An Efficient Mapping for Score of String Matching

Tetsuya Nakatoh[1], Kensuke Baba[2], Daisuke Ikeda[1], Yasuhiro Yamada[3], and Sachio Hirokawa[1]

[1] Computing and Communications Center, Kyushu University
Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, Japan
e-mail: {nakatoh,daisuke,hirokawa}@cc.kyushu-u.ac.jp

[2] PRESTO, Japan Science and Technology Corporation
Honcho 4-1-8, Kawaguchi City, Saitama 332-0012, Japan
e-mail: baba@i.kyushu-u.ac.jp

[3] Graduate School of Information Science and Electrical Engineering
Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, Japan
e-mail: yshiro@cc.kyushu-u.ac.jp

Abstract. This paper proposes an efficient algorithm to solve the problem of *string matching with mismatches*. For a text of length n and a pattern of length m over an alphabet Σ, the problem is known to be solved in $O(|\Sigma| n \log m)$ time by computing a score by the fast Fourier transformation (FFT). Atallah et al. introduced a randomized algorithm in which the time complexity can be decreased by the trade-off with the accuracy of the estimates for the score. The algorithm in the present paper yields an estimate with smaller variance compared to that the algorithm by Atallah et al., moreover, and computes the exact score in $O(|\Sigma| n \log m)$ time. The present paper also gives two methods to improve the algorithm and an exact estimation of the variance of the estimates for the score.

Keywords: string matching with mismatches, FFT, convolution, deterministic algorithm, randomized algorithm.

1 Introduction

String matching [4, 5] is the problem to obtain all the occurrences of a (short) string called a *pattern* in a (long) string called a *text*. We consider *string matching with mismatches* which allows inexact match introduced by substitution. Let Σ be an alphabet and δ the Kronecker function from $\Sigma \times \Sigma$ to $\{0, 1\}$, that is, for $a, b \in \Sigma$, $\delta(a, b)$ is 1 if $a = b$, 0 otherwise. The problem with mismatches is generally solved by computing the *score vector* $C(T, P)$ between a text $T = t_1 \cdots t_n$ and a pattern $P = p_1 \cdots p_m$ as follows:

$$C(T, P) = (c_1, \ldots, c_i, \ldots, c_{n-m+1}), \quad \text{where} \quad c_i = \sum_{j=1}^{m} \delta(t_{i+j-1}, p_j).$$

We can compute the score vector using the fast Fourier transform (FFT) in $O(n \log m)$ time, if the score vector is represented as a convolution, that is, if the Kronecker function is expressed by a product of two mappings from Σ to a set of numbers. This approach was developed by Fischer and Paterson [6] and is simply summarized in Gusfield [7]. However, practically, the time complexity of the algorithm depends on the number of alphabets. One of the reason for the difficulties is that the Kronecker function can not be written as a product of mappings directly. For example, if $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, the generalized algorithm in [7] needs three mappings ϕ_1, ϕ_2, and ϕ_3 which convert symbols into $\{1, 0\}$ as the following table.

	ϕ_1	ϕ_2	ϕ_3
a	1	0	0
b	0	1	0
c	0	0	1

Then, we have $\delta(a, b) = \sum_{\ell=1}^{3} \phi_\ell(a) \cdot \phi_\ell(b)$ and the score vector is obtained by computing the convolution $\sum_{j=1}^{m} \phi_\ell(t_{i+j-1}) \cdot \phi_\ell(p_j)$ for $1 \leq i \leq n$ three times.

Atallah et al. [1] introduced a randomized algorithm where the time complexity has a trade-off with the accuracy of the estimates for the score vector. In this algorithm, symbols are converted into complex numbers with a primitive σ-th root ω of unity and the Hermitian inner product is used for the convolution. Then, the score vector is obtained as the average of the results of convolutions with respect to all possible mappings φ_ℓ from Σ to $\{0, \ldots, |\Sigma| - 1\}$, that is,

$$c_i = \frac{1}{|\Phi|} \sum_{\ell=1}^{|\Phi|} \sum_{j=1}^{m} \omega^{\varphi_\ell(t_{i+j-1}) - \varphi_\ell(p_j)},$$

where Φ is the set of all mappings ϕ_ℓ. (A deterministic algorithm constructed by those mappings requires the computation of the convolution $|\Sigma|^{|\Sigma|}$ times.) An estimate for the score vector is the average of the results with respect to some mappings chosen independently and uniformly from Φ. Let k be the number of randomly chosen samples. Then, the time complexity is $O(kn \log m)$. They showed that the expectation of the estimates equals to the score vector and the variance is bounded by $(m - c_i)^2/k$. Baba et al. [2] improved this algorithm by simplifying the mappings which converts the strings into numbers. The codomain of the mappings is the set $\{-1, 1\}$ instead of the set of complex numbers. Then, the score vector is

$$c_i = \frac{1}{|\Phi|} \sum_{\ell=1}^{|\Phi|} \sum_{j=1}^{m} \phi_\ell(t_{i+j-1}) \cdot \phi_\ell(p_j).$$

Baba et al. [3] pointed out that the algorithms which compute the score vector by FFT are distinguished by the mappings which convert strings into numbers in each algorithm, and the exact score is obtained by repeating the $O(n \log m)$ operation $|\Phi|$ times.

In this paper, we propose an efficient algorithm to solve string matching in which the variance of the estimates is not greater than $(m - c_i)^2/k$. Moreover, the exact score vector is computed in $O(|\Sigma| n \log m)$ time. We also give a strict evaluation of the variance and introduce two methods to improve our algorithm.

2 Efficient Algorithm

We propose an efficient algorithm for string matching with mismatches. The time complexity of a deterministic algorithm and the variance of the estimates for the score vector are obtained by analyzing the mappings which convert the symbols to the numbers. Let p be the smallest prime number which is greater than or equal to the cardinality $|\Sigma|$ of the alphabet. The codomain of the mappings is the p-adic number field Z_p. Since such a prime number is less than $2|\Sigma| - 2$ (Chebyshev's theorem), a deterministic algorithm with this mappings computes the score vector between a text of length n and a pattern of length m in $O(|\Sigma| n \log m)$ time. Moreover, in the same way as the algorithm by Atallah et al, we can construct a randomized algorithm in which the variance of the estimates for the score vector is independent to $|\Sigma|$.

2.1 Efficient Mapping

Let φ be a bijective mapping from Σ to $\{0, 1, \cdots |\Sigma| - 1\}$. For $0 \leq x \leq p-1$ and $a \in \Sigma$, we define a mapping ϕ_x as

$$\phi_x(a) = \omega^{x \cdot \varphi(a)}, \tag{1}$$

where ω is a primitive p-th root of unity. Then, we have the following lemma.

Lemma 1 For any $a, b \in \Sigma$,

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

where $\overline{\omega^y} = \omega^{-y}$.

Proof. If $a = b$, we have $\phi_x(a) \cdot \overline{\phi_x(b)} = \omega^0 = 1$ for any $0 \leq x \leq p-1$. Hence, the right side of the equation is equal to 1. If $a \neq b$, the difference $\varphi(a) - \varphi(b)$ is an element of $Z_p \backslash \{0\}$. Therefore, $x \cdot (\varphi(a) - \underline{\varphi(b))}$ is valued $0, \ldots, p-1$ modulo p for $0 \leq x \leq p-1$. Thus, we have $\sum_{x=0}^{p-1} \phi_x(a) \cdot \overline{\phi_x(b)} = \sum_{x=0}^{p-1} \omega^{x \cdot (\varphi(a) - \varphi(b))} = 0$. □

Lemma 2 By using the mapping ϕ_x, the score vector between a text of length n and a pattern of length m over an alphabet Σ can be computed in $O(|\Sigma| n \log m)$ time.

Proof. By the definition of the score vector and Lemma 1, the score vector is

$$c_i = \frac{1}{p} \sum_{x=0}^{p-1} \sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)}. \tag{2}$$

Therefore, the score vector is obtained by computing the convolution

$$f(i) = \sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)} \quad (1 \leq i \leq n)$$

p times. Since $p = O(|\Sigma|)$, we have the lemma. □

2.2 Analysis of Variance

In the same way as the algorithm by Atallah et al. [1], we can construct a randomized algorithm in which an estimate for the score vector is obtained by choosing some mappings from Φ. We define a *sample* s_i of an element c_i of the score vector to be

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

Let k be the number of chosen samples. Then, an *estimate* $\hat{s}_i$ for the element c_i of the score vector is defined by

$$\hat{s}_i = \frac{1}{k} \sum_{\ell=1}^{k} s_i.$$

By Eq. (2), it is clear that the mean of the estimates is equal to c_i. The following lemma gives the upper-bound of the variance of the estimates.

Lemma 3 In a randomized algorithm constructed with the mapping ϕ_x, the variance of the estimates for the score vector is bounded by $(m - c_i)^2/k$.

Proof. We denote by $V(X)$ the variance of a random variable X. By the definition of the estimate and the basic property of variance, we have $V(\hat{s}_i) = V(s_i)/k$. Since $\phi_{x(\ell)}(a) \cdot \overline{\phi_{x(\ell)}(a)} = 1$ and $|\phi_{x(\ell)}(a) \cdot \overline{\phi_{x(\ell)}(b)}| = 1$ for any $1 \le \ell \le |\Phi|$ and any $a, b \in \Sigma$, the variance of the samples is $V(s_i) = \sum_{\ell=1}^{|\Phi|}(\sum_{j=1}^{m} \phi_{x(\ell)}(t_{i+j-1}) \cdot \overline{\phi_{x(\ell)}(p_j)} - c_i)^2/|\Phi| \le (m - c_i)^2$. □

2.3 Description of Algorithm

We describe the algorithm which uses the mapping ϕ_x in detail. The input is a text string $T = t_1 \cdots t_n$, a pattern string $P = p_1 \cdots p_m$ over an alphabet Σ, and a number k of iterations in this algorithm. The output is an estimate for the score vector $C(T, P)$ if $k < p$, the exact score vector if $k = p$, where p is the smallest prime number such that $|\Sigma| \le p$. By the standard technique [4] of partitioning the text, we can assume $n = (1 + \alpha)m$ for $\alpha = O(m)$. The algorithm is constructed by iterations of the following operations.

- convert the text into a numerical sequences $\phi_x(T) = \omega^{\varphi_x(t_1)} \cdots \omega^{\varphi_x(t_{(1+\alpha)m})}$ by the mapping ϕ_x from Σ to $\{\omega^0, \ldots, \omega^{p-1}\}$;
- convert the pattern into $\overline{\phi_x(P)} = \omega^{-\varphi_x(p_1)} \cdots \omega^{-\varphi_x(p_m)}$ by ϕ_x and pad with αm zeros;
- compute the sample s_i for $1 \le i \le (1 + \alpha)m$ as the convolution of $\phi_x(T)$ and the reverse of the padded $\overline{\phi_x(P)}$ by FFT.

The output is computed as the average of the results of the convolution for $1 \le x \le k$. If $k = p$, by Lemma 2, the output is equal to the score vector. If $k < p$, the output is regarded as an estimate for the score vector obtained by a randomized algorithm with "sampling without replacement". Therefore, by Lemma 3 the variance of the estimates is $((p - k)/(p - 1)) \cdot (V(s_i)/k)$.

Theorem 1 By the algorithm with the mapping ϕ_x, the exact score between a text of length n and a pattern of length m over an alphabet Σ is computed in $O(|\Sigma|n \log m)$ time. Moreover, an estimate for the score vector is computed in $O(kn \log m)$ time, where k is the number of iterations in the algorithm and the variance of the estimates is bounded by $(p-k)(m-c_i)^2/(p-1)k$.

In generally, the variance of the estimates obtained by sampling without replacement is

$$\frac{|\Phi|-k}{|\Phi|-1} \cdot V(\hat{s}_i)$$

where Φ is the set of all mappings which convert symbols into numbers. The cardinality $|\Phi|$ of the set is $|\Sigma|^{|\Sigma|}$ in the algorithm by Atallah et al [1]. and $2^{|\Sigma|}$ in one by Baba et al [2]. Hence, the finite-size correction term $(|\Phi|-k)/(|\Phi|-1)$ is not so effective.

A key distinguishing feature of our algorithm is that the exact score can be computed in a practical time. Since $|\Phi|$ is large in the two randomized algorithms, their deterministic versions constructed in a similar way as our algorithm are not practical for a large alphabet. Although the deterministic algorithm generalized by Gusfield [7] can be extended to a randomized algorithm in the same way as our algorithm, the variance of the estimates depends on the number of alphabets.

3 Improvement of Algorithm

We propose two techniques to improve the algorithm in the previous section.

3.1 Removal of Defective Mapping

Our mappings convert the different symbols to the distinct numerical values. But only the mapping ϕ_0 converts all symbols to 0. Therefore, we remove the mapping ϕ_0 from the set Φ. That is possible without computing convolution.

By Eq. (1), $\delta(a,b) = \frac{1}{p}\sum_{x=0}^{p-1} \phi_x(a) \cdot \overline{\phi_x(b)} = \frac{1}{p}(\sum_{x=1}^{p-1} \phi_x(a) \cdot \overline{\phi_x(b)} + \phi_0(a) \cdot \overline{\phi_0(b)}) = \frac{1}{p}(\sum_{x=1}^{p-1} \phi_x(a) \cdot \overline{\phi_x(b)} + 1)$. Therefore, the score vector is $c_i = \sum_{j=1}^{m} \frac{1}{p}(\sum_{x=1}^{p-1} \phi(t_{i+j-1}) \cdot \overline{\phi(p_j)} + 1) = \frac{1}{p}\sum_{x=1}^{p-1}\sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)} + \frac{m}{p}$. To randomize the computation of c_i, we define c'_i as follows: $c'_i = \frac{1}{p-1}\sum_{x=1}^{p-1}\sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)}$. Hence, $c_i = \frac{p-1}{p}c'_i + \frac{m}{p}$.

We define a sample s'_i of an element c'_i to be

$$s'_i = \sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)}.$$

And an estimate $\hat{s'_i}$ is defined by

$$\hat{s'_i} = \frac{1}{k}\sum_{\ell=1}^{k}\sum_{j=1}^{m} \phi_x(t_{i+j-1}) \cdot \overline{\phi_x(p_j)}$$

where $1 \leq k \leq p-1$.

And an estimate $\hat{s_i}$ for the element c_i of the score vector is defined by

$$\hat{s_i} = \frac{p-1}{p}\frac{1}{k}\sum_{\ell=1}^{k}\sum_{j=1}^{m}\phi_x(t_{i+j-1})\cdot\overline{\phi_x(p_j)} + \frac{m}{p} \tag{3}$$

where $1 \leq k \leq p-1$.

By the difinition of a variance, $V(s_i) = \frac{(p-1)^2}{p^2}V(s_i')$. Moreover, because the number of mappings decrease by one, the variance in consideration of that is bounded by

$$\frac{(p-1)^2}{p^2}\cdot\frac{p-1-k}{p-2}\cdot\frac{(m-c_i)^2}{k}. \tag{4}$$

3.2 Removal of Imaginary Part

The magnitude of $\phi_x(a)\cdot\overline{\phi_x(b)}$ in Eq. (1) is 1. We used this magnitude for the analysis of the variance until this point. However, the real part is independent of the imaginary part. Therefore, those parts of Eq. (1) can be computed separately.

Let $\Re(v)$ be a real part of a complex number v. By Lemma 1, $\frac{1}{p}\sum_{x=0}^{p-1}\phi_x(a)\cdot\overline{\phi_x(b)}$ returns 0 or 1. Therefore, we can remove the imaginary part. Then, $\delta(a,b) = \Re(\frac{1}{p}\sum_{x=0}^{p-1}\phi_x(a)\cdot\overline{\phi_x(b)})$ for any $a, b \in \Sigma$. By the definition of the score, $c_i = \sum_{j=1}^{m}\Re(\frac{1}{p}\sum_{x=0}^{p-1}\phi_x(t_{i+j-1})\cdot\overline{\phi_x(p_j)})$. Since the order of addition is not restricted, the score vector is

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

The computation of the complex number is necessary to compute convolution with FFT. We only have to omit the imaginary part after the computation of FFT. By this omission, the computation of both the sum of the imaginary part and the magnitude of complex number become unnecessary.

The variance is the poorest when inconsistent $m-c$ characters are each a kind of symbol on the text and the pattern. In such a case, $\phi_\ell(a)\cdot\overline{\phi_\ell(b)}$ is fixed without influence of j. By Eq. (1), $\Re(\phi_x(a)\cdot\overline{\phi_x(b)}) = \cos\theta_\ell$, where $\theta_\ell = \frac{2\pi x\cdot(\varphi(a)-\varphi(b))}{p}$. Then, the random variable s_i is following.

$$s_i = \sum_{j=1}^{m}\Re(\phi_\ell(a)\cdot\overline{\phi_\ell(b)}) = \sum_{j=1}^{m}\cos\theta_\ell = c_i\cos 0 + (m-c_i)\cos\theta_\ell = c_i + (m-c_i)\cos\theta_\ell.$$

The variance $V(s_i)$ of this random variable s_i are followings.

$$\begin{aligned}
V(s_i) &= \sum_{\ell=1}^{p}(c_i + (m-c_i)\cos\theta_\ell - c_i)^2\cdot\frac{1}{p} \\
&= \frac{1}{p}\sum_{\ell=1}^{p}((m-c_i)\cos\theta_\ell)^2 \\
&= \frac{1}{p}(m-c_i)^2\sum_{\ell=1}^{p}\cos^2\theta_\ell \\
&= \frac{(m-c_i)^2}{p}\sum_{\ell=1}^{p}\frac{1+\cos\theta_\ell}{2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{(m-c_i)^2}{2p}(\sum_{\ell=1}^{p} 1 + \sum_{\ell=1}^{p} \cos\theta_\ell) \\
&= \frac{(m-c_i)^2}{2p}(p+0) \\
&= \frac{(m-c_i)^2}{2} \qquad (5)
\end{aligned}
$$

By $V(\hat{s}_i) = V(s_i)/k$, the variance of the estimates $\hat{s}_i$ is bounded by

$$\frac{(m-c_i)^2}{2k}. \qquad (6)$$

3.3 Variance of Improved Algorithm

We showed two improvement points. That both can be applied to the basic algorithm at a time.

Now, the change point of the algorithm from the basis one shown in Subsection 2.3 is showed in the followings.

- We remove ϕ_0, and choose a sample from the remaining mappings.
- An estimate $\hat{s}'_i$ is computed using that samples.
- Only a real part is used for a computation of an estimate from the result of FFT.
- We compute $\hat{s}_i$ by Eq. (3), and make it the estimate of c_i.

When these improvements are applied, by Eq. (4) and Eq. (6), the variance of the estimates is bounded by

$$\frac{(p-1)^2}{p^2} \cdot \frac{p-1-k}{p-2} \cdot \frac{(m-c_i)^2}{2k}.$$

It is smaller than one in the algorithm of Section 2.

4 Exact Estimation of Variance

Atallah et al. presented an upper bound of the variance of the estimates for the score in their algorithm as $(m-c_i)^2/k$. The reason for this variance is that their set of mappings includes many mappings which convert some different symbols into same numerical value. One of the features of our mappings is that it does not convert some different symbols into same numerical value because a single exceptional mapping was removed in Subsection 3.1. Using this feature, we give an exact estimation of the variance based on our mappings.

Let a, b be symbols in Σ. If a product $\phi(a) \cdot \overline{\phi(b)}$ in one position is independent of it in other position, the estimate of $\sum_j^{(m-c_i)} \phi_x(t_j) \cdot \overline{\phi_x(p_j)}$ is 0. The two following conditions must be satisfied for that. One of those conditions is that a symbol in one position is independent of symbols in other positions. In this paper, we suppose that condition. The independence can not be expected in the general English text much. But, we expect high independence about the comparison of the product $\phi(a) \cdot \overline{\phi(b)}$.*

*In this paper, we did not get to the verification of that point. It is a future work.

Another condition is the following lemma.

Lemma 4 If all mappings convert different symbols into distinct numerical values, then the product $\phi(a) \cdot \overline{\phi(b)}$ in one position is independent of that in other position.

Proof. Let t_1, t_2, p_1, p_2 be symbols in Σ, x a value which can be returned by mappings and r the number of kinds of x. Let Φ_x be a set of the mappings which convert more than one of some symbols into x, and Φ_{xy} denotes $\Phi_x \cap \Phi_y$. We define D_x as the difference between the number of x which the mappings convert a given symbol into and the number of mappings used for it. The number of certain value x which a certain symbol a convert to is $\frac{|\Phi|}{r}$ because $\sum_{\ell=1}^{|\Phi|} \phi_\ell(a) = 0$. Then, the number of certain value x which all the symbols convert to is Φ. Therefore, $|\Phi_x| = |\Phi| - D_x$. In the mapping that converts the different symbols to the distinct numerical values, Φ_x equal to Φ.

Pr(X) denotes the probability of event X. Let A be the event $\phi(t_1) \cdot \overline{\phi(p_1)} = x$ and B the event $\phi(t_2) \cdot \overline{\phi(p_2)} = x$. And let A' be the event $\phi(t_1) = d_1$, A'' the event $\phi(p_1) = d_2$, B' the event $\overline{\phi(t_2)} = d_3$, and B'' the event $\overline{\phi(p_2)} = d_4$.

If a certain event occurred, that a result of a mapping was value x, the mapping in the next event is restricted to mappings which return value x. After the event A, a set of mappings is $\Phi_{d_1 d_2}$ because the mapping returned d_1 and d_2 were used in the event A. A probability that a mapping return a value x is (the number of combinations of the mapping and the symbol which can return x)/(the product of the number of mappings and the number of symbols). Then we have

$$\begin{aligned}
\Pr(B') &= \frac{\frac{1}{r} \cdot |\Phi| \cdot |\Sigma|}{|\Phi| \cdot |\Sigma|} = \frac{1}{r}, \\
\Pr(B''|B') &= \frac{\frac{1}{r} \cdot |\Phi| \cdot |\Sigma|}{|\Phi_{d_3}| \cdot |\Sigma|} = \frac{|\Phi|}{r \cdot |\Phi_{d_3}|}, \\
\Pr(B) &= \sum_{d_3=0}^{r-1} \Pr(B') \Pr(B''|B') = \sum_{d_3=0}^{r-1} (\frac{1}{r} \cdot \frac{|\Phi|}{r \cdot |\Phi_{d_3}|}) = \frac{1}{r^2} \sum_{d_3=0}^{r-1} (\frac{|\Phi|}{|\Phi_{d_3}|}),
\end{aligned}$$

and

$$\Pr(B|A) = \sum_{d_3=0}^{r-1} (\frac{|\Phi|}{r \cdot |\Phi_{d_1 d_2}|} \cdot \frac{|\Phi|}{r \cdot |\Phi_{d_1 d_2 d_3}|}) = \frac{1}{r^2} \sum_{d_3=0}^{r-1} (\frac{|\Phi|^2}{|\Phi_{d_1 d_2}| \cdot |\Phi_{d_1 d_2 d_3}|}).$$

We get $\Pr(B|A) \neq \Pr(B)$, hence $\phi(t_1) \cdot \overline{\phi(p_1)}$ is not independent of $\phi(t_2) \cdot \overline{\phi(p_2)}$. However, if $\Phi = \Phi_{d_1 d_2} = \Phi_{d_1 d_2 d_3}$, then $\Pr(B|A) = \Pr(B)$. This condition is satisfied only when all mappings should convert different symbols into distinct numerical values. □

Other two mappings can not satisfy the condition of Lemma 4 while only our mappings can satisfy it in case of $|\Sigma| = p$. Therefore, we add a dummy symbol in case of $|\Sigma| < p$. Then we can correct a sampling bias because we can know that by the dummy symbol in advance.

When ϕ_ℓ is drawn uniformly randomly from Φ, the random variable $\hat{s}$ is $\hat{s} = \frac{1}{k} \sum_{\ell=1}^{k} \sum_{j=1}^{m} \phi_\ell(t_j) \cdot \overline{\phi_\ell(p_j)}$.

Then, we get the following lemma.

Lemma 5 Given that the product $\phi(a)\cdot\overline{\phi(b)}$ in one position is independent of that in other position. When c symbols align in the m symbols, the variance $V(\hat{s})$ of random variable s are

$$V(\hat{s}) = \frac{m-c_i}{k}.$$

Proof. Let s_j be the random variable as $\phi_\ell(t_j)\cdot\overline{\phi_\ell(p_j)}$, then $s_j = \phi_\ell(t_j)\cdot\overline{\phi_\ell(p_j)} = \omega^{d(t_j,p_j)}$ where $d(t_j,p_j) = x\cdot(\psi(t_j)-\psi(p_j))$. $s_{(t_j=p_j)}$ denotes that s in $t_j = p_j$ and $s_{(t_j\neq p_j)}$ denotes that s in $t_j \neq p_j$.

If $t_j = p_j$, $s_j = 1$. If $t_j \neq p_j$, $s_j = \omega^{d(t_j,p_j)}$. Then, those means are $E(s_{(t_j=p_j)}) = 1, E(s_{(t_j\neq p_j)}) = \sum_{x=0}^{p-1}\omega^{d(t_j,p_j)}\cdot\frac{1}{p} = 0$. And those variance are $V(s_{(t_j=p_j)}) = (s_{(t_j=p_j)} - E(s_{(t_j=p_j)}))^2\cdot 1 = (1-1)^2\cdot 1 = 0, V(s_{(t_j\neq p_j)}) = \sum_{x=0}^{p-1}(s_{(t_j\neq p_j)} - E(s_{(t_j\neq p_j)}))^2\cdot\frac{1}{p} = \frac{1}{p}\sum_{x=0}^{p-1}(|\omega^{d(t_j,p_j)}|)^2 = \frac{1}{p}\sum_{x=0}^{p-1}1 = 1$.

Because we assume that the product $\phi(a)\cdot\overline{\phi(b)}$ in one position is independent of that in other position, a variance $V(s)$ of s are the simple total of a variance of every position. Then, $V(s) = \sum^{c}V(s_{(t_j=p_j)}) + \sum^{m-c_i}V(s_{(t_j\neq p_j)}) = \sum^{c}0 + \sum^{m-c_i}1 = m-c_i$.

Using k samples s, a variance $V(\hat{s})$ of the estimate s is $V(\hat{s}) = \frac{1}{k}V(s)$. Then

$$V(\hat{s}) = \frac{m-c_i}{k}.$$

□

This analysis can be applied to the algorithm which improvement in Section 3 was added to.

Then Eq. (5) changes as follow,

$$\begin{aligned}
V(s_{j(t_j\neq p_j)}) &= \sum_{x=0}^{p-1}(s_{j(t_j\neq p_j)} - E(s_{j(t_j\neq p_j)}))^2\frac{1}{p}\\
&= \frac{1}{p}\sum_{x=0}^{p-1}(\cos\frac{2\pi g(a,b)}{p} - 0)^2\\
&= \frac{1}{p}\sum_{x=0}^{p-1}\cos^2\frac{2\pi g(a,b)}{p}\\
&= \frac{1}{p}\sum_{x=0}^{p-1}\frac{1+\cos\frac{2\pi g(a,b)}{p}}{2}\\
&= \frac{1}{2p}(\sum_{x=0}^{p-1}1 + \sum_{x=0}^{p-1}\cos\frac{2\pi g(a,b)}{p})\\
&= \frac{1}{2} \qquad (7)
\end{aligned}$$

By Eq. (7), we analyze the variance as the proof of Lemma 5.

$$V(\hat{s}) = \frac{m-c_i}{2k}. \qquad (8)$$

By Eq. (4) and Eq. (8), we get the following theorem.

Theorem 2 The variance of the estimates for the score in our algorithm is

$$V(\hat{s}) = \frac{(p-1)^2}{p^2}\cdot\frac{p-1-k}{p-2}\cdot\frac{m-c_i}{2k}.$$

Conclusion

We gave an efficient randomized algorithm for string matching with mismatches. This randomized algorithm uses convolution with FFT, like that proposed by Atallah et al. and Baba et al. We used the mappings which convert the symbols to the p-adic number field. One of the features of our mappings is that it does not convert some different symbols into same numerical value. By that feature, the variance of the estimate of the score vector is smaller. The other feature of our mappings is that there are not so many mappings. The number of mapping is $p-1$ where $|\Sigma| \leq p < 2|\Sigma|-2$.

We analyzed the variance of the estimates for the score in this algorithm. And it is very small as compared to the randomized algorithms proposed in the past. The variance in this algorithm is $\frac{(p-1)^2}{p^2} \cdot \frac{p-1-k}{p-2} \cdot \frac{m-c_i}{2k}$. Its time complexity is $O(kn \log m)$ where k is the number of samples, and the upper bound of k is $p-1$. When k is $p-1$, this algorithm is deterministic, and the estimate becomes the real value.

Experiments with read texts and the evaluation of computation time are future work. We have a plan to apply the method for pattern extraction from Web pages [8].

References

[1] Atallah, M. J., Chyzak, F., and Dumas, P.: A Randomized Algorithm for Approximate String Matching. Algorithmica 29, 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 10, 2-12. 2003.

[3] Baba, K., Tanaka, Y., Nakatoh, T., Shinohara, A.: A Unification of FFT Algorithm for String Matching. Proc. International Symposium on Information Science and Electrical Engineering 2003, to appear.

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

[5] Crochemore, M. and Rytter, W.: Jewels of Stringology. World Scientific. 2003.

[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] Taguchi, T., Koga, Y. and Hirokawa, S.: Integration of Search Sites of the World Wide Web. Proc. of International Forum cum Conference on Information Technology and Communication, Vol. 2, pp. 25-32, 2000.