CS612 Algorithm Design and Analysis

Lecture 18. CLOSESTSTRING and CLOSESTSUBSTRING problems: random sampling and random rounding [1]

Presented by Mingfu Shao

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

[1] The slides are made based on *On the closest string and substring problems* by M. Li, B. Ma, and L. Wang.

Outline I

- Problem statements;
- ClosestString problem: random rounding technique;
- ClosestSubstring problem: random sampling technique;

Problem Statement I

Closest String

Given a set $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$ of strings each length m, find a center string s of length m minimizing d such that for every string $s_i \in \mathcal{S}$, $d(s, s_i) \leq d$. Here $d(s, s_i)$ is the Hamming distance between s and s_i.

Closest Substring

Given a set $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$ of strings each length m and an integer L, find a center string s of length L minimizing d such that for every string $s_i \in \mathcal{S}$ there is a length L substring t_i of s_i with $d(s, t_i) \leq d$.

Example of CLOSEST STRING

Given 4 strings s_1, s_2, s_3, s_4,

0	0	1	0	0	0	s1
1	1	1	0	0	0	s2
0	1	1	0	1	1	s3
0	1	0	1	0	1	s4
0	1	1	0	0	1	s

Optimal center string $s = 011000$

$$d = \max_{i=1}^{4} d(s_i, s) = \max\{2, 2, 1, 2\} = 2$$

Example of CLOSEST SUBSTRING

Given 4 strings s_1, s_2, s_3, s_4, $L = 4$,

0	0	1	0	0	0		s1
1	1	1	0	0	0		s2
0	1	1	0	1	1		s3
0	1	0	1	0	1		s4
	1	1	0	0			s

Optimal center string $s = 1100$,

$$d = \max_{i=1}^{4} d(t_i, s) = \max\{1, 0, 1, 2\} = 2$$

Basic Idea

- Applying "LP+RR" technique on all columns doesn't work due to the large size of the variables.
- In polynomial time, we can enumerate all $\binom{n}{r}$ strings for any fixed r. We can prove that, on positions that r strings all agree, denote as Q, it is a good approximate solution; more speficially, we can also prove that, $|Q| \geq m - r \cdot d_{opt}$.
- On the other positions, we use LP + Random rounding technique to obtain an approximate solution.

Algorithm I

Input : $s_1, s_2, \cdots, s_n \in \Sigma^m$, an integer $r \geq 2$ and a small number $\epsilon > 0$.
Output : a center string $s \in \Sigma^m$
Algorithm :

1. **for** each r-element subset $\{s_{i_1}, s_{i_2}, \cdots, s_{i_r}\}$ of the n input strings **do**
    1. let $Q = \{j | s_{i_1}[j] = s_{i_2}[j] = \cdots = s_{i_r}[j]\}$, $P = \{1, 2, \cdots, m\} - Q$
    2. solve the following optimization problem

    $$\begin{aligned} & \min d \\ s.t. \quad & d(s_i|_P, y) + d(s_i|_Q, s_{i_1}|_Q) \leq d, i = 1, 2, \cdots, n \end{aligned}$$

    Random rounding the fractional solution $\bar{y}$ to get a approximation solution y. Use derandomization technique to make this step deterministic rather than random.
    3. let $s'|_Q = s_{i_1}|_Q, s'|_P = y$. Calculate the radius of the solution with s' as the center string.
2. **for** $i = 1, 2, \cdots, n$ **do**

Algorithm II

1. calculate the radius of the solution with s_i as the center string.

3. output the best solution of the above two steps.

Example of CLOSEST STRING I

Suppose $r = 2$. In step 1, we should enumerate all $\binom{4}{2} = 6$ cases. In each cases, we select 2 lines, calculate P and Q.

r=2

0	0	1	0	0	0	s1
1	1	1	0	0	0	s2
0	1	1	0	1	1	s3
0	1	0	1	0	1	s4
P		Q				
y		1	0	0	0	s'

Example of Closest String II

In step 2, we fix $s'|_Q = 1000$, solve the following optimization problem :

$$
\begin{aligned}
& \min d \\
s.t. \quad & y_{10} + y_{11} = 1 \\
& y_{20} + y_{21} = 1 \\
& y_{11} + y_{21} + 0 \leq d \\
& y_{10} + y_{20} + 0 \leq d \\
& y_{11} + y_{20} + 2 \leq d \\
& y_{11} + y_{20} + 2 \leq d
\end{aligned}
$$

Solve this linear programming, and random rounding the fractional solution to integer solution :

$$y_{10} = y_{21} = 1, y_{11} = y_{20} = 0$$

Example of Closest String III

So,

$$s'|_P = 01, s' = 011000$$

$$d = \max_{i=1}^{4} d(s_i, s') = \max\{1, 1, 2, 3\} = 3$$

Try all $\binom{4}{2} = 6$ cases, obtain the minimum, denote as d_0. Then we finish step 1.
In step 2, we calculate the radius when s_i is the center string.

$$\begin{aligned}
d_1 &= \max_{i=1}^{4} d(s_1, s_i) \\
d_2 &= \max_{i=1}^{4} d(s_2, s_i) \\
d_3 &= \max_{i=1}^{4} d(s_3, s_i) \\
d_4 &= \max_{i=1}^{4} d(s_4, s_i)
\end{aligned}$$

Example of CLOSEST STRING IV

Calculate the minimal radius in both step 1 and step 2.

$$d = \min\{d_0, d_1, d_2, d_3, d_4\}$$

Analysis I

Theorem

The above algorithm is a PTAS for CLOSEST STRING problem.

Proof.

Obviously, the time complexity of the algorithm is $O\left((nm)^r n^{O(\log|\Sigma|\cdot r^2/\epsilon^2)}\right)$, which is polynomial in terms of n,m.
The proof of approximation guarantee is organized as 3 lemmas as follows :
Lemma 1 proves $s'|_Q$ is a good approximation to s with approximation rate $1+\frac{1}{2r-1}$.
Lemma 2 proves $|P| < r \cdot d_{opt}$.
Based on the above 2 lemmas, Lemma 3 proves step 1.2 obtains a approximate solution with rate $(1+\frac{1}{2r-1}+\epsilon)$. □

Analysis II

Lemma 1

If $\max_{i \le i,j \le n} d(s_i, s_j) > (1 + \frac{1}{2r-1})d_{opt}$, then there **exists** r indices $1 \le i_1, i_2, \cdots, i_r \le n$ such that for any $1 \le l \le n$,

$$d(s_l|_Q, s_{i_1}|_Q) - d(s_l|_Q, s|_Q) \le \frac{1}{2r-1} d_{opt}$$

where Q is the set of positions that $s_{i_1}, s_{i_2}, \cdots, s_{i_r}$ all agree.

- if $\max_{i \le i,j \le n} d(s_i, s_j) \le (1 + \frac{1}{2r-1})d_{opt}$, then any s_i will be a good center string. (Recall the step 2 of the algorithm)

Analysis III

Lemma 2

Let $P = \{1, 2, \cdots, m\} - Q$, then $|P| \leq r \cdot d_{opt}$ and $|Q| \geq m - r \cdot d_{opt}$.

Proof.

Let $q \in P$, then there exists some s_{i_j} such that $s_{i_j}[q] \neq s[q]$. Since $d(s_{i_j}, s) \leq d_{opt}$, each s_{i_j} contributes at most d_{opt} positions for P. Thus $|P| \leq r \cdot d_{opt}$. □

- this lemma gives the lower bound of d_{opt}, $d_{opt} \geq \frac{|P|}{r}$, which is essential for the analysis of LP + random rounding!

Analysis IV

Lemma 3

Given a string s' and a position set Q and P, $|P| < r \cdot d_{opt}$, such that for any $i = 1, 2, \cdots, n$,

$$d(s_i|_Q, s'|_Q) - d(s_i|_Q, s|_Q) \leq \rho \cdot d_{opt},$$

then step 1.2 gives a solution with approximate rate $(1 + \rho + \epsilon)$ in polynomial time for any fixed $\epsilon \geq 0$.

Before the proof, we can see that the conditions of this lemma are all satisfied by lemma 2, where $\rho = \frac{1}{2r-1}$.

Proof

Recall the optimization problem in step 1.2

$$\begin{aligned} & \min d \\ s.t. \quad & d(s_i|_P, y) + d(s_i|_Q, s'|_Q) \leq d, i = 1, 2, \cdots, n \end{aligned}$$

Analysis V

First, we show that $y = s|_P$ is a solution with cost $d \leq (1+\rho)d_{opt}$. In fact, for $i = 1, 2, \cdots, n$

$$\begin{aligned} & d(s_i|_P, s|_P) + d(s_i|_Q, s'|_Q) \\ \leq\ & d(s_i|_P, s|_P) + d(s_i|_Q, s|_Q) + \rho \cdot d_{opt} \\ \leq\ & (1+\rho)d_{opt} \end{aligned}$$

Second, rewirte the optimization problem to an ILP problem. In order for this, define 0-1 variables $y_{j,a}$, $1 \leq j \leq |P|$, $a \in \Sigma$, $y_{j,a} = 1$ means $y[j] = a$; define $\chi(s_i[j], a) = 0$ if $s_i[j] = a$ and 1 if $s_i[j] \neq a$. Then the above optimization problem can be formulated as follows

$$\min d$$

$$s.t. \quad \begin{cases} \sum_{a\in\Sigma} y_{j,a} = 1, 1 \leq j \leq |P| \\ \sum_{1\leq j\leq |P|} \sum_{a\in\Sigma} \chi(s_i[j], a) y_{j,a} + d(s_i|_Q, s'|_Q) \leq d, 1 \leq i \leq n \end{cases}$$

Solve it by LP to get a fractional solution $\bar{y}$ with cost $\bar{d}$.

Analysis VI

Random rounding $\bar{y}$ to y' by independently set $y'_{j,a} = 1$ and $y'_{j,b} = 0, b \in \Sigma - \{a\}$ with probability $\bar{y_{j,a}}$.
So $d(s_i|_P, y') = \sum_{i=1}^{|P|} \sum_{a \in \Sigma} \chi(s_i[j], a) y_{j,a}$, which is the sum of $|P|$ independent random variables.

$$\begin{aligned} E(d(s_i|_P)) &= \sum_{1 \le j \le |P|} \sum_{a \in \Sigma} \chi(s_i[j], a) \bar{y_{j,a}} \\ &\le \bar{d} - d(s_i|_Q, s'|_Q) \\ &\le (1+\rho) \cdot d_{opt} - d(s_i|_Q, s'|_Q) \end{aligned}$$

Employ Chernoff Bound

$$\Pr(X > \mu + \epsilon n) \le \exp(-\frac{1}{3} n \epsilon^2)$$

Analysis VII

we have

$$\Pr(d(s_i|_P, y') > (1+\rho)d_{opt} - d(s_i|_Q, s'|_Q) + \epsilon'|P|) \leq \exp(-\frac{1}{3}\epsilon'^2|P|)$$

Let $s'|_P = y'$ and consider all n strings, we claim

$$\Pr(d(s_i, s') < (1+\rho)d_{opt} + \epsilon'|P| \text{ for all i}) \geq 1 - n\exp(-\frac{1}{3}\epsilon'^2|P|)$$

Use standard derandomization methods, we can obtain a deterministic s' that satisfies

$$d(s_i, s') < (1+\rho)d_{opt} + \epsilon'|P|, \quad 1 \leq i \leq n$$

Recall that $|P| < r \cdot d_{opt}$, let $\epsilon' = \frac{\epsilon}{r}$, we have

$$d(s_i, s') < (1+\rho+\epsilon)d_{opt}, \quad 1 \leq i \leq n$$

Then we finish the proof.

Basic Idea

- We want to follow the algorithm of CLOSEST STRING algorithm. However, there is an obstacle: we do not know how to construct an optimization problem, for the reason that we do not konw the optimal substring in each string. Thus, the choice of a "good" substring from every string s_i is the only obstacle on the way to the solution.
- There is a "dead cycle":
  1. Suppose we know the center substring s^*, the optimal t_i can be easily determined through comparison within a sliding window.
  2. On the other hand, suppose we know all the optimal substrings t_i, s^* can also be approximated using the CLOSESTSTRING algo.
- How to break this cycle? Random sampling!
- Speficically, we first randomly sample $O(logmn)$ positions R. By trying all length $|R|$ strings, we can assume we know $s|_R$. The performance of these positions can be used to estimate the performance on all positions, i.e., for each $i \leq i \leq n$, we find the substring t'_i from s_i such that $f(t'_i) = d(s|_R, t'_i|_R) \cdot \frac{|P|}{|R|} + d(t_{i_1}|_Q, t'_i|_Q)$ is minimized. We use $t'_i, 1 \leq i \leq n$ to construct the optimization problem.

Algorithm for CLOSEST SUBSTRING problem I

Input : $s_1, s_2, \cdots, s_n \in \Sigma^m$, an integer $1 \leq L \leq m$, an integer $r \geq 2$ and a small number $\epsilon > 0$.

Ouput : center string s

Algorithm :

1. **for** each r-element subset $\{t_{i_1}, t_{i_2}, \cdots, t_{i_r}\}$, where t_{i_j} is a substring of length L from s_{i_j} **do**
    1. let $Q = \{j | t_{i_1}[j] = t_{i_2}[j] = \cdots = t_{i_r}[j]\}$, $P = \{1, 2, \cdots, m\} - Q$
    2. randomly select $\frac{4}{\epsilon^2} \log(mn)$ positions from P, denote as R
    3. **for** every string x of length $|R|$ **do**
        1. for $1 \leq i \leq n$, let t'_i be a length L substring of s_i minimizing $f(t'_i) = d(x, t'_i|_R) \cdot \frac{|P|}{|R|} + d(t_{i_1}|_Q, t'_i|_Q)$.
        2. solve the optimization problem

$$\begin{aligned} & \min d \\ s.t. \quad & d(t'_i|_P, y) + d(t'_i|_Q, t_{i-1}|_Q) \leq d, 1 \leq i \leq n \end{aligned}$$

Random rounding the fractional solution $\bar{y}$ to get a approximation solution y. Use derandomization technique to make this step deterministic rather than random.

Algorithm for CLOSEST SUBSTRING problem II

   3. Let $s'|_Q = t_{i_1}|_Q$ and $s'|_P = y$. Let $c = \max_{i=1}^{n} d(s', t'_i)$.
2. **for** every length L substring s' of s_1 **do**
   1. Let $c = \max_{i=1}^{n} \min_{t_i} d(s', ti)$.
3. output the s' with minimum c in step 1.3.3 and step 2.1.

Example of Closest Subtring I

Suppose $r = 2$. In step 1, we should enumerate all $\binom{4}{2} \times 3 \times 3 = 54$ cases. In each cases, we select 2 substring of length 4 , calculate P and Q.
We fix $s'|_Q = 00$.
Randomly select $|R| = O(\log(mn))$ positions, say, $|R| = 1$. Then enumerate all length $|R|$ strings, say, $s'|_R = 0$.

Example of CLOSEST SUBTRING II

r=2

0	0	1	0	0	0	s1
1	1	1	0	0	0	s2
0	1	1	0	1	1	s3
0	1	0	1	0	1	s4

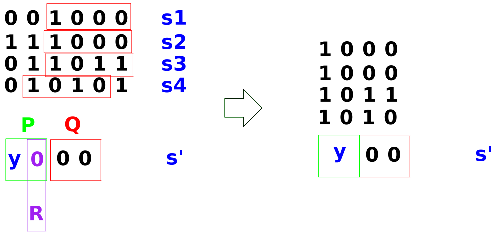
P Q

0	1	0	0	t1
1	0	0	0	t2
y	0	0	0	s'

R

Example of CLOSEST SUBTRING III

Now, for each s_i, find out the t_i' minimizing $f(u) = d(u|_R, s'|_R) \times 2 + d(u|_Q, s'|_Q)$.



Example of Closest Subtring IV

Solve the following optimization problem :

$$\begin{aligned} & \min d \\ s.t. \quad & y_{10} + y_{11} = 1 \\ & y_{20} + y_{21} = 1 \\ & y_{10} + y_{21} + 0 \le d \\ & y_{10} + y_{21} + 0 \le d \\ & y_{10} + y_{21} + 2 \le d \\ & y_{10} + y_{21} + 1 \le d \end{aligned}$$

Solve this linear programming, and random rounding the fractional solution to integer solution :

$$y_{10} = y_{21} = 0, y_{11} = y_{20} = 1$$

Example of CLOSEST SUBTRING V

So,

$$s' = 1000$$

$$d = \max_{i=1}^{4} d(t_i, s') = \max\{0, 0, 2, 1\} = 2$$

Try all 54 cases, obtain the minimum, denote as d_0. Then we finish step 1.

In step 2, we calculate the radius when t_i is the center string where t_i is a length L substring of s_i. denote the minimun d_1. Calculate the minimal radius in both step 1 and step 2.

$$d = \min\{d_0, d_1\}$$

Analysis I

Theorem

The above algorithm is a PTAS for CLOSEST SUBSTRING problem.

Proof.

The time complexity of the algorithm is $O\left((n^m)^{O(\log|\Sigma|/\delta^4)}\right)$, which is polynomial in terms of n,m.
The proof of approximation guarantee is organized as 3 lemmas as follows :
Lemma 1 proves $s'|_Q$ is a good approximation to s with approximation rate $1+\frac{1}{2r-1}$.
Define $s^*|_P = s|_P$ and $s^*|_Q = t_{i_1}|_Q$, lemma 2 proves $d(s^*, t_i') \leq d(s^*, t_i) + 2\epsilon|P|$ for all $1 \leq i \leq n$.
Based on the above 2 lemmas, Lemma 3 proves the algorithm obtains a approximate solution with rate $(1+\frac{1}{2r-1}+3\epsilon r)$. □

Analysis II

Lemma 1

There exists $t_{i_1}, t_{i_2}, \cdots, t_{i_r}$ chosen in step 1, such that for any $1 \leq l \leq n$

$$d(t_l|_Q, t_{i_1}|_Q) - d(s_l|_Q, s|_Q) \leq \frac{1}{2r-1} \cdot d_{opt}$$

Proof.

The fact follow from Lemma 1 of CLOSEST STRING directly. □

Lemma 2

Define $s^*|_P = s|_P$ and $s^*|_Q = t_{i_1}|_Q$. Then we have, with high probability

$$d(s^*, t_i') \leq d(s^*, t_i) + 2\epsilon|P|$$

for all $1 \leq i \leq n$.

Analysis III

Proof.

□

- The randomness comes from the randomly selected $|R|$. Use standard method, we can derandomize it to make this deterministic.

Lemma 3

step 1.3.3 gives an approximation solution s' with approximation rate $(1 + \frac{1}{2r-1} + 3\epsilon r)$.

Proof.

Recall the optimization problem in step 1.2

$$\begin{aligned} & \min d \\ s.t. \quad & d(t_i'|_P, y) + d(t_i'|_Q, s'|_Q) \le d, i = 1, 2, \cdots, n \end{aligned}$$

Analysis IV

$y = s|_P$ is a feasible solution. Now we calculate its radius when s^* is the center string. Recall that $s^*|_P = s|_P$ and $s^*|_Q = t_{i_1}|_Q$.
According to Lemma 2 and Lemma 1

$$\begin{aligned} d(s^*, t_i') &\leq d(s^*, t_i) + 2\epsilon|P| \\ &\leq d(s|_P, t_i|_P) + d(t_{i_1}|_Q, t_i|_Q) + 2\epsilon|P| \\ &\leq d(s|_P, t_i|_P) + d(s|_Q, t_i|Q) + 2\epsilon|P| + \frac{1}{2r-1}d_{opt} \\ &\leq (1 + \frac{1}{2r-1})d_{opt} + 2\epsilon|P|) \end{aligned}$$

In a short word, $y = s|_P$ is a solution of the optimization problem with cost at most $(1 + \frac{1}{2r-1})d_{opt} + 2\epsilon|P|)$.
Next, as we do in the CLOSEST STRING problem, we rewirte the optimization problem to an ILP and solve it and random rounding the fractional solution $\bar{y}$ to y. Define $s'|_Q = t_{i_1}|_Q$ and $s'|_P = y$.

Analysis V

$$E(d(s',t_i')) \leq \bar{d} \leq (1 + \frac{1}{2r-1} + 2\epsilon|P|)d_{opt}$$

So, chernoff bound ensures that, with high probability,

$$\begin{aligned} & d(s',t_i') \\ \leq \quad & (1 + \frac{1}{2r-1})d_{opt} + 2\epsilon|P| + \epsilon|P| \\ \leq \quad & (1 + \frac{1}{2r-1})d_{opt} + 3\epsilon|P| \\ \leq \quad & (1 + \frac{1}{2r-1} + 3\epsilon r)d_{opt} \end{aligned}$$

After derandomization, we can obtain an approximation solution s' with rate $(1 + \frac{1}{2r-1} + 3\epsilon r)d_{opt}$.