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

An Optimal Mapping for Score of String Matching with FFT

中藤 哲也[†] 馬場 謙介[‡]
森 雅生[§] 廣川 佐千男[††]

Tetsuya NAKATOH Kensuke BABA
Masao MORI Sachio HIROKAWA

文字列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，単なる文字列照合より応用範囲が広く，また難易度も高い．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アルゴリズムが幾つか提案されている．それらは文字から数値への写像の生成方法により，写像総数と，解の推定値の分散が異なる．本稿で提案するアルゴリズムは，総写像数が理論上での最小であり，推定値の分散も小さい．

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. An extension of the string matching is *string matching with mismatches*, which allows inexact match with substitution, is expected to have wider application even though it has higher complexity. Several randomized algorithms have been proposed which use fast Fourier transformation (FFT) and run fast to solve the problem. All of these algorithms introduce 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. The present paper shows an algorithm which achieves the theoretical minimum number of mappings, and yields an estimate with small variance.**

1. はじめに

文字列照合問題 [1, 2, 3] 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，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．

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.

図 1 はスコアベクトルの例である．Σ をアルファベット，δ を $\Sigma \times \Sigma$ を $\{0,1\}$ へ写像するクロネッカーのデルタ関数とすると，スコアベクトルの i 番目の要素は次式となる．

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

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．Fischer ら [4] 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 [2] が示されている．また，Abrahamson [5] 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 [6] 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 [7] 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 [8]．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り，全てのテキスト位置における一致のスコアベクトルを得る事ができる．この時，正確な値を求めるためには非常に多くの写像についての計算が必要となるが，その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う事で，計算量を $O(kn \log m)$ に抑えている．馬場ら [9] は Atallar らと同様の確率的手法において写像対象を $\{-1, 1\}$ とする事で，写像総数をより少なくする改良を行った．また，我々 [10] は，写像空間を工夫する事により，総写像数を $O(|\Sigma|)$ に抑える改良を行った．これにより，推定値の分散をより低く抑えられる事を示している．但し，何れも証明されている総写像数の下限 $|\Sigma| - 1$ [11] に比べ，大きな写像数を必要としている．

† 正会員 九州大学情報基盤研究開発センター
nakatoh@cc.kyushu-u.ac.jp
‡ 九州大学大学院システム情報科学研究院，
九州大学システム LSI 研究センター
baba@i.kyushu-u.ac.jp
§ 九州大学 大学評価情報室
mori.uoc@mbox.nc.kyushu-u.ac.jp
†† 九州大学情報基盤研究開発センター
hirokawa@cc.kyushu-u.ac.jp

本稿では，上記の各確率アルゴリズム [8, 9, 10] と同様の手法における最適な写像の生成方法を示す．提案写像は，Atallar らの提案した写像集合 [8] の部分集合であり，その写像の総数は理論上最小となる $|\Sigma|-1$ である．計算量は k 個の標本について他の手法と同じく $O(kn\log m)$ 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．

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

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，

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

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

式（1）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，スコアベクトル $C(T,P)$ 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，[8] と同様に [1] の技法を用いる事ができる．すなわち，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，

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

となる．

ところが，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理論上存在しない [11]．次に紹介する Atallar らの方法 [8]，馬場らの方法 [9]，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. 従来の写像

4.1 Atallar らの写像 [8]

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．そのため，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を用いており，その写像総数は $|\Sigma|^{|\Sigma|}$ にのぼる．この計算量 $O(|\Sigma|^{|\Sigma|}n\log m)$ 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う事で，確率アルゴリズムとして近似的に δ 関数を計算している．

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 馬場らの写像 [9]

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，全写像に関する FFT の計算が困難である点は変わらない．また，確率アルゴリズムを用いた時の，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．

4.3 中藤らの写像 [10]

$|\Sigma|\le p$ である最小の素数 p を求め，アルファベット Σ を p 乗根からなる複素数空間に写像する．計算方法は Atallar らと同じであるが，写像空間が充分に小さく，全写像を計算しても $O(|\Sigma|n\log m)$ である．確率化も Atallar らと同様に行えるが，その際のスコアベクトルの期待値の分散は，より小さな $(p-1)^2(p-k-1)(m-c_i)^2/2p^2(p-2)k$ に押さえる事ができる．

5. 最適な写像の提案

本節では，我々の提案する写像について示す．

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する．次にそれを，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．その上で，計算不要な写像を削減する事によって，本アルゴリズムが理論上最小である $|\Sigma|-1$ 個の写像計算で充分であることを示す．

5.1 決定性アルゴリズムと写像

$\sigma=|\Sigma|$ と定義する．φ を Σ から $\{0,1,\cdots,\sigma-1\}$ への全単射とする．$0\le x\le\sigma-1$ と $a\in\Sigma$ について，写像 ϕ_x を次のように定義する．

$$\phi_x(a)=\omega^{x\cdot\varphi(a)}$$

但し ω は 1 の原始 σ 乗根である．

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．

$$\delta(a,b)=\frac{1}{\sigma}\sum_{x=0}^{\sigma-1}\phi_x(a)\cdot\overline{\phi_x(b)}.\qquad(2)$$

但し，$\overline{\phi_x}$ は ϕ_x の共役複素数とする．

証明 1 写像の積 $\phi_x(a)\cdot\overline{\phi_x(b)}$ 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．

$a=b$ の場合，

$$\phi_x(a)\cdot\overline{\phi_x(b)}=\phi_x(a)\cdot\overline{\phi_x(a)}$$

$$= \omega^0$$
$$= 1$$

したがって，あらゆる $0 \le x \le \sigma - 1$ について，式（2）の左辺は常に 1 となる．

$a \neq b$ の場合，

$$\phi_x(a) \cdot \overline{\phi_x(b)} = \omega^{x \cdot \varphi(a)} \cdot \omega^{x \cdot \varphi(b)}$$
$$= \omega^{x \cdot \{\varphi(a) - \varphi(b)\}}$$

である．特定の a, b に関して $\varphi(a) - \varphi(b)$ を一定とみなせるために，$x \cdot \{\varphi(a) - \varphi(b)\}$ は x の値毎に一定の距離を持つ．また，$\omega^\sigma = \omega^0$ である．従って，$\omega^{x \cdot \{\varphi(a) - \varphi(b)\}}$ は，ω の作る単位円上に一定の間隔を持って均等に散らばる．それ故，その総和は 0 であり，式（2）の左辺は常に 0 となる． ∎

補題 2 アルファベット Σ 上のテキスト（長さ n）とパターン（長さ m）間のスコアベクトルは，$O(\sigma n \log m)$ の計算によって求まる．

証明 2 スコアベクトルの定義と補題 1 より，スコアベクトルの i 番目の要素は次式で計算できる．

$$c_i = \frac{1}{\sigma} \sum_{x=0}^{\sigma-1} \sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)}. \quad (3)$$

それゆえ，スコアベクトルは σ 回の畳み込み $\sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)}$（但し $1 \le i \le n$）で計算できる． ∎

［補題 2］よりスコアベクトル $C(T, P)$ は，

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

の計算を σ 回行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，アルファベットサイズ σ と同じ回数の FFT 計算が必要である．そこで，Atallah ら [8] のアルゴリズムにならい，写像 ϕ_x の確率化を試みる．出力は複数の x の値による計算結果の平均値である．我々は，スコアベクトルの i 番目の要素 c_i の**サンプル** $s_i^{(\ell)}$ を次のように定義する．

$$s_i^{(\ell)} = \sum_{j=1}^{m} \phi_{x^{(\ell)}}(t_{i+j-1}) \cdot \overline{\phi_{x^{(\ell)}}(p_j)}$$

但し，$x^{(\ell)}$ は ℓ によって定まる 0 から $\sigma - 1$ の整数である．

k をサンプルの個数とする．スコアベクトルの i 番目の要素の**推定値** $\hat{s}_i$ は，次のように定義される．

$$\hat{s}_i = \frac{1}{k} \sum_{\ell=0}^{k-1} s_i^{(\ell)}$$

式（3）により，この推定値の期待値が c_i に等しい事は明らかである．下記の補題で，推定値の分散の上界を与える．

補題 3 長さ n のテキストと長さ m のパターンの間のスコアベクトルの i 番目の要素 c_i に関する推定値 $\hat{s}_i$ の分散は，k 個のサンプルを用いた場合，$(m - c_i)^2/k$ に制限される．

証明 3 推定値の定義と分散の基本的性質から，推定値の分散 $V(\hat{s}_i)$ は $V(\hat{s}_i) = V(s_i^{(\ell)})/k$ である．Φ を $\{\phi_0, \cdots, \phi_{\sigma-1}\}$ と定義すると，あらゆる $0 \le \ell \le \sigma - 1$ と $a, b \in \Sigma$ について，$\phi_{x^{(\ell)}}(a)^2 = 1$ でかつ $|\phi_{x^{(\ell)}}(a) \cdot \phi_{x^{(\ell)}}(b)| = 1$ であるので，このサンプルの分散 $V(s_i^{(\ell)})$ は

$$V(s_i^{(\ell)}) = \sum_{\ell=0}^{\sigma-1} \frac{(\sum_{j=1}^{m} \phi_{x^{(\ell)}}(t_{i+j-1}) \cdot \phi_{x^{(\ell)}}(p_j) - c_i)^2}{|\Phi|}$$
$$\le (m - c_i)^2$$

である． ∎

5.3 計算不要な写像の削除

前節における基本的な写像では，その数はアルファベット Σ のサイズ $|\Sigma|$ と同じであった．本節では，この写像数を理論的な下限である $|\Sigma| - 1$ に一致させる．

写像 ϕ_0 の値は全てのシンボルに関して 1 である．それゆえ，畳み込みの計算が必要ではない．写像集合 Φ から，この写像 ϕ_0 を取り除き，代わりに 1 を与える．それにより，スコアベクトルの i 番目の要素は，

$$c_i = \frac{1}{\sigma} \sum_{x=0}^{\sigma-1} \sum_{j=1}^{m} \left(\phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)} \right)$$
$$= \frac{1}{\sigma} \left(\sum_{x=1}^{\sigma-1} \sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)} + \sum_{j=1}^{m} 1 \right)$$
$$= \frac{1}{\sigma - 1} \sum_{x=1}^{\sigma-1} \left(\frac{\sigma - 1}{\sigma} \sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)} + \frac{m}{\sigma} \right)$$

となる．すなわち，$\sigma - 1$ 回の畳み込み計算により正確なスコアベクトルが得られる．

加えて，スコアベクトルの i 番目の要素のサンプルを次式のように定義する．

$$s_i^{(\ell)} = \frac{\sigma - 1}{\sigma} \sum_{j=1}^{m} \phi_{x^{(\ell)}}(t_{i+j-1}) \cdot \overline{\phi_{x^{(\ell)}}(\sigma_j)} + \frac{m}{\sigma}$$

すると，推定値 $\hat{s}_i$ の期待値は，c_i に一致することは明らかである．次の補題により，推定値の分散の上界を与える．

補題 4 $1 \le x \le \sigma - 1$ に関する写像 ϕ_x を持ったアルゴリズムにおいて，スコアベクトルの推定値の分散の上界は，次式となる．

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

証明 4 補題 3 の証明と分散の基本的性質により，

$$V(s_i^{(\ell)}) \le \frac{(\sigma - 1)^2 (m - c_i)^2}{\sigma^2}$$

を得る．$|\Phi| = \sigma - 1$ であるので，推定値の分散 $V(\hat{s}_i)$ は，

$$\frac{(\sigma - 1 - k)}{(\sigma - 2) k} \cdot V(s_i^{(\ell)})$$

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 $\ell \in \{0, .., k-1\}$ **do begin**
  文字列 T の各文字を写像 ϕ_{x^ℓ} で変換し，T_ℓ とする．
    （T_ℓ は長さ $(1+\alpha)m$ の複素数列である．）
  文字列 P の各文字を写像 $\overline{\phi_{x^\ell}}$ で変換し，P_ℓ とする．
  P_ℓ の後ろに αm 個の 0 を付加する．
    （P_ℓ は長さ $(1+\alpha)m$ の複素数列である．）
  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
サンプルの合計 $\sum_\ell c_\ell$ を計算し，
$C(T, P)$ の一部として出力する．

図 2　確率アルゴリズム
Fig. 2 Randomized Algorithm

である．それゆえ，推定値の分散の上界は

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

となる． ∎

よって，次の定理を得る．

定理 2 スコアベクトルの推定値は，全写像 Φ 中から選んだ k 個のサンプルを用いた次式での計算で求められる．

$$\hat{s}_i = \frac{1}{k}\sum_{\ell=1}^{k}\left(\frac{\sigma-1}{\sigma}\sum_{j=1}^{m}\phi_{x^{(\ell)}}(t_{i+j-1})\cdot\overline{\phi_{x^{(\ell)}}(\sigma_j)} + \frac{m}{\sigma}\right)$$

この計算の結果得られる推定値の分散の上界は，

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

である．

以上により得られたアルゴリズムを図 2 に示す．

6. まとめと今後の課題

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，最適な写像の生成方法を提案した．本手法の写像総数は $|\Sigma| - 1$ であり，これは FFT を用いる場合の理論的な下限である．本写像を用いることにより，アルファベット Σ 上の文字列に関して，長さ n のテキスト T と長さ m のパターン P の一致のスコアベクトルの推定値 $C(T, P) = (c_1, \ldots, c_i, \ldots, c_{n-m+1})$ は計算量 $O(kn \log m)$ で求められ，その推定値の分散は，サンプル数 k について

$$(\sigma-1)^2(\sigma-1-k)(m-c_i)^2/\sigma^2(\sigma-2)k$$

で抑えられる．また，サンプル数 k の上限は $|\Sigma| - 1$ であるので，本写像を用いる場合には必要に応じて，あるいは計算力に応じて，正確なスコアベクトルの計算も可能である．

一方，提案手法の有効性に関しては，推定値の分散の理論的な上界を与えるに留まった．今後，本手法の実装・実験を通して，その有効性を明らかにして行く予定である．

[文献]

[1] Crochemore, M. and Rytter, W.: “Text Algorithms”, Oxford University Press, New York. (1994).
[2] Gusfield, D.: “Algorithms on Strings, Trees, and Sequences”, Cambridge University Press, New York. (1997).
[3] Crochemore, M. and Rytter, W.: “Jewels of Stringology”, World Scientific, (2003).
[4] 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), pp.113–125 (1974).
[5] Abrahamson, K.: “Generalized string matching”, SIAM Journal on Computing 16, 6, pp.1039–1051 (1987).
[6] Karloff, H.: “Fast algorithms for approximately counting mismatches”, Information Processing Letters 48, 2, pp.53–60 (1993).
[7] Amir, A., Lewenstein, M. and Porat, E.: “Faster algorithms for string matching with k mismatches”, Symposium on Discrete Algorithms, pp.794–803 (2000).
[8] Atallah, M. J., Chyzak, F. and Dumas, P.: “A Randomized Algorithm for Approximate String Matching”, Algorithmica 29, pp.468–486 (2001).
[9] 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(1), pp.2–12 (2003).
[10] Nakatoh, T., Baba, K., Ikeda, D., Yamada, Y. and Hirokawa, S.: “An Efficient Mapping for Score of String Matching”, Journal of Automata, Languages and Combinatorics, Vol. 10, No. 5/6, pp.697-704 (2005).
[11] Baba, K., Tanaka, Y., Nakatoh, T. and Shinohara, A.: “A Generalization of FFT Algorithm for String Matching”, Proc. of International Symposium on Information Science and Electrical Engineering (ISEE2003), pp.191–194 (2003).

中藤 哲也　Tetsuya NAKATOH
九州大学情報基盤研究開発センター助教．1992 九州大学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．情報処理学会正会員．

馬場 謙介　Kensuke BABA
九州大学大学院システム情報科学研究院助教．九州大学システム LSI 研究センター助教．2002 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．情報処理学会正会員．

森 雅生　Masao MORI
九州大学 大学評価情報室 助教．1996 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，暗号プロトコルの検証と補完の研究に従事．情報処理学会正会員．

廣川 佐千男　Sachio HIROKAWA
九州大学情報基盤研究開発センター教授．1979 九州大学大学院理学研究科修士課程修了，理学博士．検索エンジン，Web マイニング，計算論理学の研究に従事．情報処理学会正会員．電子情報通信学会正会員．ACM 正会員．