Approximating Text-to-Pattern Hamming Distances

Yonggang Jiang

July 13, 2022

Problem

Inputs:

- Pattern string P and text string T with $|P| = m, |T| = n$.

Outputs:

- An $1 + \epsilon$ approximation of hamming distance between P and **any** size m substring of T.

Problem

Inputs:

- Pattern string P and text string T with $|P| = m, |T| = n$.

Outputs:

- An $1 + \epsilon$ approximation of hamming distance between P and **any** size m substring of T.

Assumption:

- $n = \Theta(m)$.
- Assume ϵ is a small constant.

Problem

Inputs:

- Pattern string P and text string T with $|P| = m, |T| = n$.

Outputs:

- An $1 + \epsilon$ approximation of hamming distance between P and **any** size m substring of T.

Assumption:

- $n = \Theta(m)$.
- Assume ϵ is a small constant.

Goal: Running time $O(n \log^{1.5} n)$.

Approximate Hamming Distance $HD(X, Y)$

Assumption: We already know a rough approximation $k = \Theta(HD(X, Y))$. (can be removed by guessing $k = 2^i$ for $i = 1, 2, ..., \log n$)

Approximate Hamming Distance $HD(X, Y)$

Assumption: We already know a rough approximation $k = \Theta(HD(X, Y))$. (can be removed by guessing $k = 2^i$ for $i = 1, 2, ..., \log n$)

Recall $\tilde{O}(m/k)$ algorithm: Sample $i \in [m]$ for $\Theta\left(\frac{m}{k}\right)$ times, and count how many times we have $X[i] \neq Y[i]$.

Approximate Hamming Distance $HD(X, Y)$

Assumption: We already know a rough approximation $k = \Theta(HD(X, Y))$. (can be removed by guessing $k = 2^i$ for $i = 1, 2, ..., \log n$)

Recall $\tilde{O}(m/k)$ algorithm: Sample $i \in [m]$ for $\Theta\left(\frac{m}{k}\right)$ times, and count how many times we have $X[i] \neq Y[i]$.

Too slow since we have $\Theta(n)$ instances! Idea: sample many positions at one time.

New algorithm for Approx $HD(X, Y)$

1. Let c be a sufficiently large constant. Let p be a uniform random prime in $[ck \log m, 2ck \log m]$.

New algorithm for Approx $HD(X, Y)$

1. Let c be a sufficiently large constant. Let p be a uniform random prime in $[ck \log m, 2ck \log m]$.
2. Independently uniformly sample $B_1, B_2, ..., B_c \subseteq [p]$ with $|B_i| = \log m$. Compute

$$x_i := h(\bigodot_{j \mod p \in B_i} X[j])$$

$$y_i = h(\bigodot_{j \mod p \in B_i} Y[j])$$

for each i. (Here $h(S)$ is the polynomial rolling hashing $\sum S[i] \cdot rand^i \mod m^c$.)

New algorithm for Approx $HD(X, Y)$

1. Let c be a sufficiently large constant. Let p be a uniform random prime in $[ck \log m, 2ck \log m]$.
2. Independently uniformly sample $B_1, B_2, ..., B_c \subseteq [p]$ with $|B_i| = \log m$. Compute

$$x_i := h(\bigodot_{j \mod p \in B_i} X[j])$$

$$y_i = h(\bigodot_{j \mod p \in B_i} Y[j])$$

   for each i. (Here $h(S)$ is the polynomial rolling hashing $\sum S[i] \cdot rand^i \mod m^c$.)
3. Let d as the number of i that $x_i \neq y_i$. Return

$$f(d) = p \cdot \left(1 - \left(1 - \frac{d}{c}\right)^{\frac{1}{\log m}}\right)$$

Correctness

What to approximate

$$M := \{i \in [m] \mid X[i] \neq Y[i]\}$$

Instead we pick an appropriate parameter p and approximate

$$M' := \{b \in [p] \mid \bigodot_{i \mod p = b} X[i] \neq \bigodot_{i \mod p = b} Y[i]\}$$

Correctness

What to approximate

$$M := \{i \in [m] \mid X[i] \neq Y[i]\}$$

Instead we pick an appropriate parameter p and approximate

$$M' := \{b \in [p] \mid \bigodot_{i \mod p = b} X[i] \neq \bigodot_{i \mod p = b} Y[i]\}$$

The following happens with large constant probability.

- With random prime p in $[ck \log m, 2ck \log m]$, we have $|M'| \approx |M|$.
- $f(d) \approx |M'|$.

$|M| \approx |M'|$

$$M' := \{b \in [p] \mid \bigodot_{i \bmod p = b} X[i] \neq \bigodot_{i \bmod p = b} Y[i]\}$$
$$= \{i \bmod p \mid i \in M\}$$

p is a random prime from $[\hat{p}, 2\hat{p}]$ where $\hat{p} = ck \log m$.
Trivial fact: $|M'| \leq |M|$.

$|M| \approx |M'|$

$$M' := \{b \in [p] \mid \bigodot_{i \bmod p = b} X[i] \neq \bigodot_{i \bmod p = b} Y[i]\}$$
$$= \{i \bmod p \mid i \in M\}$$

p is a random prime from $[\hat{p}, 2\hat{p}]$ where $\hat{p} = ck \log m$.
Trivial fact: $|M'| \leq |M|$.
Want to prove $|M| - |M'| < \epsilon |M'|$ (ϵ is a small constant).

$|M| \approx |M'|$

$$M' := \{b \in [p] \mid \bigodot_{i \mod p = b} X[i] \neq \bigodot_{i \mod p = b} Y[i]\}$$
$$= \{i \mod p \mid i \in M\}$$

p is a random prime from $[\hat{p}, 2\hat{p}]$ where $\hat{p} = ck \log m$.
Trivial fact: $|M'| \leq |M|$.
Want to prove $|M| - |M'| < \epsilon |M'|$ (ϵ is a small constant).

Proof.

$$|M| - |M'| \leq \#\{i, j \in M \mid |i - j| \mod p = 0\}$$

$$\mathbb{E}\left[|M'| - |M|\right] \leq |M|^2 \cdot \frac{\log_{\hat{p}} m}{\hat{p} / \log \hat{p}} = O\left(|M|/c\right)$$

Markov's inequality: $\Pr[|M'| - |M| \geq \epsilon |M|] \leq O(1/\epsilon c)$

$\square$

$f(d) \approx |M'|$

$$E[d] = c \cdot \left(1 - \left(1 - \frac{|M'|}{p}\right)^{\log m}\right) = f^{-1}(|M'|)$$

$f(d) \approx |M'|$

$$E[d] = c \cdot \left(1 - \left(1 - \frac{|M'|}{p}\right)^{\log m}\right) = f^{-1}(|M'|)$$

- Use Chernoff bound to get $d \approx E[d] = f^{-1}(|M'|)$.

$f(d) \approx |M'|$

$$E[d] = c \cdot \left(1 - \left(1 - \frac{|M'|}{p}\right)^{\log m}\right) = f^{-1}(|M'|)$$

- Use Chernoff bound to get $d \approx E[d] = f^{-1}(|M'|)$.
- Use the following lemma

Lemma

If the absolute derivative of $\log f(2^x)$ *is bounded by* c, *and* $(1-\epsilon)y < x < (1+\epsilon)y$, *then* $(1-2c\epsilon')f(y) < f(x) < (1+c\epsilon')f(y)$.

$f(d) \approx |M'|$

$$E[d] = c \cdot \left(1 - \left(1 - \frac{|M'|}{p}\right)^{\log m}\right) = f^{-1}(|M'|)$$

- Use Chernoff bound to get $d \approx E[d] = f^{-1}(|M'|)$.
- Use the following lemma

Lemma

If the absolute derivative of $\log f(2^x)$ is bounded by c, and $(1-\epsilon)y < x < (1+\epsilon)y$, then $(1-2c\epsilon')f(y) < f(x) < (1+c\epsilon')f(y)$.

Further facts: If k is too large, $E[d]/c$ will be too close to 0; if k is too small, $E[d]/c$ will be too close to 1: we can distinguish the cases where $k >> |M|$ or $k << |M|$.

text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \mod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \mod p \in B, j \in [m]} T[i+j]\right)$$

For any i.

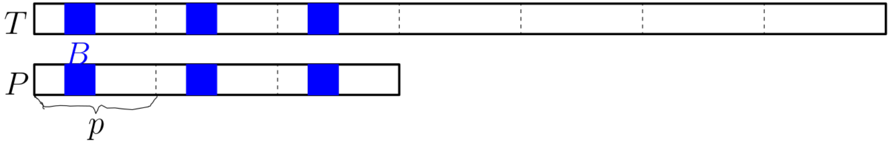
text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \bmod p \in B, j\in[m]} P[j]\right)$$

$$h\left(\bigodot_{j \bmod p \in B, j\in[m]} T[i+j]\right)$$

For any i.



text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \mod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \mod p \in B, j \in [m]} T[i+j]\right)$$

For any i.

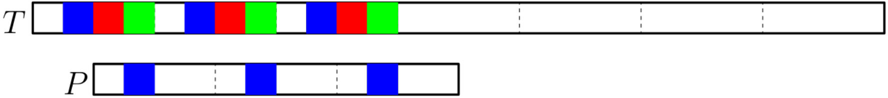
text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} T[i+j]\right)$$

For any i.



text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} T[i+j]\right)$$

For any i.

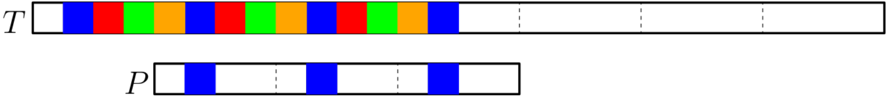
text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} T[i+j]\right)$$

For any i.



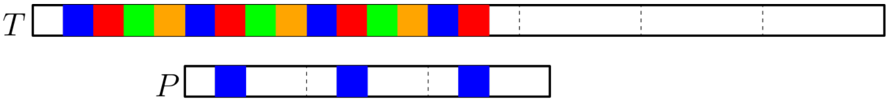
text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \mod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \mod p \in B, j \in [m]} T[i+j]\right)$$

For any i.



text-pattern P, T hamming distance

Compute

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} P[j]\right)$$

$$h\left(\bigodot_{j \bmod p \in B, j \in [m]} T[i+j]\right)$$

For any i.

text-pattern P, T hamming distance

For each color in T, we have

Lemma

Given a string S with $|S| = n/p$, computing $h(S')$ for any fix length substring S' of S cost $O(|S|) = O(n/p)$ running time.

text-pattern P, T hamming distance

For each color in T, we have

Lemma

Given a string S with $|S| = n/p$, computing $h(S')$ for any fix length substring S' of S cost $O(|S|) = O(n/p)$ running time.

$O\left(\frac{n}{p} \cdot p \cdot |B|\right) = O(n \log n)$ for T.

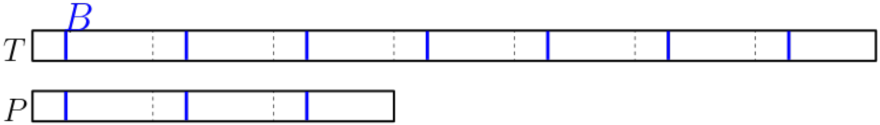
$O\left(\frac{m}{p} \cdot |B|\right) = O(n/k)$ for P.

Improve running time

$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.

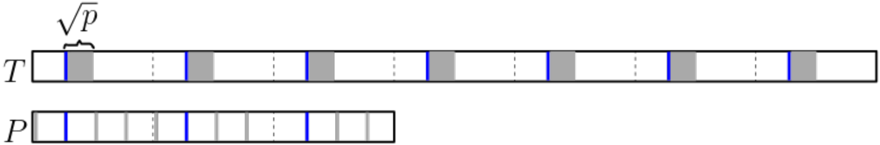
Improve running time

$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.



Improve running time

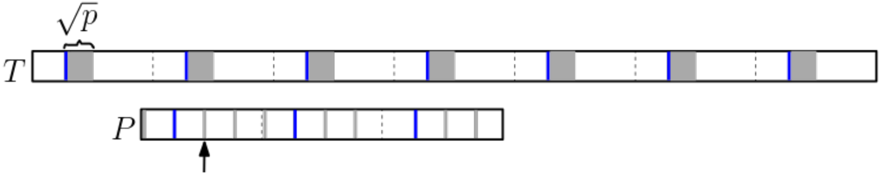
$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.



Colors $B + (1...\sqrt{p})$ in T and colors $B + (1...\sqrt{p}) \cdot \sqrt{p}$ in P.

Improve running time

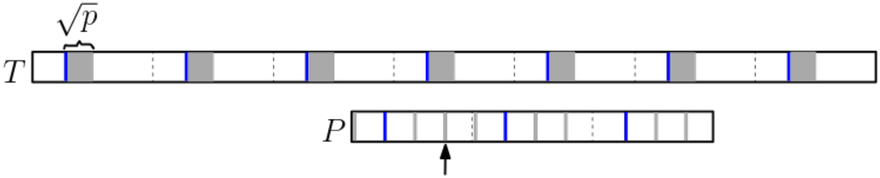
$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.



Colors $B + (1 \ldots \sqrt{p})$ in T and colors $B + (1 \ldots \sqrt{p}) \cdot \sqrt{p}$ in P.

Improve running time

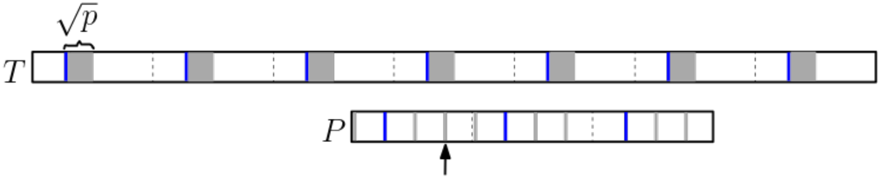
$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.



Colors $B + (1 ... \sqrt{p})$ in T and colors $B + (1 ... \sqrt{p}) \cdot \sqrt{p}$ in P.

Improve running time

$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.



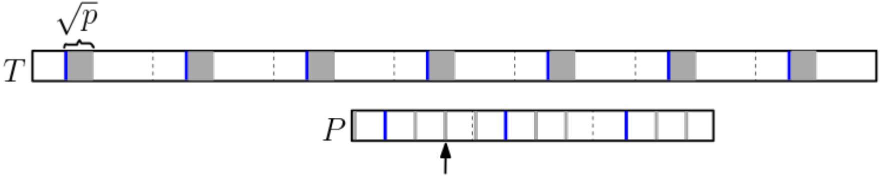
Colors $B + (1 \ldots \sqrt{p})$ in T and colors $B + (1 \ldots \sqrt{p}) \cdot \sqrt{p}$ in P.

$$O\left(\frac{n}{p} \cdot \sqrt{p} \cdot |B|\right) = O(n\sqrt{\log n}/\sqrt{k}) \text{ for } T.$$

$$O\left(\frac{m}{p} \cdot \sqrt{p} \cdot |B|\right) = O(n\sqrt{\log n}/\sqrt{k}) \text{ for } P.$$

Improve running time

$\sqrt{p}$ colors in T and $\sqrt{p}$ colors in P.



Colors $B + (1 \ldots \sqrt{p})$ in T and colors $B + (1 \ldots \sqrt{p}) \cdot \sqrt{p}$ in P.

$$O\left(\frac{n}{p} \cdot \sqrt{p} \cdot |B|\right) = O(n\sqrt{\log n}/\sqrt{k}) \text{ for } T.$$
$$O\left(\frac{m}{p} \cdot \sqrt{p} \cdot |B|\right) = O(n\sqrt{\log n}/\sqrt{k}) \text{ for } P.$$

Correctness: uniform random B (blue line) $\Rightarrow$ for any position of P, uniform random allignment.

Algorithm

For each $k = 2^i$ where $i \in [\log m]$ do:

1. Let p be a random prime from $\hat{p}, 2\hat{p}$ where $\hat{p} = \Theta(k \log m)$.
   $O(k \log m)$

Algorithm

For each $k = 2^i$ where $i \in [\log m]$ do:

1. Let p be a random prime from $\hat{p}, 2\hat{p}$ where $\hat{p} = \Theta(k \log m)$. $O(k \log m)$
2. Uniformly sample $B \in [p], |B| = \log m$ for c times. $O(\log m)$

Algorithm

For each $k = 2^i$ where $i \in [\log m]$ do:

1. Let p be a random prime from $\hat{p}, 2\hat{p}$ where $\hat{p} = \Theta(k \log m)$. $O(k \log m)$
2. Uniformly sample $B \in [p], |B| = \log m$ for c times. $O(\log m)$
3. Compute $h\left(\bigodot_{k+v\sqrt{p} \mod p \in B, k \in [m]} P[k]\right)$ for any sampled B and any $v \in [\sqrt{p}]$
   $O\left(\frac{m}{p} \cdot \frac{p}{k} \cdot \sqrt{p}\right) = O\left(\frac{m\sqrt{p}}{k}\right) = O\left(m\sqrt{\log m}/\sqrt{k}\right)$;
   $h\left(\bigodot_{k+i-u \mod p \in B, k \in [m]} T[i+k-1]\right)$ for any sampled B, any shift $u \in [\sqrt{p}]$ and for any i
   $O\left(\frac{n}{p} \cdot \frac{p}{k} \cdot \sqrt{p}\right) = O\left(n\sqrt{\log m}/\sqrt{k}\right)$.

Algorithm

For each $k = 2^i$ where $i \in [\log m]$ do:

1. Let p be a random prime from $\hat{p}, 2\hat{p}$ where $\hat{p} = \Theta(k \log m)$. $O(k \log m)$
2. Uniformly sample $B \in [p], |B| = \log m$ for c times. $O(\log m)$
3. Compute $h\left(\bigodot_{k+v\sqrt{p} \mod p \in B, k\in[m]} P[k]\right)$ for any sampled B and any $v \in [\sqrt{p}]$ $O\left(m\sqrt{\log m}/\sqrt{k}\right)$;
   $h\left(\bigodot_{k+i-u \mod p\in B, k\in[m]} T[i+k-1]\right)$ for any sampled B, any shift $u \in [\sqrt{p}]$ and for any i $O\left(n\sqrt{\log m}/\sqrt{k}\right)$.
4. For any i, let the number of B that $h\left(\bigodot_{k+v_i\sqrt{n} \mod p\in B, k\in[m]} P[k]\right) = h\left(\bigodot_{k+i-u_i\sqrt{q} \mod p\in B, k\in[m]} T[i+k-1]\right)$ be d_i. Let $\tilde{d}_i = f^{-1}(d_i)$. If $\tilde{d}_i/c$ is not too closer to 0 or 1, then let $f(\tilde{d}_i)$ be the estimation for $HD(P, T[i, ..., i+m-1])$.$O(n)$

Algorithm

For each $k = 2^i$ where $i \in [\log m]$ do:

1. Let p be a random prime from $\hat{p}, 2\hat{p}$ where $\hat{p} = \Theta(k \log m)$. $O(k \log m)$
2. Uniformly sample $B \in [p], |B| = \log m$ for c times. $O(\log m)$
3. Compute $h\left(\bigodot_{k+v\sqrt{p} \mod p \in B, k \in [m]} P[k]\right)$ for any sampled B and any $v \in [\sqrt{p}]$ $O\left(m\sqrt{\log m}/\sqrt{k}\right)$; $h\left(\bigodot_{k+i-u \mod p \in B, k \in [m]} T[i+k-1]\right)$ for any sampled B, any shift $u \in [\sqrt{p}]$ and for any i $O\left(n\sqrt{\log m}/\sqrt{k}\right)$.

For any i:

- Binary search for k:
  1. For any i, let the number of B that $h\left(\bigodot_{k+v_i\sqrt{n} \mod p \in B, k \in [m]} P[k]\right) = h\left(\bigodot_{k+i-u_i\sqrt{q} \mod p \in B, k \in [m]} T[i+k-1]\right)$ be d_i. Let $\tilde{d}_i = f^{-1}(d_i)$. If $\tilde{d}_i/c$ is not too closer to 0 or 1, then let $f(\tilde{d}_i)$ be the estimation for $HD(P, T[i, ..., i+m-1])$. $O(n)$

Correctness

Lemma

For any i **and any** k, *the algorithm returns a correct estimation of* $HD(P, T[i, ..., i+m-1])$, *or corretly determine whether* k *is too large or too small with large constant probability.*

Correctness

Lemma

For any i **and any** k, *the algorithm returns a correct estimation of* $HD(P, T[i, ..., i+m-1])$, *or corretly determine whether* k *is too large or too small with large constant probability.*

Boost the probability by running $O(\log n)$ times of the algorithm and take the **median** of each result for each i and k.

Correctness

Lemma

For any i **and any** k, *the algorithm returns a correct estimation of* $HD(P, T[i, ..., i+m-1])$, *or corretly determine whether* k *is too large or too small with large constant probability.*

Boost the probability by running $O(\log n)$ times of the algorithm and take the **median** of each result for each i and k.

Total running time

$$O(\log n) \cdot \left(\sum_{k \in [logm]} O\left(\frac{m\sqrt{\log m}}{\sqrt{k}} \right) + \log\log n \cdot O(n) \right)$$

$$= O(n \log^{1.5} n)$$