A Generalization of FFT Algorithms for String Matching

Baba, Kensuke
Faculty of Information Science and Electrical Engineering, Kyushu University

Tanaka, Yoshihito
Faculty of Information Science and Electrical Engineering, Kyushu University

Nakatoh, Tetsuya
Faculty of Information Science and Electrical Engineering, Kyushu University

Shinohara, Ayumi
Faculty of Information Science and Electrical Engineering, Kyushu University

https://hdl.handle.net/2324/15559

出版情報：Proceedings of the International Symposium on Information Science and Electrical Engineering. 2003, pp.191-194, 2003-11. Kyushu University
バージョン：
権利関係：

A Generalization of FFT Algorithms for String Matching[*]

Kensuke Baba[†] Yoshihito Tanaka Tetsuya Nakatoh Ayumi Shinohara

Abstract

There exists an algorithm which solves string matching problem with mismatches by computing a vector by the fast Fourier transformation (FFT), however, the time complexity depends on the size of the alphabet. Atallah *et al.* introduced a randomized algorithm in which the time complexity has a trade-off with the accuracy of the estimates for the vector and it was improved by Baba *et al.* This paper generalize these three algorithms in terms of the functions which convert characters into numbers. The generalization provides that the exact vector is obtained by repeating the FFT computation at least $\sigma - 1$ times, where σ is the size of the alphabet. Moreover, it gives the exact variance of the estimates for the vector.

1 Introduction

String matching is to find all occurrences of a pattern string in a text string. It is important for many kinds of processes of strings in biology, web-mining, and so on. The problem which allows substitution to introduce the variations of a pattern is called *string matching with mismatches.* Refer the textbooks [3, 4, 6] to know the history and various results. String matching with mismatches is generally solved by computing the vector such that for given a text $T = t_1 \cdots t_n$ and a pattern $P = p_1 \cdots p_m$, its i-th element is the number of matches between the substring $t_i \cdots t_{i+m-1}$ of the text T and the pattern P.

The vector can be computed in $O(n \log m)$ time by the fast Fourier transformation (FFT) while a naive algorithm takes $O(mn)$ time. This approach was essentially developed by Fischer and Paterson [5] and a generalized algorithm on this approach is described simply in [6]. In the generalized algorithm, two strings are converted into binary strings with respect to each character in the alphabet for the computation of FFT. Hence, practically, the time complexity of the algorithm depends on the size of the alphabet.

Atallah *et al.* [1] introduced a randomized algorithm of which time complexity is $O(kn \log m)$, where k is the number of samples. In this algorithm, a text and a pattern are converted into two sequences of complex numbers by a function from the alphabet to a set of complex numbers. The expectation of the estimates for the vector is equal to the exact value and the variance is independent of the size of the alphabet. Baba *et al.* [2] improved the algorithm by simplifying the function which converts characters into numbers. The range of the function is the set $\{-1, 1\}$ instead of the set of complex numbers, and the variance of the estimates is still independent of the size of the alphabet.

In this paper, we generalize the algorithms for string matching with mismatches which use FFT and analyze its deterministic version and randomized version. Each algorithm is distinguished by corresponding functions which convert characters into numbers. By this generalization, we show that, in the deterministic version, at least $\sigma - 1$ times repetition of the $O(n \log m)$ computation of the FFT is necessary, where σ is the size of the alphabet. As to the randomized version, we analyze the variance of the estimates generally, which implies that the two randomized algorithms in [1] and [2] give the same variance.

2 Preliminaries

Let Σ be an alphabet. An element of Σ^* is called a *string.* We denote the size of a set S by $|S|$. Let $\sigma = |\Sigma|$. The i-th character of a string w is denoted by w_i and i is called its *position* on w.

The *score vector* between a text string and a pattern string is the vector whose i-th element is the number of matches between the text and the pattern when the first character of the pattern is positioned at the i-th character of the text. Let δ be the Kronecker function from $\Sigma \times \Sigma$ to $\{0, 1\}$, that is, for $a, b \in \Sigma$, $\delta(a, b)$ is 1 if $a = b$, and 0 otherwise. For a text string $T = t_1 \cdots t_n$ and a pattern string $P = p_1 \cdots p_m$, the score vector $C(T, P)$ between T and P is $(c_1, \ldots, c_i, \ldots, c_{n-m+1})$, where $c_i = \sum_{j=1}^{m} \delta(t_{i+j-1}, p_j)$.

The *convolution* $f(i) = \sum_{j=1}^{m} g(j) \cdot h(i - j)$ for $1 \le i \le m$ of two functions g and h can be computed in $O(m \log m)$ time by FFT. Therefore, the score vector between T and P each of length m can be computed in $O(m \log m)$ time if the Kronecker function is expressed

[*] An edited version of this report was published in: *Proc. International Symposium on Information Science and Electrical Engineering 2003 (ISEE 2003), pp.191–194, Kyushu University, Nov, 2003.*

[†] Faculty of Information Science and Electrical Engineering, Kyushu University, *baba@i.kyushu-u.ac.jp*

by (a linear function of) a product of two functions from Σ to a set of numbers. We additionally apply the standard technique [3] of partitioning the text, then the previous argument is extended for T of length $n > m$. We part T into overlapping chunks each of size $(1+\alpha)m$ and process each chunk separately. Processing one chunk gives αm components of C. Since we have $n/\alpha m$ chunks and each chunk can be computed in $O((1+\alpha)m\log((1+\alpha)m))$ by FFT, the total time complexity is $(n/\alpha m)\cdot O((1+\alpha)m\log((1+\alpha)m)) = O(n\log m)$ by choosing $\alpha = O(m)$.

In the rest of this section, we describe briefly the three algorithms, that is, a deterministic algorithm generalized by Gusfield and two randomized algorithms by Atallah *et al.* and by Baba *et al.*

Standard Algorithm A standard algorithm which computes the exact score vector by FFT is summarized in [6]. In the algorithm, a text string and a pattern string are converted into binary strings with respect to each character and the score vector is the sum of all results of convolutions. For example, when $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, the functions ϕ_1, ϕ_2, ϕ_3 which convert the characters into $\{1, 0\}$ are the following:

	ϕ_1	ϕ_2	ϕ_3
a	1	0	0
b	0	1	0
c	0	0	1

Then, $\delta(a,b) = \sum_{\ell=1}^{3}\phi_\ell(a)\cdot\phi_\ell(b)$. Therefore, the i-th element c_i of the score vector between $T = t_1\cdots t_n$ and $P = p_1\cdots p_m$ is $c_i = \sum_{\ell=1}^{3}\sum_{j=1}^{m}\phi_\ell(t_{i+j-1})\cdot\phi_\ell(p_j)$. Thus, 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 $1 \le i \le n$ three times. In general, the i-th element is

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

and the time complexity is $O(\sigma n\log m)$ since $|\Phi| = \sigma$.

Algorithm by Atallah et al. Atallah *et al.* introduced a randomized algorithm to compute the score vector. A function ϕ_ℓ which converts characters into numbers is defined to be $\phi_\ell(a) = \omega^{\varphi_\ell(a)}$ for $a \in \Sigma$, where φ_ℓ is a functions from Σ to the set $\{0,\ldots,\sigma-1\}$ and ω is the primitive σ-th root of unity. Then, the exact score vector is the average of the results for all possible φ_ℓ, that is,

$$c_i = \frac{1}{|\Phi|}\sum_{\ell=1}^{|\Phi|}\sum_{j=1}^{m}\phi_\ell(t_{i+j-1})\cdot\overline{\phi_\ell(p_j)}, \tag{2}$$

where Φ is the set of all functions ϕ_ℓ and $\overline{\omega^x} = \omega^{-x}$. Since $|\Phi| = \sigma^\sigma$, the exact score vector is computed in $O(\sigma^\sigma n\log m)$. An estimate for the score vector is the average with respect to some functions chosen independently and uniformly from Φ. Therefore, the expectation of the estimates is equal to the score vector and the variance is independent of the size of an alphabet.

Theorem 1 ([1]) *An estimate for the score vector between a text of length n and a pattern of length m can be computed in time $O(kn\log m)$, where k is the number of iterations in the algorithm. The mean of the estimation equals to the score vector, and the variance of each entry is bounded by $(m-c_i)^2/k$.*

Algorithm by Baba et al. Another randomized algorithm is given in [2]. This algorithm can be regarded as a simplification of the previous one. The codomain of the function ϕ_ℓ is simply the set $\{-1, 1\}$ and $|\Phi| = 2^\sigma$. The score vector is

$$c_i = \frac{1}{|\Phi|}\sum_{\ell=1}^{|\Phi|}\sum_{j=1}^{m}\phi_\ell(t_{i+j-1})\cdot\phi_\ell(p_j). \tag{3}$$

Hence, the deterministic version runs in $O(2^\sigma n\log m)$ time. The randomized version is obtained in the same way and the variance of the estimations is bounded by $(m-c_i)^2/k$. Therefore, Theorem 1 is also shown by this algorithm.

3 Generalization of the Algorithms

We define a function ψ from Σ to the set of numerical vectors to be

$$\psi(a) = (\phi_1(a),\ldots,\phi_{|\Phi|}(a))$$

for $a \in \Sigma$. Then, ψ is expressed by σ vectors on $|\Phi|$-dimensional space, for example, the vectors in the example of the standard algorithm are $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$. The following lemma gives conditions to compute the score vector by FFT in terms of ψ.

Lemma 1 *The score vector between a text of length n and a pattern of length m can be computed by FFT in $O(dn\log m)$ time, if we have a function ψ from Σ to the d-dimensional vector space such that:*
(1) $\psi(a) \ne \psi(b)$ for any $a, b \in \Sigma$ $(a \ne b)$;
(2) $|\psi(a)|$ is constant for any $a \in \Sigma$;
(3) $\langle\psi(a)|\psi(b)\rangle$ is constant for any $a, b \in \Sigma$ $(a \ne b)$.

Proof. By (2) and (3), $\langle\psi(a)|\psi(a)\rangle = \alpha$ and $\langle\psi(a)|\psi(b)\rangle = \beta$ for any $a, b \in \Sigma$ $(a \ne b)$ with constants α and β. By (1), α is not equal to β. Hence, the Kronecker function is $\delta(a,b) = (\langle\psi(a)|\psi(b)\rangle - \beta)/(\alpha-\beta)$. Therefore, by the definition of the score vector,

$$\begin{aligned} c_i &= \sum_{j=1}^{m}\frac{1}{\alpha-\beta}(\langle\psi(t_{i+j-1})|\psi(p_j)\rangle - \beta) \\ &= \sum_{\ell=1}^{|\Phi|}\frac{1}{\alpha-\beta}\left(\sum_{j=1}^{m}\phi_\ell(t_{i+j-1})\cdot\phi_\ell(p_j) - \frac{\beta m}{|\Phi|}\right) \quad (4) \end{aligned}$$

Thus, the exact score vector is computed in $O(|\Phi| n \log m)$ time. □

This lemma implies that the number of the iterations of the $O(n \log m)$ computation is equal to the cardinality of the set Φ of all functions which convert characters into numbers. In the standard algorithms, $\alpha = 1$ and $\beta = 0$, hence Eq. (1) is obtained from Eq. (4). In the same way, we have Eq. (2) and Eq. (3) since $\alpha = |\Phi|$ and $\beta = 0$ in the two algorithms.

The generalization provides the lower-bound of the dimension of the vector space which is the range of the function ψ.

Theorem 2 *The exact score vector between a text of length n and a pattern of length m over an alphabet Σ is obtained by repeating the $O(n \log m)$ computation of FFT at least $\sigma - 1$ times.*

To prove the theorem, first, we show an algorithm to compute the exact score vector by repeating the FFT computation $\sigma - 1$ times. The precise procedure of the algorithm is constructed in the same way as one by Atallah et al., hence we show only the existence of $\sigma - 1$-vectors which satisfy the conditions in Lemma 1.

Let $\Sigma = \{a_1, \cdots, a_\sigma\}$. We define a function ψ from Σ to the $(\sigma - 1)$-dimensional vector space as follows:

$$\psi(a_r) = \begin{cases} (p_1, \cdots, p_{r-1}, q_r, 0, \cdots, 0) & 1 \le r < \sigma, \\ (p_1, \cdots, p_{\sigma-2}, -q_{\sigma-1}) & r = \sigma, \end{cases}$$

where

$$p_r = \sqrt{\frac{\sigma}{(\sigma - r + 1)(\sigma - r)}} \text{ and } q_r = -\sqrt{\frac{\sigma(\sigma - r)}{\sigma - r + 1}}.$$

Then, the following lemma is straightforward.

Lemma 2 *For $1 \le r, s \le \sigma$, $\delta(a_r, a_s) = (\langle \psi(a_r) | \psi(a_s) \rangle + 1)\sigma$.*

Next, we show that the dimension of any vectors which satisfy the conditions in Lemma 1 is at least $\sigma - 1$.

Proposition 1 *Let σ and d be natural numbers such that $\sigma \ge 3$ and $d < \sigma - 1$. Then, no σ vectors $v_1, \ldots, v_\sigma$ in d-dimensional vector space $\mathbf{R}^d$ satisfy the following three conditions:*
(1) $v_r \ne k v_s$ for any $1 \le r, s \le \sigma$ $(r \ne s)$ and $k \in \mathbf{R}$;
(2) for some constant α, $|v_r| = \alpha$ for any $1 \le r \le \sigma$;
(3) for some constant β, $\langle v_r | v_s \rangle = \beta$ for any $1 \le r, s \le \sigma$ $(r \ne s)$.

Proof. We first show that if $v_1, \ldots, v_\sigma \in \mathbf{R}^{\sigma-1}$ satisfy the conditions then $\beta = -\alpha^2/(\sigma - 1)$ and $v_1 + \cdots + v_\sigma = 0$. The case $\sigma = 3$ is easy. Let $u_1, \ldots, u_{\sigma-1}$ be $\sigma - 1$ vectors such that $u_i = v_i - \langle v_i | v_\sigma \rangle v_\sigma / |v_\sigma|^2 = v_i - \beta v_\sigma / \alpha^2$ $(1 \le i \le \sigma - 1)$. We check $u_1, \ldots, u_{\sigma-1}$ satisfy the conditions. Suppose $u_i = k u_j$. Then $v_i - \beta v_\sigma/\alpha^2 = k(v_j - \beta v_\sigma/\alpha^2)$. It is easy to see that $k \ne 1$. Then, $v_i = k v_j - (k-1)\beta v_\sigma/\alpha^2$. Since $\sigma \ge 4$, we can take v_l with $l \ne i, j, \sigma$. Then, $\beta = \langle v_i | v_l \rangle = \langle k v_j - (k-1)\beta v_\sigma/\alpha^2 | v_l \rangle = k\beta - (k-1)\beta^2/\alpha^2$. Since $v_1, \ldots, v_\sigma$ is in $(\sigma - 1)$-dimensional vector space, $\beta \ne 0$. Hence, $\beta = \alpha^2$. However, this is contradiction, since $v_1, \ldots, v_\sigma$ satisfy the first condition, again. The second and the third conditions follow immediately, since for any i and any $j \ne i$, $|u_i|^2 = \alpha^2 - \beta^2/\alpha^2$ and $\langle u_i | u_j \rangle = \beta - \beta^2/\alpha^2$. Since the σ-th elements of all $u_1, \ldots, u_{\sigma-1}$ are 0, they are in the $(\sigma - 2)$-dimensional vector space. Hence, by induction hypothesis, $\beta - \beta^2/\alpha^2 = \langle u_i | u_j \rangle = -|u_i|^2/(\sigma - 2) = -(\alpha^2 - \beta^2)/\alpha^2(\sigma - 2)$. Therefore, $\beta = -\alpha^2/(\sigma - 1)$, since $v_1, \ldots, v_\sigma$ satisfy the first condition. Moreover, $v_1 + \cdots + v_\sigma = \sum_{i=1}^{\sigma-1}(v_i - \beta v_\sigma/\alpha^2) = u_1 + \cdots + u_{\sigma-1} = 0$. This complete the proof of the claim. Now, we show the proposition. Suppose there exist σ-vectors $v_1, \ldots, v_\sigma$ in d-dimensional vector space. Then, by the claim, $v_1 + \cdots + v_{d+1} = 0$ and $v_2 + \cdots + v_{d+2} = 0$. Hence, $v_1 = v_{d+2}$, which is contradiction. □

By Lemmas 1 and 2 and Proposition 1, we have Theorem 2.

4 Randomized Algorithm

The deterministic algorithm described in Section 3 can be extended to a randomized version in which an estimate is provided by choosing some functions independently and uniformly from Φ. We define a *sample* $s_{i,\ell}$ of an element c_i of the score vector between a text $T = t_1 \cdots t_n$ and a pattern $P = p_1 \cdots p_m$ to be

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

for a chosen function ϕ_ℓ. Then, by Eq. (4), the estimate e_i obtained from k samples whose expectation equals to the score c_i is

$$e_i = \frac{|\Phi|}{k} \sum_{\ell=1}^{k} \frac{1}{\alpha - \beta} \left(s_{i,\ell} - \frac{\beta m}{|\Phi|} \right),$$

where $\alpha = |\psi(a)|$ and $\beta = \langle \psi(a) | \psi(b) \rangle$ $(a \ne b)$. By the basic property of variance, the variance of the estimates is $V(e_i) = |\Phi|^2 V(s_{i,\ell})/k(\alpha - \beta)^2$.

Since each c_i is defined in a similar way and we can assume $n = m$ without any loss of generality, we omit i in the rest of this paper.

In order to analyze the variance accurately, we define a function $\rho_{T,P}$ from $\Sigma \times \Sigma$ to the set of non-negative integer for each text $T = t_1 \ldots t_m$ and each pattern $P = p_1 \ldots p_m$ to be

$$\rho_{T,P}(a, b) = |\{j \mid t_j = a, p_j = b, 1 \le j \le m\}|.$$

We omit the subscription T, P of $\rho_{T,P}$ in the sequel.

Lemma 3 *The variance of the sample is*

$$\begin{aligned} V(s_\ell) &= \sum_{a,b\in\Sigma} \rho(a,a)\rho(b,b)\left(\xi(a,a,b,b) - \frac{\alpha^2}{|\Phi|^2}\right) \\ &+ 2\sum_{a,b,c\in\Sigma,b\neq c} \rho(a,a)\rho(b,c)\left(\xi(a,a,b,c) - \frac{\alpha\beta}{|\Phi|^2}\right) \\ &+ \sum_{a,b,c,d\in\Sigma,a\neq b,c\neq d} \rho(a,b)\rho(c,d)\left(\xi(a,b,c,d) - \frac{\beta^2}{|\Phi|^2}\right), \end{aligned}$$

where $\xi(a,b,c,d) = \sum_{\ell=1}^{|\Phi|} \phi_\ell(a)\cdot\phi_\ell(b)\cdot\phi_\ell(c)\cdot\phi_\ell(d)/|\Phi|$.

Proof. Let $E(s_\ell)$ be the mean of the samples. Then $V(s_\ell) = \sum_{\ell=1}^{|\Phi|} s_\ell^2/|\Phi| - E(s_\ell)^2$. By the definition of ρ,

$$\begin{aligned} \frac{1}{|\Phi|}\sum_{\ell=1}^{|\Phi|} s_\ell^2 &= \frac{1}{|\Phi|}\sum_{\ell=1}^{|\Phi|}\left(\sum_{a,b\in\Sigma}\phi_\ell(a)\phi_\ell(b)\rho(a,b)\right)^2 \\ &= \sum_{a,b,c,d\in\Sigma} \rho(a,b)\rho(c,d)\xi(a,b,c,d) \\ &= \sum_{a,b,c,d\in\Sigma,a=b,c=d} \rho(a,b)\rho(c,d)\xi(a,b,c,d) \\ &\quad + 2\sum_{a,b,c,d\in\Sigma,a=b,c\neq d} \rho(a,b)\rho(c,d)\xi(a,b,c,d) \\ &\quad + \sum_{a,b,c,d\in\Sigma,a\neq b,c\neq d} \rho(a,b)\rho(c,d)\xi(a,b,c,d). \end{aligned}$$

On the other hand,

$$\begin{aligned} E(s_\ell)^2 &= \left(\frac{\alpha-\beta}{|\Phi|}E(e) + \frac{\beta m}{|\Phi|}\right)^2 \\ &= \frac{\alpha^2}{|\Phi|^2}\sum_{a,b\in\Sigma}\rho(a,a)\rho(b,b) \\ &\quad + \frac{2\alpha\beta}{|\Phi|^2}\sum_{a,b,c\in\Sigma,b\neq c}\rho(a,a)\rho(b,c) \\ &\quad + \frac{\beta^2}{|\Phi|^2}\sum_{a,b,c,d\in\Sigma,a\neq b,c\neq d}\rho(a,b)\rho(c,d). \end{aligned}$$

□

Theorem 3 *In the randomized algorithms by Atallah et al. and by Baba et al., the variance of the estimates for the score vector is exactly equal to* $\sum_{a,b\in\Sigma,a\neq b}\rho(a,b)(\rho(a,b)+\rho(b,a))/k$.

Proof. In these algorithms, $\alpha = |\Phi|$ and $\beta = 0$. Since $\phi_\ell(a)^2 = 1$ for any $\phi_\ell \in \Phi$ and any $a \in \Sigma$, we have $\xi(a,a,b,b) = 1$ and $\xi(a,a,b,c) = 0$ for any $a,b,c \in \Sigma$ ($b \neq c$). Moreover, if $a \neq b$ and $c \neq d$, then $\xi(a,b,c,d)$ is 1 if either $a = c$ and $b = d$, or $a = d$ and $b = c$, and 0 otherwise. Therefore, by Lemma 3, the variance of the estimates is $V(s_\ell)/k = \sum_{a,b\in\Sigma,a\neq b}\rho(a,b)(\rho(a,b)+\rho(b,a))/k$. □

It should be remarked that the variance for the algorithm by Baba *et al.* is obtained by Lemma 6 in [2].

We can extend the standard algorithm and the algorithm in the proof of Theorem 2 to randomized versions. However, by Lemma 3, the variance of the estimates in each algorithm depends on the size of an alphabet.

5 Conclusion

We generalized three algorithms for the problem of string matching with mismatches which use FFT and analyze its deterministic version and randomized version. The generalization reduce the difference of each algorithm to the difference of the functions which convert characters into numbers. This observation provides that the exact score is obtained by repeating the computation by FFT at least $\sigma - 1$ time, where σ is the size of an alphabet, and the variances in the two randomized algorithms by Atallah *et al.* and by Baba *et al.* have the same value.

References

[1] M. J. Atallah, F. Chyzak, and P. Dumas. A randomized algorithm for approximate string matching. *Algorithmica*, 29:468–486, 2001.

[2] K. Baba, A. Shinohara, M. Takeda, S. Inenaga, and S. Arikawa. A note on randomized algorithm for string matching with mismatches. *Nordic Journal of Computing*, 10:2–12, 2003.

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

[4] M. Crochemore and W. Rytter. *Jewels of Stringology.* World Scientific, 2003.

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

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