Sublinear algorithms

neděle 16. listopadu 2025 22:03

Q: Are there some simple algorithms for edit distance?

Algorithm (Saha '14): on input x & y: $O(n)$-time

$i = j = 1$; $k = 0$;

while $i \le n$ & $j \le n$ do

if $x_i \neq y_j$ then flip a coin & k++ $\begin{cases} i\text{++} & \text{w. prob } \frac{1}{2} \\ j\text{++} & \text{w. prob } \frac{1}{2} \end{cases}$

else i++, j++

output $k + (n-i) + (n-j)$;

($n-i$: remainder of x; $n-j$: remainder of y)

Claim: $ED(x,y) \le c \le O(ED(x,y))^2$ w. prob $\ge 2/3$.

- We will prove this later.

Q: Can we derandomize the algorithm?

Algorithm (K.-Saha '23): x & $y \in \Sigma^n$, $\sigma = 011100000\cdots0$ (positions $4i-3$ to $4i-1$ are 1)

$i = j = 1$; $k = 0$;

while $i \le n$ & $j \le n$ do

if $x_i \neq x_j$ then k++; if $\sigma_k = 0$ then i++; else j++;

else i++ j++;

output $k + (n-i) + (n-j)$

max +2 +2 +2

$0\,111\,00000\,111\cdots1$ $000\cdots0\,111\cdots$

−2 −2 −2

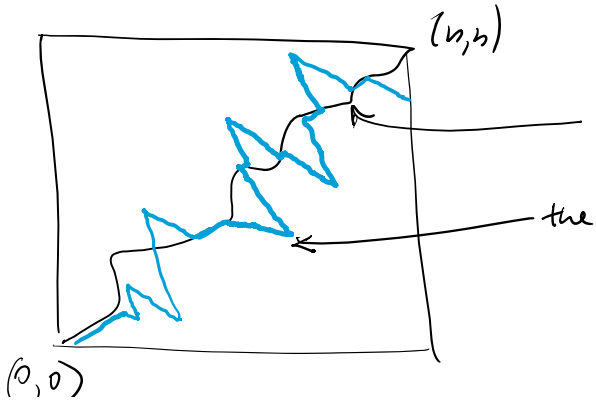
min

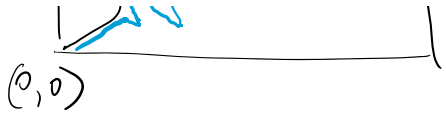
block k: $\sum_{i=1}^{k} \sigma_i \in [-2k, 2k]$

each block +2 / −2.

Claim: $ED(x,y) \le k \le \frac{9}{4} \cdot ED(x,y)^2$

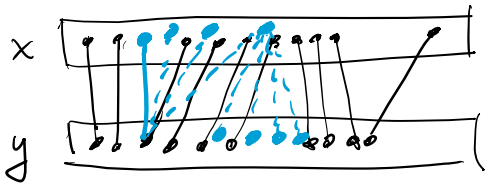
Pf: $G_{x,y}$





- the algorithm is scanning for the optimal path, after $\frac{k}{2}$ rounds (blocks of 0's & 1's in σ) $i-j$ ranges over diagonals $[-\frac{k}{2}, \frac{k}{2}]$

$k = ED(x,y)$



$i-j$ = slope of the edge ... diagonal

the optimal path is restricted to diagonals $[-\frac{k}{2}, \frac{k}{2}]$

On every mismatch our algorithm encounters we change a diagonal. Since we scan over consecutive diagonals we have to hit the optimal path. Then we follow the optimal path until the nearest mismatch on that path. Then we might departure from it, but within next block of σ we encounter it again.

$\Rightarrow$ we get $\sum_{i=1}^{k/2} 2i-1$ mismatches during the first $\frac{k}{2}$ blocks of σ & then we get $\approx i$ mismatches, $i = \frac{k}{2}+1, \dots, \frac{3}{2}k$, during the next k rounds (blocks of σ).

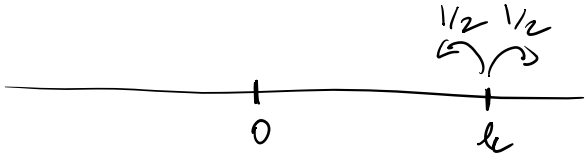
$\Rightarrow$ we encounter $\sum_{i=1}^{\frac{3}{2}k} 2i-1 \le 2\sum_{i=1}^{\frac{3}{2}k} i \le 2\cdot\frac{\left(\frac{3}{2}k\right)^2}{2} = \frac{9}{4}k^2$ mismatches.

The algorithm produces a matching $\Rightarrow ED(x,y) \le k$.

$\rightarrow$ the randomized algorithm makes a random decision on each mismatch whether to increase / decrease its diagonal.

$\approx$ random walk on a line

≈ random walk on a line



which wants to hit 0 (the optimal diagonal).

Algorithm (Saha - Kociumaka '20): $x \& y \in \Sigma^*$, $p \in [n]$

$i = j = 1$; $k = 0$;

while $i \le n$ & $j \le n$ do

Sample $s \in_R \{0,1\}$ s.t. $s = \begin{cases} 1 & \text{w.p. } \frac{2\log n}{p} \\ 0 & \text{w.p } 1 - \frac{2\log n}{p} \end{cases}$

if $s = 1$ & $x_i \neq y_j$ then

with prob $1/2$ { i++ / j++

k++

else i++, j++;

output k

Claim: With probability $\ge 2/3$ $\quad \frac{1}{p} \cdot ED(x,y) \le k \le O(ED(x,y))^2$

$\rightarrow k \cdot p$ satisfies $ED(x,y) \le k \cdot p \le p \cdot O(ED(x,y))^2$ w.h.p

The algorithm can be implemented so that it directly selects the next position when $s = 1$ & increases i & j simultaneously by that amount.

$\tilde{O}(n/p)$-time pk^2-approx

(selecting the length of the "jump" uniformly at random from $[0, \frac{p}{2\log n}]$ would work as well.)

→ the algorithm is constructing some alignment, matching x_i to y_j

Claim: The probability that the algorithm misses the next p mismatches (to observe $s = 1$)

is $\le \left(1 - \frac{2\log n}{p}\right)^p \le n^{-2}$.

at most $(1 - \frac{2\log n}{p})$

is $\leq \left(1 - \frac{2\log n}{p}\right)^p \leq n^{-2}$.

Pf: each mismatch has probability at most $\left(1 - \frac{2\log n}{p}\right)$ of being missed ($s=0$).

$\Rightarrow$ missing p mismatches has prob. $\left(1 - \frac{2\log n}{p}\right)^p \leq \frac{1}{n^2}$ ∎

$\Rightarrow$ Except w. prob. $\frac{k}{n^2} \leq \frac{n}{n^2} = \frac{1}{n}$, there are $<p$ unobserved mismatches per every observed mismatch.

$\Rightarrow$ $ED(x,y) \leq k \cdot p$ w.p. $\geq 1 - \frac{1}{n}$.

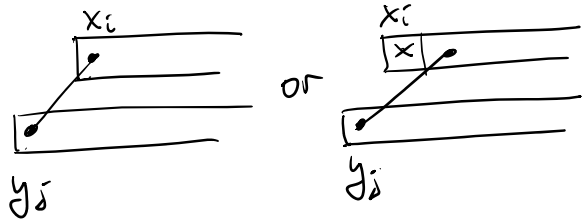
Remains to prove that $k \leq O(ED(x,y))^2$ w.p. $\geq 2/3$

define $D_t = INSIDE(x_{i\ldots n}, y_{j\ldots n})$ ... at iteration t of the alg.

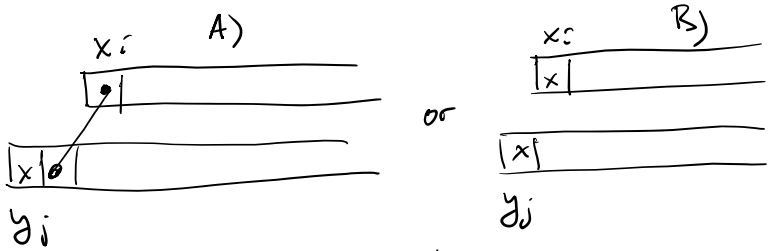
$\forall t$ $D_{t+1} \in D_t + \{-1, 0, 1\}$

observe: 1) if $s=0$ then $D_{t+1} = D_t$ (both i & j are incremented)

consider opt. match of $x_{i\ldots n}$ & $y_{j\ldots n}$:



2) if $s=1$ & $x_i \neq y_j$



$\forall$ opt alignment x_i is matched (symmetrically for y_j)

w.p. $\frac{1}{2}$ $\begin{cases} D_{t+1} = D_t - 1 & (\text{del } y_j) \\ D_{t+1} = D_t + 1 & (\text{del } x_i) \end{cases}$ $\quad D_{t+1} = D_t - 1$

3) if $s=1$ & $x_i = y_j$ then $D_{t+1} = D_t$.

3) if $s=1$ & $x_i = y_j$ then $D_{t+1} = D_t$.

4) if $D_t = 0$ then $D_{t+1} = 0$

define a random walk W_t: in case 1) & 3) $W_{t+1} = W_t$

in case 2) if A) occurs $W_{t+1} = W_t + (D_{t+1} - D_t)$

if B) occurs $W_{t+1} = W_t + \begin{cases} +1 & w.p. \frac{1}{2} \\ -1 & w.p. \frac{1}{2} \end{cases}$

$W_0 = 2\,ED(x,y)$

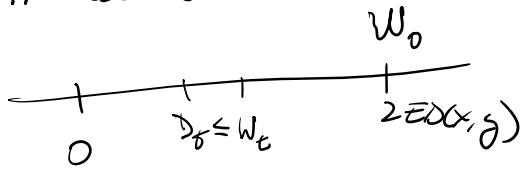
- case 2) occurs k times
- $W_t \geq D_t$ (in B), D_t decreases but W_t increases w.p. $\frac{1}{2}$)

& initially $W_0 = 2 \cdot ED(x,y) \geq D_0$

$\Rightarrow$ if $W_t = 0 \Rightarrow D_t = 0$ & W_t stays zero. (no 2) occurs after W_t reaches 0)

- W_t has a distribution of a random walk on a line starting at $2 \cdot ED(x,y)$ & stops when it reaches 0.

↑ unbiased ($\frac{1}{2}$ / $\frac{1}{2}$)



Fact: $Pr[W_0 \ldots W_t \text{ does not reach } 0] \leq \frac{6 \cdot W_0}{\sqrt{t}} = \frac{12 \cdot ED(x,y)}{\sqrt{t}}$

$Pr[k > (36 \cdot ED(x,y))^2] \leq Pr[W_0 \ldots W_{(36 \cdot ED(x,y))^2} \text{ doesn't reach zero}]$

$D_0 \ldots D_{(36 \cdot ED(x,y))^2} \neq 0$ (equiv.)

$\leq \frac{12 \cdot ED(x,y)}{36 \cdot ED(x,y)} \leq \frac{1}{3}$

$\Rightarrow Pr[k \leq 36^2 \cdot ED(x,y)^2] \geq \frac{2}{3}$ ∎