2-Approximation Algorithms for Median and Centre String Problems

Y.H. Chen* C. Y. Tang Q. S. Wu

Abstract

Given a finite set of strings, the median string problem is to find a string that minimizes the sum of its Levenshtein distance to each string of the set. We also study the centre string problem that finds a string minimizing the maximum of distances to each string of the set . These problems are both shown to be NP-hard [4]. In this paper, we shall show a $2 - \frac{2}{n}$ *approximation algorithm on median string problem and a 2-approximation algorithm on centre string problem.*

Keywords: Levenshtein distance, median string, centre string

1 Introduction

An alphabet Σ is a non-empty set of symbols. Generally it will be assumed to contain at least two symbols. For a given alphabet Σ, the set of all finite-length strings of symbols from Σ is denote as Σ^*. For a string w, $|w|$ denotes the length of w. Given two strings w and w' in Σ^*, we say w rewrites into w' in one step if one of the following correction rules holds:

1. $w = axb, w' = ab$, and $a, b \in \Sigma^*, x \in \Sigma$ (single-symbol deletion x in w)
2. $w = ab, w' = axb$, and $a, b \in \Sigma^*, x \in \Sigma$ (single-symbol insertion x in w)
3. $w = axb, w' = ayb$, and $a, b \in \Sigma^*, x, y \in \Sigma, x \neq y$ (single-symbol substitution)

We use the notation $w \xrightarrow{k} w'$ to denote that w rewrites into w' by k operations of single-symbol deletion, single-symbol insertion, and single-symbol substitution.

Definition 1 *Given two strings w and w' , the Levenshtein distance between w and w' denoted as $d(w, w')$ is the smallest k such that $w \xrightarrow{k} w'$.*

Example: $d(xyxx, xxy) = 2$, because string $xyxx$ rewrites into xxy by a deletion of y from $xyxx$ and a substitution of last x by y.

Definition 2 *The global distance between a string w and a finite set of stings W over an alphabet Σ, which is denoted as $(D(w, W))$, is $D(w, W) = \sum_{w' \in W} d(w, w')$.*

Remarks. Levenshtein distances have the following characteristics:

1. $d(w_i, w_i) = 0$.
2. $d(w_i, w_j) = d(w_j, w_i)$.
3. $d(w_i, w_j) + d(w_i, w_k) \leq d(w_j, w_k)$.

*Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C. Email: mr884336@cs.nthu.edu.tw

The third property is called the *triangular inequality.* $\diamond$

Levenshtein distance has been thoroughly studied (see [1] for instance). It can be computed in polynomial time by a dynamic programming algorithm that computes this distance in $O(|u| * |v|)$ time, where $|u|$ and $|v|$ are lengths of these two strings u, v [6]. A finite set W of n strings $w_1, w_2, \ldots, w_n$ over an alphabet Σ, the median string problem is to find a string w such that the sum of its Levenshtein distance to each string of the set is minimized [4].

Definition 3 (median string problem)
Instance: A set W of n strings $w_1, w_2, \ldots, w_n$ over an alphabet Σ.
Goal: Find a string w in Σ^* such that $D(w, W)$ is minimized.

Definition 4 *Given a finite set of strings W over an alphabet Σ, a string m over Σ^* is a* median *of W if and only if for any given string w over Σ ,$D(m, W) \leq D(w, W)$.*

A finite set W of n strings $w_1, w_2, \ldots w_n$ over an alphabet Σ, the *centre string problem* is to find a string w that minimizes the maximum of Levenshtein distances to each string of the set [4].

Definition 5 (centre string problem)
Instance: A set W of n strings $w_1, w_2, \ldots, w_n$, over an alphabet Σ. Goal: Find a string w in Σ^* such that

$$\max_{w' \in W} d(w, w')$$

is minimized.

These two problems can be applied in computational molecular biology [5] and pattern recognition [2]. Unfortunately the two problems were shown to be NP-hard [4] by reduction from the longest common subsequence problem [3]. In this paper we shall show a $2 - \frac{2}{n}$ approximation algorithm on median string problem and a 2-approximation algorithm on centre string problem

For example, let $W = \{xxy, yy, xyxy\}$. Then $D(xy, W)$ =4 , since $d(xy, xxy) = 1, d(xy, yy) = 1$, and $d(xy, xyxy) = 2$, and

$$\max_{w' \in W} d(xy, w') = 2.$$

The string xyy is a median of W, since $D(xyy, W) = 3$ is minimal. The string yxy is a centre of W, since

$$\max_{w' \in W} d(yxy, w') = 1$$

is minimal. Notice that median may not be unique. For example, $xyxy$ is also an alternative median. Centre is also not unique, either, for xyy is an alternative centre.

2 An Approximation Algorithm for Media String Problem

Given a set $W = \{w_1, w_2, \ldots, w_n\}$. Our algorithm will select a string w_i from $\{w_1, w_2, \ldots, w_n\}$ such that $D(w_i, W)$ is minimal.

Algorithm: (median string problem)
Input: Given a set W of n strings $w_1, w_2, \ldots, w_n$.
Output: A string w_a.
Step 1: // compute the Levenshtein distances
    for i = 1 to n-1
        for j = i+1 to n
            Compute $d(w_i, w_j)$ by applying the dynamic programming algorithm in [6]
        next
    next
Step 2: // find w_a
    for i = 1 to n
        Compute $D(w_i W)$
    next

Find i such that $D(w_i, W)$ is minimal
for all $i \in \{1, 2, \ldots, n\}$
$w_a \leftarrow w_i$
Step 3: Return w_a and stop.

Example: $W = \{xxy, yy, xyxy\}$,

- $w_1 = xxy, D(w_1, W) = 0 + 2 + 1 = 3.$
- $w_2 = yy, D(w_2, W) = 2 + 0 + 2 = 4.$
- $w_3 = xyxy, D(w_3, W) = 1 + 2 + 0 = 3.$

Our algorithm may report w_1 as the approximate solution.

The time complexity of this algorithm is $O(n^2 + |u|^2)$, where $|u|$ is the maximum length of strings in W.

Theorem 1 *Let w_a denote the string chosen by our approximation algorithm, and w^* denote the optimal median string, then $\sum_{i=1}^{n} d(w_a, w_i) \leq 2\sum_{i=1}^{n} d(w^*, w_i)$.*

Proof: By our approximation algorithm,

$$\sum_{i=1}^{n} d(w_a, w_i) \leq \sum_{i=1}^{n} d(w_j, w_i) \text{ for all } j = 1, 2, \ldots, n.$$

Therefore,

$$\sum_{j=1}^{n}\sum_{i=1}^{n} d(w_a, w_i) \leq \sum_{j=1}^{n}\sum_{i=1}^{n} d(w_j, w_i). \tag{1}$$

Because the Levenshtein distance satisfies the triangular inequality, $d(w_j, w_i) \leq d(w_j, w^*) + d(w^*, w_i)$ for any string $w^* \in \sum^*$. Substitute this into Equation (1), we have

$$\sum_{j=1}^{n}\sum_{i=1}^{n} d(w_a, w_i) \leq \sum_{j=1}^{n}\sum_{i=1}^{n} (d(w_j, w^*) + d(w^*, w_i)) \tag{2}$$

$$\begin{aligned} n\sum_{i=1}^{n} d(w_a, w_i) &\leq \sum_{j=1}^{n}\sum_{i=1}^{n} d(w_j, w^*) + \sum_{j=1}^{n}\sum_{i=1}^{n} d(w^*, w_i) \\ &= n\sum_{j=1}^{n} d(w_j, w^*) + n\sum_{i=1}^{n} d(w^*, w_i) \\ &= n\sum_{i=1}^{n} d(w^*, w_i) + n\sum_{i=1}^{n} d(w^*, w_i) \\ &= 2n\sum_{i=1}^{n} d(w^*, w_i) \end{aligned} \tag{3}$$

Therefore,

$$\sum_{i=1}^{n} d(w_a, w_i) \leq 2\sum_{i=1}^{n} d(w^*, w_i).$$

Remarks. Since $d(w_i, w_i) = 0$, the right hand side of Equation (2) we have added extra $d(w_i, w^*) + d(w_i, w^*)$. This can be subtracted to get a more precise bound. Therefore the right hand side of Equation (2) will be

$$\sum_{j=1}^{n}\sum_{i=1}^{n} (d(w_j, w^*) + d(w^*, w_i)) - 2\sum_{i=1}^{n} d(w^*, w_i).$$

Thus Equation (3) will become

$$2n\sum_{i=1}^{n} d(w^*, w_i) - 2\sum_{i=1}^{n} d(w^*, w_i).$$

Therefore,

$$\sum_{i=1}^{n} d(w_a, w_i) \leq (2 - \frac{2}{n})\sum_{i=1}^{n} d(w^*, w_i).$$

Please note that the performance ratio $2 - \frac{2}{n}$ is a tight bound. For example, when $n = 3$ and $W = \{aab, bb, abba\}$. The approximation will choose aab as the approximate solution, which has $D(aab, W) = 4$. The optimal median string is $w^* = abb$ with $D(abb, W) = 3$. The performance ratio is $\frac{4}{3}$, which is exactly $2 - \frac{2}{n}$. $\diamond$

3 An Approximation Algorithm for the Centre String Problem

Theorem 2 *Randomly choosing a string from W, this string will be a 2-approximation solution*

Proof: Let w^* be the optimal center and $D = \max_{w \in W}(w^*, w)$ and w_c be the string chosen by our approximation algorithm. According to the triangular inequality,

$$\begin{aligned}\max_{w \in W} d(w_c, w) &\leq \max_{w \in W}(d(w_c, w^*) + d(w^*, w)) \\ &= d(w_c, w^*) + \max_{w \in W} d(w^*, w) \\ &\leq D + D \\ &= 2D.\end{aligned}$$

So the solution we obtained is an approximation solution that is less than or equal to twice of the optimal solution. $\diamond$

Remarks: It is easy to see that the ratio 2 is tight for our algorithm.

4 Conclusion and Future Direction

By defining the concept of *median* and *centre* on strings, we develop a $(2 - \frac{2}{n})$-approximation algorithm for the media string problem and a 2-approximation algorithm for the centre string problem.

Since the performance ratio of these approximation algorithms are not shown to be optimal, it is quite possible that these algorithms can be improved to get better performance ratios.

References

[1] A. V . Aho, "Algorithms for finding patterns in strings", *Handbook of Theoret. Comput. Sci.*, Elservier, Amsterdam, pp. 290–300, 1990.

[2] K.S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Englewood Cliffs,NJ, 1982.

[3] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman: San Francisco, CA, (1979) p.228.

[4] C. Higuera and F. Casacuberta, "Topology of strings: Median string is NP-complete", *Theoret. Comput. Sci.* **230** pp. 39–48, 2000.

[5] J. Setubal and J. Meidanis, *Introduction to computational molecular biology*, PWS Publishing Company, 1997.

[6] R .Wagner and M . Fisher, "The string-to-string correction problem", *Journal of ACM* **21** (1974), pp 168-178.