，$\omega^0 = 1$ かつ $\sum_{n=0}^{\sigma-1} \omega^n = 0$ であるので，任意の a，$b \in \Sigma$ について

$$\frac{1}{|\Phi|} \sum_{\phi \in \Phi} \omega^{\phi(a)} \cdot \omega^{-\phi(b)} = \delta(a, b)$$

である．Atallah らは Φ から写像をランダムに選ぶことでアルゴリズムの確率化を行っている．

Baba ら[2] は，文字から数値へのより単純な写像を用いて，Atallah らと同様のアルゴリズムを提案している．Ψ' を Σ から $\{-1, 1\}$ への写像全体からなる集合とすると，

$$\frac{1}{|\Psi'|} \sum_{\psi \in \Psi'} \psi(a) \cdot \psi(b) = \delta(a, b)$$

が成り立つ．推定値の分散の上限は Atallah らによるものと等しい．

Schoenmeyr ら[9] は，Atallah らのアルゴリズムの Φ を単射に制限し，推定値の分散が小さくなるように改良した．Φ' を Σ から $\{0, 1, \ldots, \sigma - 1\}$ への単射全体からなる集合とすると，任意の a，$b \in \Sigma$ について

$$\frac{\sigma - 1}{\sigma} \left(\frac{1}{|\Phi'|} \sum_{\phi \in \Phi'} \omega^{\phi(a)} \cdot \omega^{-\phi(b)} \right) + \frac{1}{\sigma} = \delta(a, b)$$

が成り立つ．推定値の分散の上限は $\sigma(\sigma - 3)(m - c_i)^2/2(\sigma - 1)^2 k$ である．

3.3　非復元抽出による精度の向上

3.2 節の確率的アルゴリズムは，文字から数値への写像の集合から要素をランダムに抽出することで確率化を行っている．一般に，要素数 s の母集団からの非復元抽出によるサンプル k 個の平均の分散は，復元抽出の場合に対し $(s-k)/(s-1)$ 倍になる．これは，s が小さいほど小さくなり，より厳密解に近い推定値が得られる．抽出の母集団となる写像の集合の要素数は，Atallah らのアルゴリズムでは σ^σ，Baba らのアルゴリズムでは 2^σ，Schoenmeyr らのアルゴリズムでは $\sigma!$ であり，いずれも非復元抽出による推定値の精度の向上は小さい．

Nakatoh ら[8] は，σ 以上の最小の素数 p について，要素数が $p - 1$ の写像の集合を用いたアルゴリズムを提案した．非復元抽出の確率的アルゴリズムの場合，推定値の分散の低下は $(p - 1 - k)/(p - 1)$ である．1 以上の整数 n について $n \leq p < 2n$ である素数 p が存在するので，k の増加にともない $(p - 1 - k)/(p - 1)$ による分散の低下が期待できる．たかだか $k = 2\sigma - 2$ 個のサンプルで分散が 0 になる．

Baba ら[3] は，畳み込み中の積の演算として単純な積を用いるならば，2 つの文字についての関数 δ を実現するのに，文字から数値への写像が少なくとも $\sigma - 1$ 個必要であることを示した．このことから，非復元抽出による精度は，$\sigma - 1$ 個の写像を用いる場合に最も良くなることが分かる．その分散は復元抽出の分散の $(\sigma - 1 - k)/(\sigma - 1)$ 倍である．本稿のアルゴリズムは，$\sigma - 1$ 個の集合から写像を選択するので，この最大の精度向上が得られる．

4. 提案アルゴリズム

本章では，スコアベクトルの推定値を高い精度で出力するモンテカルロ型アルゴリズムを提案する．まず，文字から複素数への $\sigma - 1$ 個の写像を用いて，FFT による決定性アルゴリズムが得られることを示す．次に，$\sigma - 1$ 個から写像をランダムに選ぶことで得られる確率的アルゴリズムの推定値の分散の上限を示す．

4.1　決定性アルゴリズム

φ を Σ から $\{0, 1, \ldots, \sigma - 1\}$ への単射とする．また，1 の原始 σ 乗根 ω について，

$$\Phi = \{\phi_\ell \mid \phi_\ell(a) = \omega^{\ell\varphi(a)}, 1 \leq \ell \leq \sigma - 1\}$$

とする．このとき，$\omega^0 = 1$ かつ，任意の整数 $n > 0$ について $\sum_{\ell=0}^{\sigma-1} \omega^{\ell n} = 0$ だから，任意の a，$b \in \Sigma$ について，

$$\sum_{\ell=1}^{\sigma-1} \phi_\ell(a) \cdot \overline{\phi_\ell(b)} = \sum_{\ell=0}^{\sigma-1} \omega^{\ell(\varphi(a) - \varphi(b))} - 1 = \sigma\delta(a, b) - 1$$

である．ただし，$\bar{c}$ は c の複素共役を表す．よって，

$$
\begin{aligned}
c_i &= \sum_{k=1}^{m} \delta(t_{i+j-1}, p_j) \\
&= \sum_{j=1}^{m} \left(\frac{1}{\sigma} \sum_{\ell=1}^{\sigma-1} \phi_\ell(t_{i+j-1}) \cdot \overline{\phi_\ell(p_j)} + \frac{1}{\sigma} \right) \\
&= \frac{1}{\sigma-1} \sum_{\ell=1}^{\sigma-1} \left(\frac{\sigma-1}{\sigma} \sum_{j=1}^{m} \phi_\ell(t_{i+j-1}) \cdot \overline{\phi_\ell(p_j)} + \frac{m}{\sigma} \right) \quad (1 \le i \le n-m+1)
\end{aligned}
$$

が成り立つ．しかるに，3.1 節の基本アルゴリズムと同様に，FFT による $O(n \log m)$ 時間の計算を $|\Phi| = \sigma - 1$ 回繰り返すことによってスコアベクトルを計算することができる．

定理 1　長さがそれぞれ n と m の Σ 上のテキストとパターンのスコアベクトルは，基本アルゴリズムにより $O(n \log m)$ 時間の計算の $|\Sigma| - 1$ 回の繰返しで求められる．

4.2　確率的アルゴリズム

4.1 節の決定性アルゴリズムは，3.2 節のアルゴリズムと同様に，Φ から写像をランダムに選ぶことで確率化が可能である．選ばれた写像 ϕ_ℓ について，スコアベクトルのサンプルを

$$
s_i^{(\ell)} = \frac{\sigma-1}{\sigma} \sum_{j=1}^{m} \phi_\ell(t_{i+j-1}) \cdot \overline{\phi_\ell(p_j)} + \frac{m}{\sigma} \quad (1 \le i \le n-m+1)
$$

とする．また，L を $\{1, 2, \ldots, \sigma-1\}$ から選ばれた k 個の整数として，スコアベクトルの推定値を

$$
\hat{s_i} = \frac{1}{k} \sum_{\ell \in L} s_i^{(\ell)} \quad (1 \le i \le n-m+1)
$$

とする．このとき，明らかに $\hat{s_i}$ の期待値は c_i である．また，推定値として $\hat{s_i}$ の実数部 $\Re(\hat{s_i})$ だけを考慮しても期待値は変わらない．

$V[x]$ を x の分散，$E[x]$ を期待値とする．$\hat{s_i}$ の定義と分散の基本的性質より，

$$
V[\Re(\hat{s_i})] = \frac{1}{k} V[\Re(s_i^{(\ell)})]
$$

である．また，

$$
\begin{aligned}
V[\Re(s_i^{(\ell)})] &= V\left[\frac{\sigma-1}{\sigma} \sum_{j=1}^{m} \Re\left(\phi_\ell(t_{i+j-1}) \cdot \overline{\phi_\ell(p_j)} \right) + \frac{m}{\sigma} \right] \\
&= \frac{(\sigma-1)^2}{\sigma^2} V\left[\sum_{j=1}^{m} \Re\left(\phi_\ell(t_{i+j-1}) \cdot \overline{\phi_\ell(p_j)} \right) \right] \\
&\le \frac{(\sigma-1)^2}{\sigma^2} \left(\sum_{j=1}^{m} \sqrt{V\left[\Re\left(\phi_\ell(t_{i+j-1}) \cdot \overline{\phi_\ell(p_j)} \right) \right]} \right)^2
\end{aligned}
$$

である．ここで，a，$b \in \Sigma$ について $V\left[\Re\left(\phi_\ell(a) \cdot \overline{\phi_\ell(b)}\right)\right]$ は，$a = b$ のとき 0 である．$a \ne b$ のとき，$\theta = \varphi(a) - \varphi(b)$ として，

$$
\begin{aligned}
V\left[\Re\left(\phi_\ell(a) \cdot \overline{\phi_\ell(b)}\right)\right] &= E\left[\Re\left(\phi_\ell(a) \cdot \overline{\phi_\ell(b)}\right)^2\right] - \left(E\left[\Re\left(\phi_\ell(a) \cdot \overline{\phi_\ell(b)}\right)\right]\right)^2 \\
&= \frac{1}{\sigma-1} \sum_{\ell=1}^{\sigma-1} \cos^2 \frac{2\pi\ell\theta}{\sigma} - \left(-\frac{1}{\sigma-1}\right)^2 \\
&= \frac{1}{\sigma-1} \sum_{\ell=1}^{\sigma-1} \left(\frac{1}{2} + \frac{1}{2} \cos \frac{4\pi\ell\theta}{\sigma} \right) - \frac{1}{(\sigma-1)^2} \\
&= \frac{1}{2} + \frac{1}{2(\sigma-1)} \left(\sum_{\ell=0}^{\sigma-1} \cos \frac{4\pi\ell\theta}{\sigma} - 1 \right) - \frac{1}{(\sigma-1)^2} \\
&= \frac{\sigma(\sigma-3)}{2(\sigma-1)^2}
\end{aligned}
$$

であり，これが任意の 0 でない θ について成り立つ．よって，

$$
V\left[\Re\left(\hat{s_i}\right)\right] \le \frac{(\sigma-3)(m-c_i)^2}{2\sigma k}
$$

である．

また，L を非復元抽出によって決めるならば，母集団の要素数は $\sigma - 1$ であるから，スコアベクトルの推定値の分散の上限は

$$
\frac{\sigma-1-k}{\sigma-2} V\left[\Re\left(\hat{s_i}\right)\right] = \frac{(\sigma-3)(\sigma-1-k)(m-c_i)^2}{2\sigma(\sigma-2)k}
$$

である．

定理 2　確率的アルゴリズムにより $O(kn \log m)$ 時間で計算されるスコアベクトルの推定

値は，期待値がスコアベクトルに等しく，分散の上限は $(\sigma-3)(\sigma-1-k)(m-c_i)^2/2\sigma(\sigma-2)k$ である．

5. 推定値精度の評価実験

提案アルゴリズムの精度に関して，前章で分散の上界を理論的に示しており，それにより従来アルゴリズムに比べて分散の上界が小さいことが明らかである．しかしながら，実際の推定値の精度に関しては明確ではない．本章では，従来アルゴリズムの 1 つとして Atallah らのアルゴリズムを取り上げ，提案アルゴリズムと実験による精度の比較を行うことで，それらの平均的な精度の違いを明らかにする．

5.1 実験環境の実装

提案アルゴリズムの実装および従来アルゴリズムの実装には，Perl 言語を用い，Linux マシン上に実験システムを含めて構築した．FFT の計算には，Perl のモジュールである Math::FFT を用いた．

各アルゴリズムの精度の確認が目的であるため，各スコアの値それぞれについて統計処理が可能となる十分な頻度のマッチングが，実験対象のデータに必要である．このため本実験においては，実験に用いるテキストおよびパターンともに乱数を用いて次のように人工的に生成した．

Σ を有限のアルファベットとする．パターン $P = p_1p_2\cdots p_m$ は，$1 \le j \le m$ について p_j を Σ からランダムに選んで生成する．次に，パターン P から近似文字列を生成する演算子，$\texttt{Substitution}(P)$ を定義する．Substitution は，ランダムに決定した個数，ランダムに選ばれた位置の文字を，それぞれ Σ からランダムに選ばれた文字で置き換える演算子である．テキスト T の生成は，$T =$ “”(空文字列) から開始し，$T \leftarrow T.\texttt{Substitution}(P)$ [⋆1] を T が必要な長さに到達するまで繰り返し適用する．

実験ごとに必要とするデータが異なるため，実験用データのアルファベットサイズ Σ や長さなどの詳細は，各実験の説明において記載する．

5.2 精度の評価

提案アルゴリズムの推定値の分散の理論的上界は，$(\sigma-3)(\sigma-1-k)(m-c_i)^2/2\sigma(\sigma-2)k$ である．また，Atallah らのアルゴリズムの推定値の分散の上界は $(m-c_i)^2/k$ である．

いずれも，スコア c_i，パターン長 m，アルファベットサイズ σ，サンプル数 k をパラメータとして含む．また，テキスト長 n は分散の理論値に関与していない．したがって，本実験でも，c_i，m，σ，k をパラメータとし，提案アルゴリズム，従来アルゴリズムを実行し，推定値の正解値に対する分散を求めることで精度を評価する．

各パラメータのうち，サンプル数 k による両アルゴリズムの違いが最も重要であるため，k はつねに変動パラメータとする．残りの 3 パラメータのうちの 2 つを固定したうえで，1 つのパラメータを変化させ，3 次元のグラフを生成する．

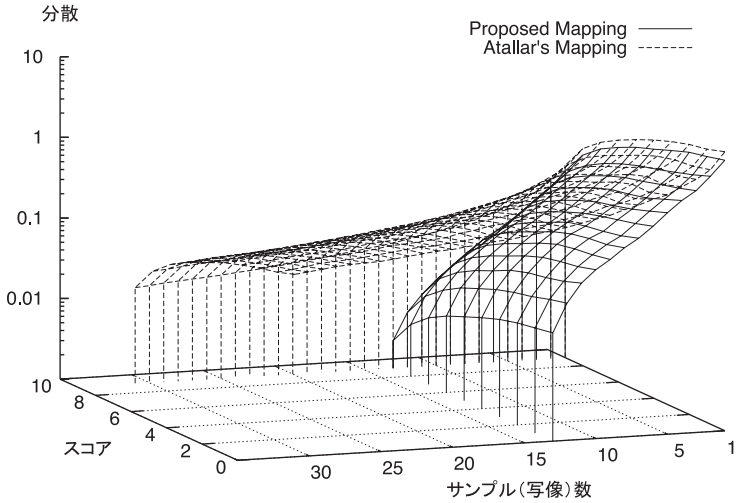

図 2　スコア c_i に応じた精度の変化
Fig. 2　Variation of variance according to Score c_i.

5.2.1 スコア c_i による精度の変化

本項では，スコア c_i が異なる場合の各精度に関して，分散で評価する．本実験では，残りの 2 つのパラメータを $\sigma = 16$，$m = 10$ に固定した．また，テキスト T のサイズは $n = 50,000$ とし，100 セットの T，P に関して分散を求め，平均をとった．その結果，**図 2** の実験結果を得た．

図 2 からは，どちらのアルゴリズムもスコア c_i が大きいと精度が高いこと，提案アルゴリズムの精度が全域で従来アルゴリズムの精度を上回っていること，提案アルゴリズムではサンプル数 k の増加で精度が素早く向上し $k = \sigma - 1$ で厳密解が求まること，従来アルゴリズムではサンプル数 k の増加に対して精度の向上が少なく $k = 2\sigma$ ではまだ厳密解が求ま

⋆1 .(dot) は文字列を結合する演算子

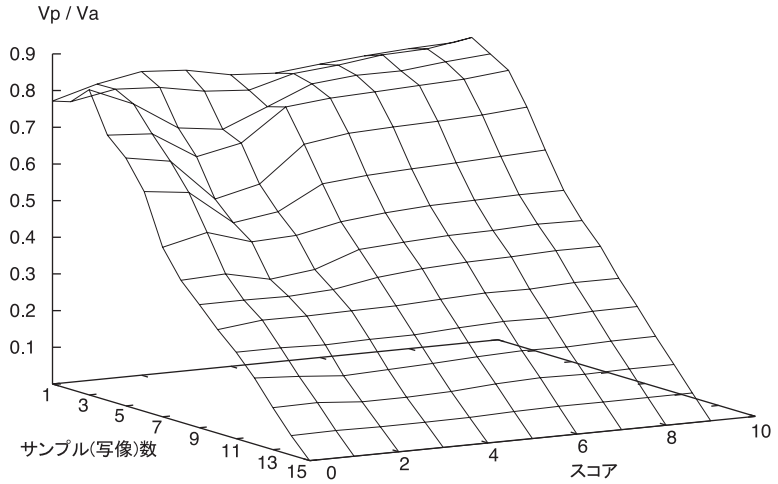

図 3　分散の比率
Fig. 3　Ratio of variance.

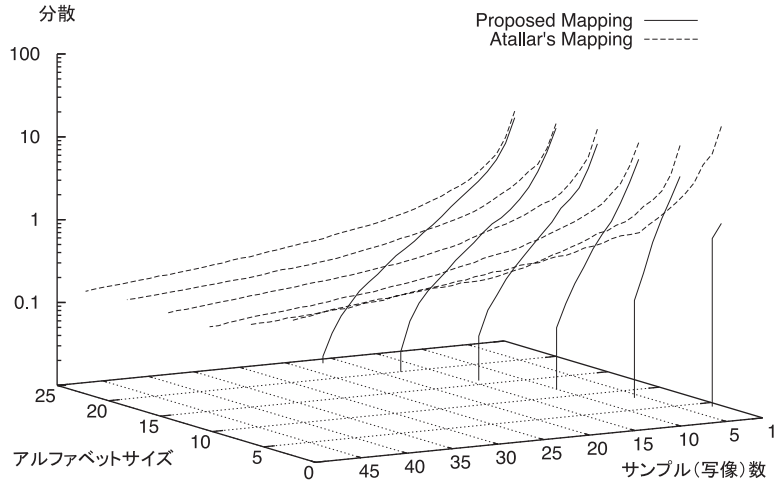

図 4　アルファベットサイズ σ による精度の変化
Fig. 4　Variation of variance according to σ.

らないことが読み取れる．

次に同じデータを用い，提案アルゴリズムの分散 V_p と従来アルゴリズムの分散 V_a の比 V_p/V_a を求め，**図 3** に示した．

2 つのアルゴリズムの分散の上界を示す理論式の比は，式 (1) に示すとおり k に比例して減少する．実験による分散の平均値においても，この図 3 からほぼサンプル k に比例して分散の比 V_p/V_a が減少していることが分かる．

$$\frac{\frac{(\sigma-3)(\sigma-1-k)(m-c_i)^2}{2\sigma(\sigma-2)k}}{\frac{(m-c_i)^2}{k}} = \frac{(\sigma-3)(\sigma-1-k)}{2\sigma(\sigma-2)} \tag{1}$$

5.2.2　アルファベットサイズ σ による精度の変化

アルファベットサイズ σ が推定値の精度に与える影響を実験的に確認する．本実験において，残りの 2 つのパラメータを $c_i = 8$，$m = 20$ に固定した．テキスト T のサイズは $n = 50,000$ とし，100 セットの T，P に関して分散を求め，平均をとった．その結果，**図 4** の結果を得た．

従来アルゴリズムでは，精度がアルファベットサイズにほとんど影響されないこと，サンプル数 k の増加に対する精度の向上が少なく $k = 2\sigma$ ではまだ厳密解が求まらないことが

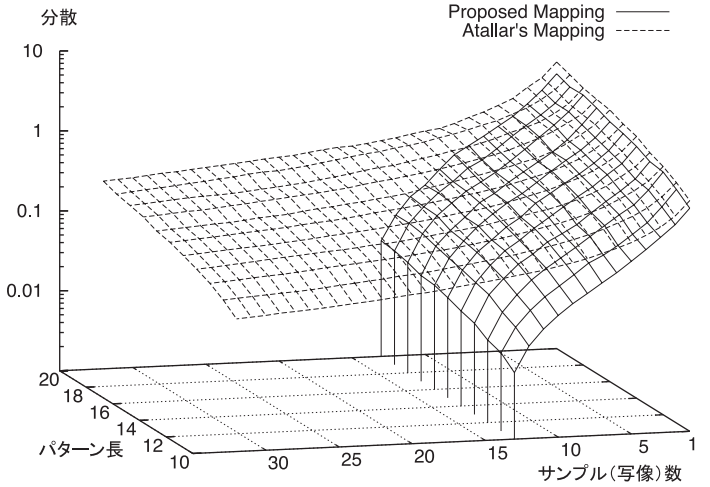

図 5　パターン長 m による精度の変化
Fig. 5　Variation of variance according to m.

図 4 から読み取れる．また，提案アルゴリズムでは，精度が従来アルゴリズムの精度をグラフ全域で上回ること，サンプル数 k の増加に応じて精度が素早く向上し $k = \sigma - 1$ で厳密解が求まることが，同じく図 4 から読み取れる．

5.2.3　パターン長 m による精度の変化

パターン長 m の違いによる精度の違いを実験的に確認する．本実験においては，$\sigma = 16$, $c_i = 8$，$n = 10000$ とし，パターン長 m を 10 から 20 まで 1 ずつ変えた各 100 セットの T，P を生成した．提案アルゴリズムおよび従来アルゴリズムでの推定値の分散を求め，得られた結果を**図 5** に示した．

いずれのアルゴリズムでもパターン長が増えるとパターン中の不一致部分が増加するため，解の分散が大きくなる．図 5 においても，矛盾しない結果が得られている．提案アルゴリズムによる解の推定値の精度は，図 5 の全域において従来アルゴリズムによる精度よりも良く，サンプル数 k が $k = \sigma - 1$ においては推定値が厳密解に一致している．

6. ま と め

我々は，不一致を許す文字列照合のスコア計算に FFT を用いる際の，最適な写像の生成アルゴリズムを提案している．提案アルゴリズムでの写像の総数はアルファベットサイズ $|\Sigma|$ に対して $|\Sigma| - 1$ であり，これは FFT を用いる場合の理論的な下限である．本稿では，提案アルゴリズムを実装し，文字列照合のスコアを求める際の推定値の精度について，従来アルゴリズムとの実験的な比較を行った．従来アルゴリズムに比べ精度が良いこと，サンプル数を $|\Sigma| - 1$ とすることで厳密解が得られることなどを確認した．

参 考 文 献

1) Atallah, M., Chyzak, F. and Dumas, P.: A Randomized Algorithm for Approximate String Matching, *Algorithmica*, Vol.29, No.3, pp.468–486 (2001).
2) Baba, K., Shinohara, A., Takeda, M., Inenaga, S. and Arikawa, S.: A Note on Randomized Algorithm for String Matching with Mismatches, *Nordic Journal of Computing*, Vol.10, pp.2–12 (2003).
3) Baba, K., Tanaka, Y., Nakatoh, T. and Shinohara, A.: A Generalization of FFT Algorithm for String Matching, *Proc. International Symposium on Information Science and Electrical Engineering*, pp.191–194 (2003).
4) Crochemore, M. and Rytter, W.: *Text algorithms*, Oxford University Press, Inc. New York, NY, USA (1994).
5) Crochemore, M. and Rytter, W.: *Jewels of Stringology*, World Scientific Publishing Company (2002).
6) Fischer, M.J. and Paterson, M.S.: String-matching and other products, *Proc. SIAM-AMS Applied Mathematics Symposium*, Massachusetts Institute of Technology Cambridge, MA, USA, pp.113–125 (1974).
7) Gusfield, D.: *Algorithms on strings, trees and sequences*, Cambridge University Press, New York (1997).
8) Nakatoh, T., Baba, K., Ikeda, D., Yamada, Y. and Hirokawa, S.: An Efficient Mapping for Computing the Score of String Matching, *Journal of Automata, Languages and Combinatorics*, Vol.10, No.5/6, pp.697–704 (2005).
9) Schoenmeyr, T. and Zhang, D.Y.: FFT-based Algorithms for the String Matching with Mismatches Problem, *Journal of Algorithms*, Vol.57, No.2, pp.130–139 (2005).
10) 中藤哲也，馬場謙介，森　雅生，廣川佐千男：FFT を用いた近似文字列照合のスコア計算のための最適な写像，*DBSJ Letters*, Vol.6, No.3, pp.25–28 (2007).

(平成 21 年 6 月 20 日受付)
(平成 21 年 8 月 10 日採録)

(担当編集委員　相澤 彰子)

中藤　哲也（正会員）
九州大学情報基盤研究開発センター助教．1992 年九州大学大学院総合理工学研究科修士課程修了．検索エンジン，Web マイニング，文字列処理アルゴリズムの研究に従事．日本データベース学会正会員．

馬場　謙介（正会員）
九州大学附属図書館研究開発室准教授．2002 年九州大学大学院システム情報科学研究科博士課程修了．博士（理学）．文字列処理アルゴリズムの研究に従事．IEEE，EATCS 各会員．

池田　大輔（正会員）

九州大学大学院システム情報科学研究院准教授．1997 年九州大学大学院システム情報科学研究科情報理学専攻博士後期課程退学．博士（理学）．ウェブ・テキストマイニング，学術情報基盤の研究に従事．統計科学研究会，ACM，EATCS 各会員．

森　　雅生（正会員）

九州大学大学評価情報室助教．1996 年九州大学大学院総合理工学研究院博士後期課程単位取得後退学．Web マイニング，WebDB の統合制御（マッシュアップ）手法の研究に従事．

廣川佐千男（正会員）

九州大学情報基盤研究開発センター教授．1979 年九州大学大学院理学研究科修士課程修了，博士（理学）．検索エンジン，テキストマイニング，計算論理学の研究に従事．電子情報通信学会正会員．