An Efficient Algorithm for String Matching with Mismatches

中藤, 哲也
九州大学情報基盤センター

馬場, 謙介
科学技術振興事業団

山田, 泰寛
九州大学大学院システム情報科学府

池田, 大輔
九州大学情報基盤センター

他

近似文字列照合のための効率的なアルゴリズム

中藤　哲也† 　馬場　謙介†† 　山田　泰寛††† 　池田　大輔† 　廣川 佐千男†

† 九州大学 情報基盤センター　〒 812-8581 福岡市東区箱崎 6-10-1
††† 九州大学大学院 システム情報科学府　〒 812-8581 福岡市東区箱崎 6-10-1
†† 科学技術振興事業団　〒 332-0012 埼玉県川口市本町 4-1-8
E-mail: †{nakatoh,daisuke,hirokawa}@cc.kyushu-u.ac.jp, ††baba@i.kyushu-u.ac.jp,
†††yshiro@matu.cc.kyushu-u.ac.jp

あらまし　文字列中に存在する特定のパターンを見つけ出す問題は，文字列照合問題と呼ばれ，Web 上の情報からの検索や 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っている．近似文字列照合問題の一種である不一致を許す文字列照合問題は，単なる文字列照合問題より応用範囲が広く，また難易度も高い．我々は，高速フーリエ変換 (FFT) を利用してこの問題の解を高速に計算する効率的なアルゴリズムを提案する．全ての位置でのマッチングのスコアを求めるもので，ミスマッチ数に制限はない．すなわち k-mismatch 問題より難易度が高い．本アルゴリズムは，k 個のサンプルを用いた確率アルゴリズムで，計算量は $O(kn \log m)$ である．この k は $1 \leqq k \leqq |\Sigma|$ の範囲を持ち，$k = |\Sigma|$ においては常に正しいスコアが得られる決定性アルゴリズムとなる．すなわち，本アルゴリズムにおいては，精度と計算量の兼ね合いを，正確なスコアベクトルを得る事も含め自由に選ぶ事が可能である．
キーワード　文字列照合，近似文字列照合，不一致，FFT，畳み込み，確率アルゴリズム，決定性アルゴリズム

An Efficient Algorithm for String Matching with Mismatches

Tetsuya NAKATOH†, Kensuke BABA††, Yasuhiro YAMADA†††, Daisuke IKEDA†, and Sachio HIROKAWA†

† Computing and Communications Center, Kyushu University
††† Graduate School of Information Science and Electrical Engineering, Kyushu University
Hakozaki 6-10-1,Higashi-ku,Fukuoka,812-8581 Japan
†† Japan Science and Technology Corporation　Honcho 4-1-8, Kawaguchi City, Saitama, 332-0012 Japan
E-mail: †{nakatoh,daisuke,hirokawa}@cc.kyushu-u.ac.jp, ††baba@i.kyushu-u.ac.jp,
†††yshiro@matu.cc.kyushu-u.ac.jp

Abstract　The problem to find out a pattern from the string is called *String matching problem*. That has a wide application range, such as text search, search form the database and an information extraction from Web. Specially, *String matching with mismatches problem* is more difficult problem than String matching problem. We propose a new efficient algorithm to solve String matching with mismatches problem fast by utilizing fast Fourier transformation (FFT). That does not restrict the number of mismatches. That is a randomized algorithm, and its time complexity is $O(kn \log m)$, where k is the number of randomly sampled estimations and its value is in the range of 1 to $|\Sigma|$. We can compute an exact score vector with $k = |\Sigma|$. Exactly, our algorithm can be deterministic, too. We can choose a balance of time complexity and precision freely.
Key words　String Matching, mismatch, FFT, convolution, randomized algorithm, deterministic algorithm.

1. はじめに

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 上の文字列とする時，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である．この問題の解は $O(n)$ で得られる事が知られている．これに対し，パターン P を編集したものをテキスト T 中に見つける問題を**近似文字列照合問題**と言い，単なる文字列照合問題より難易度は高い．編集の種類としては，代入や挿入，削除などがあり，その違いは編集距離として定義される．

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　テキスト acbabbaccb とパターン abbac の間のスコアベクトル

これらと関連する問題として，編集に代入のみを許した**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 上の全ての位置で得られるので，この問題はテキスト T 上の全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 $C(T,P)=(c_1,\ldots,c_{n-m+1})$ を求める問題とみなす事ができる．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，例えば単純にシンボルの比較を m 回ずつ $n-m+1$ 個の i について行なえば良く，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．この問題に関しては，これまでに多くの研究が行なわれている [5] [7]．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，Abrahamson [1] は，一般化された文字列照合を $O(n\sqrt{m\log m})$ で得るアルゴリズムを示した．ハミング距離の近似を得る方法としては，Karloff [8] が $O(\frac{n}{\epsilon^2}\log^c m)$ のアルゴリズムを示している．c は定数項，ϵ は近似の良さを表している．不一致数を k に制限した k-mismatch 問題にもさまざまな解決が図られており，Amir ら [3] は $O(n\sqrt{k\log k})$ と $O((n+\frac{nk^3}{m})\log k)$ の二つのアルゴリズムを示した．

近年，Atallar らはスコアベクトルを求めるためのモンテカルロ型確率アルゴリズムを提案した [2]．これは，スコアベクトルをテキストとパターンに対応する二つの写像関数の畳み込みで表し，FFT を利用して計算する方法である．この時，正確な値を求めるために必要となる非常に多くの写像についてその全てを計算する代わりに，ランダムに選んだ k 個の写像標本の平均によってスコアベクトルの推定を確率的に行なう事で，計算量を $O(kn\log m)$ に抑えている．また，馬場ら [4] は Atallar らと同様の手法において写像数をより少なくする写像の改良を行なっている．

本研究では，馬場ら [4] と同様の手法において，写像の数を大きく削減する手法を示す．計算量は k 個の標本について同じく $O(kn\log m)$ であるが，標本の元となる総写像空間は充分に小さく $1\leqq k\leqq|\Sigma|$ である．$k=|\Sigma|$ とする事で決定性アルゴリズムとして扱うことも可能である．k の取り方で計算量と精度を自由にコントロールできるので，入力の状況に応じた処理が可能である．

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 $|S|$ によって表す．

シンボルの一致を表す関数 δ:$\Sigma\times\Sigma\to\{0,1\}$ を次式のように定義する．

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

但し，$a,b\in\Sigma$ である．

テキスト $T=t_1t_2\cdots t_n\in\Sigma^*$ とパターン $P=p_1p_2\cdots p_m\in\Sigma^*$ について，

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

として得られるベクトル $C(T,P)=(c_1,c_2,\ldots,c_{n-m+1})$ を T と P の間のスコアベクトルと呼ぶ．つまり，c_i は P の先頭のシンボルが T の i 番目の位置にあるときの一致の数である．

3. FFT を使うための条件

c_i は式 (1) の定義より $\sum_{j=1}^{m}\delta(t_{i+j-1},p_j)$ である．この δ 関数を何らかの方法で，t の関数と p の関数の積へ変換することができれば，ベクトル $C(T,P)$ の計算は，T と P の畳み込み (convolution) に変換可能である．畳み込みで表現されたベクトル $C(T,P)$ は高速フーリエ変換 (FFT) によって，$O(n\log n)$ で計算できる．

更に，計算量を $O(n\log m)$ に落とすために，[2] と同様に [5] の技法を用いる事ができる．テキスト T を重なった部分を持つ長さ $(1+\alpha)m$ ずつの塊に分割し，そのそれぞれについて別々に操作を行なう．ひとつの塊についての操作によって，C の成分のうち αm 個が求められる．塊の数は $n/(\alpha m)$ であり，それぞれの塊は FFT によって $O((1+\alpha)m\log((1+\alpha)m))$ で計算されるので，$\alpha=O(m)$ ととると全体の計算量は，

$$\begin{aligned}&\frac{n}{\alpha m}O((1+\alpha)m\log((1+\alpha)m))\\ &\quad=O(\frac{(1+\alpha)}{\alpha}n\log((1+\alpha)m))\\ &\quad=O(n\log m)\end{aligned}$$

である．

シンボルの一致を表す δ 関数を，t の関数と p の関数の積で直接表現する方法は無い．次に紹介する Atallar らの方法 [2]，馬場らの方法 [4]，本稿で提案する方法はいずれも，アルファベットの集合をまず複数の写像に変換し，各写像に対する結果の総和が δ 関数と一致するような関数を導入して，δ 関数を間接的に表現している．

4. 従来の確率アルゴリズム

4.1 Atallar らの方法 [2]

アルファベット Σ を $|\Sigma|$ 乗根からなる複素数空間に写像し，ある写像の複素数とその共役複素数との積を用いることで，シンボルが一致する場合の δ 関数を 1 にしている．しかし，この方法によって不一致の場合に 0 を得るためには，複素平面の単位円上に均一に散らばるように複数の写像を設定し，その積の総和を計算する必要がある．これには個々のシンボルの $|\Sigma|$ 乗根からなる複素数空間への全ての写像の総和が必要であり，その写像総数は $|\Sigma|^{|\Sigma|}$ にのぼる．この計算量は現実的でない．そのため，Atallar らは $|\Sigma|^{|\Sigma|}$ 個の写像集合からランダムに k 個の写像を取り出して計算を行なう事で，近似的に δ 関数を実現している．

計算量は $O(kn\log m)$ であり，期待値はスコアベクトル C に等しく，分散は $(m-c_i)^2/k$ に押さえられる事が示されている．

4.2 馬場らの方法 [4]

Atallar らの方法を改良し，アルファベット Σ を $\{-1,1\}$ に写像する．シンボルが一致する場合の積は，そのシンボルが $\{-1,1\}$ どちらに写像されていても 1 である．不一致の場合は，積が $\{-1,1\}$ の両方に均等に散らばるように複数の写像を行なうことで，その総和を 0 としている．必要な全写像の数は $2^{|\Sigma|}$ であり，Atallar らによる写像集合より小さい．しかしながら，$2^{|\Sigma|}$ 個の全ての写像に関して FFT を計算する事は困難である．

計算量，期待値，分散とも，Atallar らの方法と同じである事が示されている．

5. 効率的なアルゴリズム

本節では，我々の提案する新しい効率的なアルゴリズムについて示す．始めに準備として，$|\Sigma|$ 個の写像を用いた決定性アルゴリズムについて説明した上で，k 個のサンプルを用いる確率アルゴリズムへと発展させる．

5.1 決定性アルゴリズム

アルファベット Σ のある一つのシンボルのみを 1 とし，その他のシンボルを -1 とする $\{1,-1\}$ への写像の集合を Ψ_Σ とする．このとき，明らかに $|\Psi_\Sigma|=|\Sigma|$ である．文脈から明らかな場合，Ψ_Σ を Ψ と略記する．

あるシンボル $x\in\Sigma$ を 1 に変換する写像 $\psi_x\in\Psi$ は次式のように定義される．

$$\psi_x(a)=\begin{cases}1 & a=x\ \text{のとき}\\ -1 & a\neq x\ \text{のとき}\end{cases}$$

このとき，次の補題が得られる．

［補題 1］ 二つのシンボル $a,b\in\Sigma$ について，それらの一致を表す δ 関数は，次式で表される．

$$\delta(a,b)=\frac{1}{4}\sum_{\psi_x\in\Psi}\psi_x(a)\psi_x(b)-\frac{|\Sigma|}{4}+1$$

［証明 1］ 写像の積 $\psi_x(a)\psi_x(b)$ から得られる値を，$a=b$ である場合と $a\neq b$ である場合に分けて考える．

I）$a=b$ の場合

$$\begin{cases}\psi_x(a)\psi_x(b)=1\cdot 1=1 & x=a=b\ \text{のとき}\\ \psi_x(a)\psi_x(b)=-1\cdot -1=1 & x\neq a=b\ \text{のとき}\end{cases}$$

すなわち，$a=b$ の場合は ψ_x に関わらずその積は 1 となる．よって $a=b$ の場合の写像の合計 $F_{a=b}$ は，

$$F_{a=b}=\sum_{\psi_x\in\Psi}\psi_x(a)\psi_x(b)=\sum_{\psi_x\in\Psi}1=|\Psi|=|\Sigma| \tag{2}$$

II）$a\neq b$ の場合

$x=a$ である場合，$x=b$ である場合，$x\neq a$ でかつ $x\neq b$ である場合の 3 通りに分類でき，写像の積の合計 $F_{a\neq b}$ は，これらの和である．

$$\begin{aligned}F_{a\neq b} &= \sum_{\psi_x\in\Psi}\psi_x(a)\psi_x(b)\\ &= \psi_a(a)\psi_a(b)+\psi_b(a)\psi_b(b)+\sum_{\substack{\psi_x\in\Psi\\ x\neq a,x\neq b}}\psi_x(a)\psi_x(b)\\ &= 1\cdot(-1)+(-1)\cdot 1+\sum_{\substack{\psi_x\in\Psi\\ x\neq a,x\neq b}}(-1)\cdot(-1)\\ &= (-1)+(-1)+\sum^{|\Psi|-2}1\\ &= |\Psi|-4\\ &= |\Sigma|-4\end{aligned} \tag{3}$$

ここで，δ 関数は，$F_{a=b}$ を 1 に，$F_{a\neq b}$ を 0 に変換する事で得られる．そのような線形変換 $\delta(a,b)=\alpha F+\beta$ を求める．

連立一次方程式

$$\begin{cases}\alpha F_{a=b}+\beta=1\\ \alpha F_{a\neq b}+\beta=0\end{cases}$$

を解くと，次式が得られる．

$$\begin{aligned}\delta(a,b) &= \frac{1}{4}F+\frac{4-|\Sigma|}{4}\\ &= \frac{1}{4}\sum_{\psi_x\in\Psi}\psi_x(a)\psi_x(b)+\frac{4-|\Sigma|}{4}\end{aligned} \tag{4}$$

□

この δ 関数により，スコア c_i に関して次の補題が得られる．

［補題 2］ $1\leqq i\leqq n-m+1$ である，あらゆる i について，

$$c_i=\frac{1}{4}\sum_{x\in\Psi}\sum_{j=1}^{m}\psi_x(t_{i+j-1})\psi_x(p_j)-\frac{m|\Sigma|}{4}+m$$

［証明 2］ 式（1）に式（4）を代入すると，

$$\begin{aligned}c_i &= \sum_{j=1}^{m}\delta(t_{i+j-1},p_j)\\ &= \sum_{j=1}^{m}\left(\frac{1}{4}\sum_{\psi_x\in\Psi}\psi_x(a)\psi_x(b)+\frac{4-|\Sigma|}{4}\right)\\ &= \sum_{j=1}^{m}\left(\frac{1}{4}\sum_{\psi_x\in\Psi}\psi_x(a)\psi_x(b)\right)+\sum_{j=1}^{m}\frac{4-|\Sigma|}{4}\end{aligned}$$

$$= \frac{1}{4}\sum_{\psi_x\in\Psi}\sum_{j=1}^{m}\psi_x(a)\psi_x(b) + \frac{m(4-|\Sigma|)}{4}.$$

□

スコアベクトル $C(T,P)$ は，［補題 2］より，

$$\sum_{j=1}^{m}\psi_x(t_{i+j-1})\psi_x(p_j)$$

の計算を $|\Sigma|$ 回行ない，線形変換を行なうことで計算できる事が分かる．よって，次の定理を得る．

［定理 1］ スコアベクトル $C(T,P)$ は $O(|\Sigma|n\log m)$ で正確に計算できる．

5.2 確率アルゴリズム

前節のアルゴリズムは効率的ではあるが，それでもなお，アルファベットサイズ $|\Sigma|$ と同じ回数の FFT 計算が必要であるので確率化を試みる．

写像集合 Ψ_Σ より，一様に無作為に k 個を取り出した部分集合 Ψ_k について考える．$|\Psi_k| = k$ である．この部分集合 Ψ_k についての δ 関数が求まれば，前節の決定性アルゴリズムから確率アルゴリズムが得られる．

［補題 3］ 二つのシンボル $a,b\in\Sigma$ について，それらの一致を表す δ 関数は，写像 Ψ_Σ から k 個を取り出して作った部分集合 Ψ_k を用いて次の期待値で表される．

$$\delta_k(a,b) = \frac{|\Sigma|}{4k}\sum_{\psi_x\in\Psi_k}\psi_x(a)\psi_x(b) + \frac{4-|\Sigma|}{4}$$

［証明 3］ ２つの写像の積 $\Psi_x(a)\cdot\Psi_x(b)$ の合計の期待値 E を，$a=b$ の場合と $a\neq b$ の場合に分けて求める．

I）$a=b$ の場合

期待値 $E_{a=b}$ は，

$$E_{a=b} = \sum_{\psi_x\in\Psi_k}\psi_x(a)\psi_x(b) = \sum_{\psi_x\in\Psi_k}1 = |\Psi_k| = k$$

である．これは，$k=|\Sigma|$ において，式（2）と一致する．

II）$a\neq b$ の場合

ψ_a と ψ_b が Ψ_k に含まれるかどうかによって，２つの写像の積の値は異なるので，それぞれの場合における写像の積の合計値 X_i とその確率 p_i[(注1)] を求め，期待値 E を計算する．

また各 X_i について，$x=a$ である場合，$x=b$ である場合，$x\neq a$ でかつ $x\neq b$ である場合の 3 通りに分類できるのは，前節と同様である．

II-a）$\psi_a\in\Psi_k$ でかつ $\psi_b\in\Psi_k$ の場合

写像の積の合計値 X_2 は，

$$\begin{aligned}
X_2 &= \sum_{\psi_x\in\Psi_k}\psi_x(a)\psi_x(b)\\
&= \psi_a(a)\psi_a(b) + \psi_b(a)\psi_b(b) + \sum_{\substack{\psi_x\in\Psi_k\\ x\neq a,\, x\neq b}}\psi_x(a)\psi_x(b)\\
&= 1\cdot(-1) + (-1)\cdot 1 + \sum_{\substack{\psi_x\in\Psi_k\\ x\neq a,\, x\neq b}}(-1)\cdot(-1)\\
&= (-2) + \sum^{|\Psi_k|-2} 1\\
&= |\Psi_k| - 4\\
&= k-4
\end{aligned}$$

ψ_a と ψ_b の両方が Ψ_k に含まれる確率 p_2 は，

$$\begin{aligned}
p_2 &= \frac{{}_2C_2\cdot{}_{(|\Sigma|-2)}C_{(k-2)}}{{}_{|\Sigma|}C_k}\\
&= \frac{k(k-1)}{|\Sigma|(|\Sigma|-1)}
\end{aligned}$$

II-b）ψ_a と ψ_b のどちらか一方が Ψ_k に含まれる場合

含まれる一方の写像を仮に ψ_a とすると，写像の積の合計値 X_1 は，

$$\begin{aligned}
X_1 &= \sum_{\psi_x\in\Psi_k}\psi_x(a)\psi_x(b)\\
&= \psi_a(a)\psi_a(b) + \sum_{\psi_x\in\Psi_k, x\neq a}\psi_x(a)\psi_x(b)\\
&= 1\cdot(-1) + \sum_{\psi_x\in\Psi_k, x\neq a}(-1)\cdot(-1)\\
&= (-1) + \sum^{|\Psi_k|-1} 1\\
&= |\Psi_k| - 2\\
&= k-2
\end{aligned}$$

確率 p_1 は，$\psi_a\in\Psi_k$ と $\psi_b\in\Psi_k$ の両方の場合があるので，

$$\begin{aligned}
p_1 &= \frac{{}_1C_1\cdot{}_{(|\Sigma|-2)}C_{(k-1)}}{{}_{|\Sigma|}C_k}\times 2\\
&= \frac{2k(|\Sigma|-k)}{|\Sigma|(|\Sigma|-1)}
\end{aligned}$$

II-c）$\psi_a\notin\Psi_k$ でかつ $\psi_b\notin\Psi_k$ である場合

写像の積の合計値 X_0 は，

$$\begin{aligned}
X_0 &= \sum_{\psi_x\in\Psi_k}\psi_x(a)\psi_x(b)\\
&= \sum_{\psi_x\in\Psi_k, x\neq a, x\neq b}(-1)\cdot(-1)\\
&= \sum^{|\Psi_k|}1\\
&= |\Psi_k|\\
&= k
\end{aligned}$$

ψ_a と ψ_b の両方が，Ψ_k に含まれない確率 p_0 は，

$$\begin{aligned}
p_0 &= \frac{{}_2C_0\cdot{}_{(|\Sigma|-2)}C_k}{{}_{|\Sigma|}C_k}\\
&= \frac{(|\Sigma|-k)(|\Sigma|-k-1)}{|\Sigma|(|\Sigma|-1)}
\end{aligned}$$

(注1)：n 個の集合に m 個の特別な要素が入っているとする．そこから無作為に k の要素を取り出すとすると，m 個の要素のうち，x 個がそれに含まれる確率は，次の式で求められる．

$$p_x = \frac{{}_mC_x\cdot{}_{(n-m)}C_{(k-x)}}{{}_nC_k}$$

以上より期待値 $E_{a\neqq b}$ は，

$$\begin{aligned} E_{a\neqq b} &= \sum_i X_i \cdot p_i \\ &= X_2 \cdot p_2 + X_1 \cdot p_1 + X_0 \cdot p_0 \\ &= (k-4) \cdot \frac{k(k-1)}{|\Sigma|(|\Sigma|-1)} + (k-2) \cdot \frac{2k(|\Sigma|-k)}{|\Sigma|(|\Sigma|-1)} \\ &\quad + k \cdot \frac{(|\Sigma|-k)(|\Sigma|-k-1)}{|\Sigma|(|\Sigma|-1)} \\ &= \frac{k(|\Sigma|-4)}{|\Sigma|} \end{aligned}$$

これは，$k = |\Sigma|$ において式（3）と一致する事が分かる．

以上により，k 個のサンプルによる写像の積の合計値として，次の期待値 E を得た．

$$E = \begin{cases} k & a = b \text{ のとき} \\ \frac{k(|\Sigma|-4)}{|\Sigma|} & a \neqq b \text{ のとき} \end{cases}$$

前節と同様に，δ 関数を得るため線形変換 $\delta(a,b) = \alpha E + \beta$ を求める．

$$\begin{cases} \alpha E_{a=b} + \beta = 1 \\ \alpha E_{a\neqq b} + \beta = 0 \end{cases}$$

この連立一次方程式を解く事により，次式が得られる．

$$\begin{aligned} \delta_k(a,b) &= \frac{|\Sigma|}{4k} E + \frac{4-|\Sigma|}{4} \\ &= \frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi_k} \psi_x(a)\psi_x(b) + \frac{4-|\Sigma|}{4} \end{aligned} \quad (5)$$

□

この式は $k = |\Sigma|$ において，式（4）と一致する事が分かる．

得られた δ 関数により，スコア c_i に関して次の補題が得られる．

［補題 4］ $1 \leqq i \leqq n-m+1$ である，あらゆる i について，

$$c_i = \frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi_k} \sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j) + \frac{m(4-|\Sigma|)}{4}$$

［証明 4］ 式（1）に式（5）を代入すると，

$$\begin{aligned} c_i &= \sum_{j=1}^{m} \delta_k(t_{i+j-1}, p_j) \\ &= \sum_{j=1}^{m} \left(\frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi} \psi_x(t_{i+j-1})\psi_x(p_j) + \frac{4-|\Sigma|}{4} \right) \\ &= \sum_{j=1}^{m} \left(\frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi} \psi_x(t_{i+j-1})\psi_x(p_j) \right) + \sum_{j=1}^{m} \frac{4-|\Sigma|}{4} \\ &= \frac{|\Sigma|}{4k} \sum_{\psi_x \in \Psi} \sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j) + \frac{m(4-|\Sigma|)}{4}. \end{aligned}$$

□

スコアベクトル $C(T,P)$ は，［補題 4］より，

$$\sum_{j=1}^{m} \psi_x(t_{i+j-1})\psi_x(p_j)$$

の計算を k 回行ない，一度の線形変換を行なうことで確率的に求まる．

よって，次の定理を得る．

Procedure ComputeScore
入力：Σ^* 上のテキスト $T = t_1 \cdots t_{(1+\alpha)m}$
入力：Σ^* 上のパターン $P = p_1 \cdots p_m$
出力：スコアベクトル $C(T,P)$
for $\psi_\ell \in \Psi_k$ **do begin**
　$\psi_\ell(T)$ を求め T_ℓ とする．
　　（T_ℓ は長さ $(1+\alpha)m$ の $\{0,1\}$ 上の列である．）
　$\psi_\ell(P)$ の後ろに αm 個の 0 を付加し P_ℓ とする．
　　（P_ℓ は長さ $(1+\alpha)m$ の $\{0,1\}$ 上の列である．）
　P_ℓ の列を反転する．これを P_ℓ^R とする．
　T_ℓ と P_ℓ^R の畳み込み c_ℓ を FFT により計算する．
end
$\sum_{\psi_\ell \in \Psi'_\Sigma} c_\ell$ を計算し，
必要な一次変換を行なって $C(T,P)$ として出力する．

図 2　アルゴリズム

［定理 2］ スコアベクトル $C(T,P)$ は k 個のサンプルを用いて確率的に計算できる．得られた値の期待値は正しい値と一致する．計算量は，$O(kn\log m)$ である．

得られたアルゴリズムを図 2 に示す．

6. ま と め

不一致を許す文字列照合のための効率的なアルゴリズムを与えた．Atallar ら [2] や馬場ら [4] によるアルゴリズムと同様，FFT による畳み込みを用いるものであるが，本アルゴリズムは少ない写像をベースとした確率アルゴリズムであり，正確なスコアを求める決定性アルゴリズムとしての動作を含め，精度と計算量を自由に設定できる．計算量は $O(kn\log m)(1 \leqq k \leqq |\Sigma|)$ である．

本アルゴリズムにおける推定値の正確な分散を求めることは今後の課題である．また，実証システムを構築し，実装上の問題の確認を行なうと同時に，テキスト検索やウェブマイニングなどへの応用実験を行なってゆく予定である．

文　献

[1] Abrahamson, K.: Generalized string matching. SIAM Journal on Computing 16, 6, 1039-1051. 1987.

[2] Atallah, M. J., Chyzak, F., and Dumas, P.: A Randomized Algorithm for Approximate String Matching. Algorithmica 29, 468-486. 2001.

[3] Amir, A., Lewenstein, M. and Porat, E.: Faster algorithms for string matching with k mismatches. Symposium on Discrete Algorithms, pp.794-803, 2000.

[4] Baba, K., Shinohara, A., Takeda, M., Inenaga, S., and Arikawa, S.: A Note on Randomized Algorithm for String Matching with Mismatches. Proceedings of the Prague Stringology Conference '02, 9–17, 2002.

[5] Crochemore, M. and Rytter, W.: Text Algorithms. Oxford University Press, New York. 1994.

[6] Fischer, M. J. and Paterson, M. S.: String-matching and other products. In Complexity of Computation (Proceedings of the SIAM-AMS Applied Mathematics Symposium, New York, 1973), 113-125. 1974.

[7] Gusfield, D.: Algorithms on Strings, Trees, and Sequences. Cambridge University Press, New York. 1997.

[8] Karloff, H.: Fast algorithms for approximately counting mismatches. Information Processing Letters 48, 2, 53-60. 1993.