Streaming and communication complexity of Hamming distance

Tatiana Starikovskaya
IRIF, Université Paris-Diderot

(Joint work with Raphaël Clifford, ICALP'16)

Approximate pattern matching

Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Approximate pattern matching

Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

"Big Data" Applications

- Computational biology
- Signal processing
- Text retrieval

Standard algorithms: $\Omega(n)$ space

Model of computation

Problem

Pattern P of length n, text T
Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

Text T

c

Model of computation

Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

Text T

c a

Model of computation

Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

Text T

c a a

Model of computation

Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

Text T

c	a	a	b

Model of computation

Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

Text T

c a a b c

Model of computation

Problem

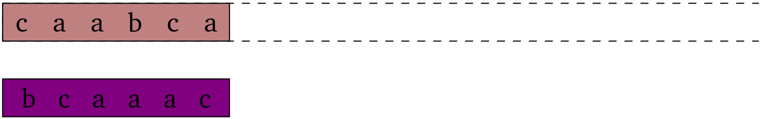
Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

Text T



Pattern P

Model of computation

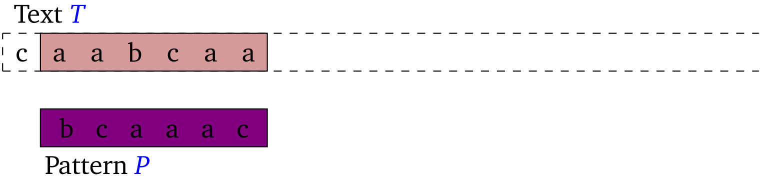
Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**



Model of computation

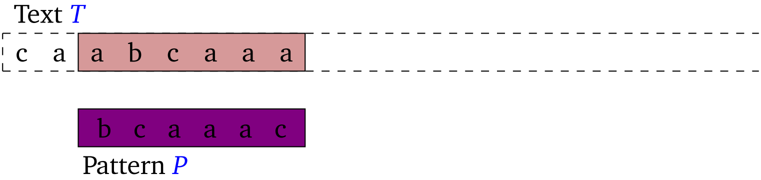
Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**



Model of computation

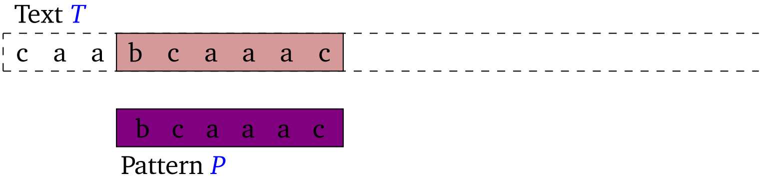
Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**



Model of computation

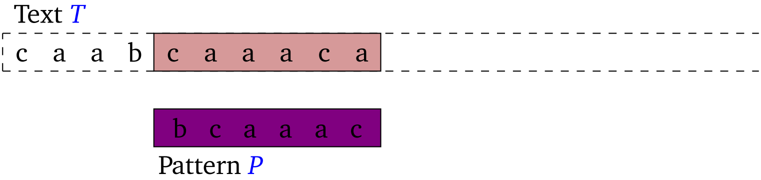
Problem

Pattern P of length n, text T

Find the Hamming distance between P and each n-length substring of T

Model

- T = stream of characters
- Length of the text and size of the universe are extremely large
- Can't store a copy of T or P
- Space = total space used; **Time = time per character of T**

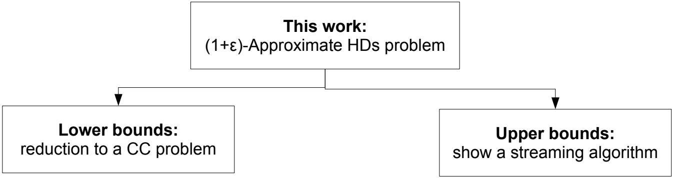


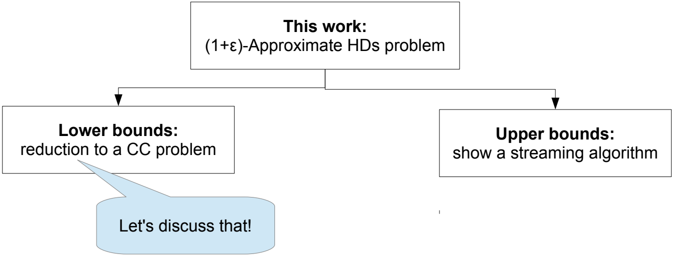
What is known: Hamming distance

- All distances
  - Space $\Omega(n)$ **[Folklore]**
  - Time $\mathcal{O}(\log^2 n)$ **[Clifford et al., CPM'11]**

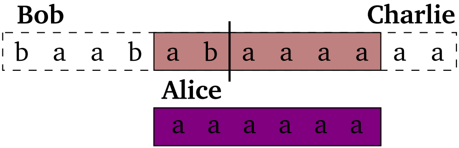
What is known: Hamming distance

- All distances
  - Space $\Omega(n)$ **[Folklore]**
  - Time $\mathcal{O}(\log^2 n)$ **[Clifford et al., CPM'11]**
- Only distances $\leq k$ **[Clifford et al., SODA'16]**
  - Exact values: space $\mathcal{O}(k^2 \operatorname{polylog} n)$, time $\mathcal{O}(\sqrt{k}\log k + \operatorname{polylog} n)$
  - $(1+\varepsilon)$-approx.: space $\mathcal{O}(\varepsilon^{-2}k^2 \operatorname{polylog} n)$, time $\mathcal{O}(\varepsilon^{-2} \operatorname{polylog} n)$


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
Let's discuss that!

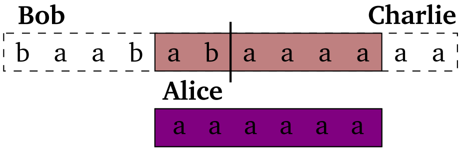
Lower bound for all HDs, approximate



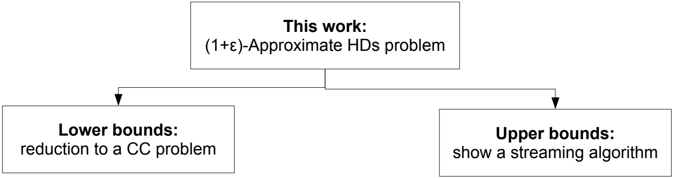
3-parties CC problem

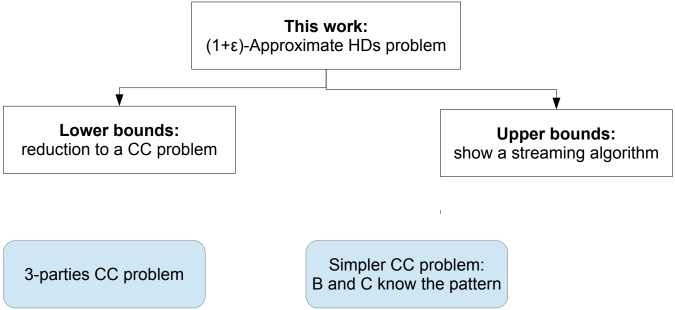
- **Alice** holds the pattern, **Bob** holds $T[1,n]$, **Charlie** holds $T[n+1,2n]$
- **Charlie**'s output: $(1+\varepsilon)$-HD for each alignment of P and T
  Min. communication between **Alice**, **Bob**, and **Charlie**?

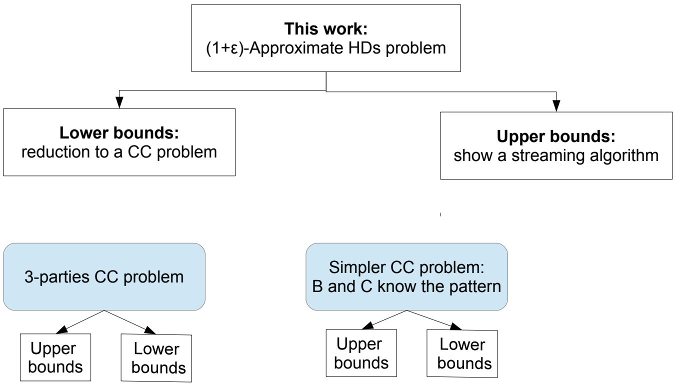
Lower bound for all HDs, approximate

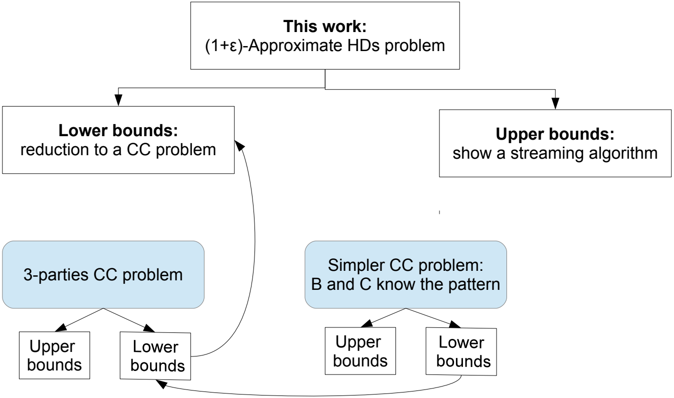


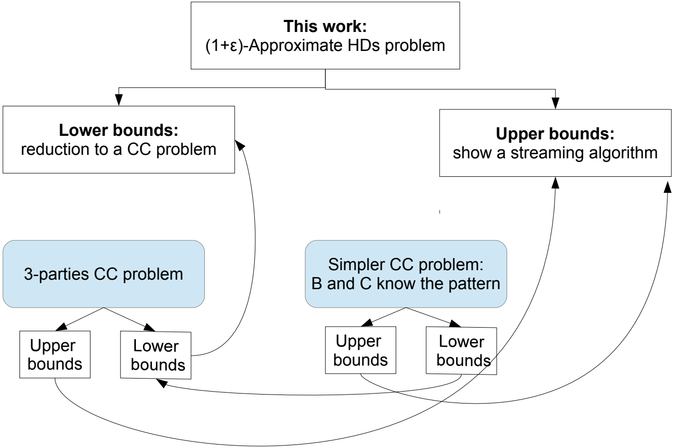
- Streaming algorithm: T = stream, not allowed to store a copy of P or T, output = $(1+\varepsilon)$-HDs
- At time $= n$ it stores all the information needed to compute the $(1+\varepsilon)$-HDs
- Comm. protocol: send this information from **A** and **B** to **C**
- Lower bound for the CC problem $\Rightarrow$ streaming lower bound


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
3-parties CC problem
Simpler CC problem:
B and C know the pattern


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
3-parties CC problem
Simpler CC problem:
B and C know the pattern
Upper bounds
Lower bounds
Upper bounds
Lower bounds

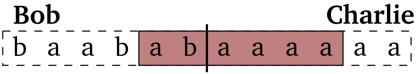

This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
3-parties CC problem
Upper bounds
Lower bounds
Simpler CC problem:
B and C know the pattern
Upper bounds
Lower bounds


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
3-parties CC problem
Simpler CC problem:
B and C know the pattern
Upper bounds
Lower bounds
Upper bounds
Lower bounds

Communication complexity

Simpler CC problem: **B** and **C** know the pattern

Lower bound: $\Omega(\varepsilon^{-1} \log^2 \varepsilon^{-1} n)$



- Window counting: $(1+\varepsilon)$-approx. of #(b) in a sliding window of width n = $(1+\varepsilon)$-approx. of HD between $P = aa\ldots a$ and T
- $\Omega(\varepsilon^{-1} \log^2 \varepsilon^{-1} n)$ bits **[Datar et al., 2013]**

3-parties CC problem

Lower bound: $\Omega(\varepsilon^{-1}\log^2\varepsilon^{-1}n+\varepsilon^{-2}\log n)$

Bob **Charlie**

b a a b a b | a a a a a a

- Output = $(1+\varepsilon)$-HD between $T[1,n]$ and $T[n+1,2n]$ = $(1+\varepsilon)$-approx. of HD between $T = T[1,n]00\ldots0$ (**Bob** and **Charlie**) and $P = T[n+1,2n]$ (**Alice**)
- $\Omega(\varepsilon^{-2}\log n)$ bits **[Jayram & Woordruff, 2013]**

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Intuition

- Sketch of a string is a **very short** vector
- L_2-distance between sketches $\approx$ HD between strings

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Intuition

- Sketch of a string is a **very short** vector
- L_2-distance between sketches $\approx$ HD between strings

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \underbrace{\begin{pmatrix} \pm 1 & \pm 1 & \dots \\ \pm 1 & \ddots & \\ \vdots & & \end{pmatrix}}_{Y} \underbrace{\begin{pmatrix} S[1] \\ S[2] \\ \vdots \end{pmatrix}}_{S}$$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$$(1-\varepsilon)\cdot HD(S_1,S_2) \le \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \le (1+\varepsilon)\cdot HD(S_1,S_2)$$

Proof

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$$(1-\varepsilon)\cdot HD(S_1,S_2) \le \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \le (1+\varepsilon)\cdot HD(S_1,S_2)$$

Proof

$$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = \mathbb{E}[\varepsilon^2 \cdot |Y(S_1 - S_2)|_2^2] = \varepsilon^2 \cdot \mathbb{E}[|Y(S_1-S_2)|_2^2] =$$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$(1-\varepsilon)\cdot HD(S_1,S_2) \leq \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \leq (1+\varepsilon)\cdot HD(S_1,S_2)$

Proof

$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = \mathbb{E}[\varepsilon^2 \cdot |Y(S_1 - S_2)|_2^2] = \varepsilon^2 \cdot \mathbb{E}[|Y(S_1 - S_2)|_2^2] =$
$= \varepsilon^2 \cdot \mathbb{E}[\sum_{j=1}^{1/\varepsilon^2} \left(Y_j(S_1 - S_2)\right)^2] = \mathbb{E}[\left(Y_1(S_1 - S_2)\right)^2] = |S_1 - S_2|_2^2$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length} = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$(1-\varepsilon)\cdot HD(S_1, S_2) \leq \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \leq (1+\varepsilon)\cdot HD(S_1, S_2)$

Proof

$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = |S_1 - S_2|_2^2$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$(1-\varepsilon) \cdot HD(S_1, S_2) \leq \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \leq (1+\varepsilon) \cdot HD(S_1, S_2)$

Proof

$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = |S_1 - S_2|_2^2$

$\mathbf{Var}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = \varepsilon^2 \cdot \mathbf{Var}[\big(Y_1(S_1 - S_2)\big)^2] \leq$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$(1-\varepsilon)\cdot HD(S_1,S_2) \le \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \le (1+\varepsilon)\cdot HD(S_1,S_2)$

Proof

$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = |S_1 - S_2|_2^2$

$\mathbf{Var}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = \varepsilon^2 \cdot \mathbf{Var}[\left(Y_1(S_1 - S_2)\right)^2] \le$
$\le \varepsilon^2 \cdot \mathbb{E}[\left(Y_1(S_1 - S_2)\right)^4] \le \varepsilon^2 C \cdot \mathbb{E}[\left(Y_1(S_1 - S_2)\right)^2]^2 = \varepsilon^2 C \cdot |S_1 - S_2|_2^4$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length} = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$$(1-\varepsilon)\cdot HD(S_1,S_2) \leq \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \leq (1+\varepsilon)\cdot HD(S_1,S_2)$$

Proof

$$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = |S_1 - S_2|_2^2$$

$$\mathbf{Var}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] \leq \varepsilon^2 C \cdot |S_1 - S_2|_2^4$$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

Formal definition (binary alphabets)

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables

$$\underbrace{sketch(S)}_{\text{length } = 1/\varepsilon^2} = \mathbf{Y}S$$

Lemma

$(1-\varepsilon)\cdot HD(S_1, S_2) \leq \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \leq (1+\varepsilon)\cdot HD(S_1, S_2)$

Proof

$\mathbb{E}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] = |S_1 - S_2|_2^2$

$\mathbf{Var}[\varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2] \leq \varepsilon^2 C \cdot |S_1 - S_2|_2^4$

By Chebyshev's inequality, with constant probability:

$(1-\varepsilon)\cdot |S_1 - S_2|_2^2 \leq \varepsilon^2 \cdot |sketch(S_1) - sketch(S_2)|_2^2 \leq (1+\varepsilon)\cdot |S_1 - S_2|_2^2$

Important notion: $(1+\varepsilon)$-approximate sketch for HD

One more trick

- Y can be generated from $\mathcal{O}(\log n)$ random bits (random $\rightarrow$ preudorandom)

Important notion: $(1+\varepsilon)$-approximate sketch for HD

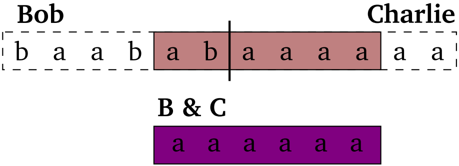
One more trick

- Y can be generated from $\mathcal{O}(\log n)$ random bits (random $\rightarrow$ preudorandom)

Summary

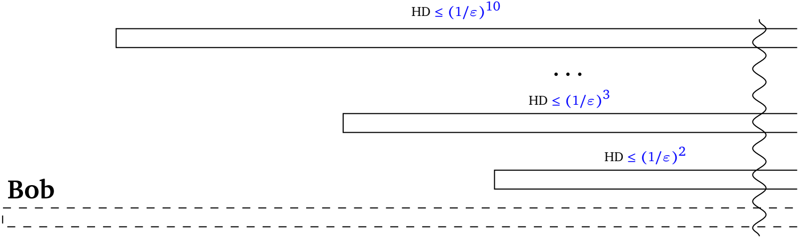
- Sketch of a string is a vector of length $\mathcal{O}(\varepsilon^{-2}\log n)$ bits
- Sketches give $(1+\varepsilon)$-approximation of HD

Simpler CC problem: **B** and **C** know the pattern



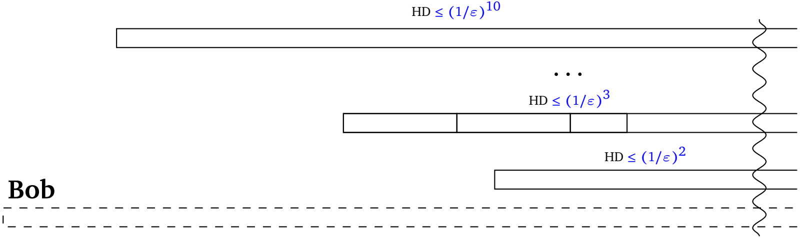
- **B** knows $T[1, n]$, **C** knows $T[n+1, 2n]$, **B** and **C** know P
- **Observation:** **C** doesn't need any information to compute HDs between suffixes of P and $T[n+1, 2n]$

Simpler CC problem: **B** and **C** know the pattern



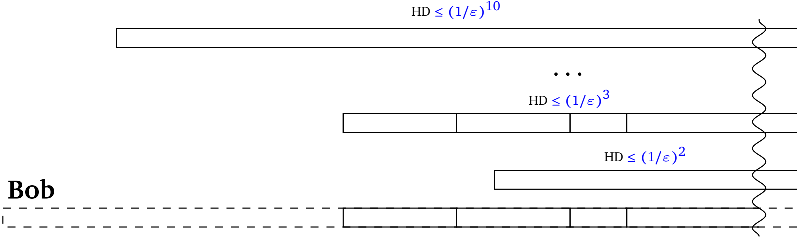
- Select $\mathcal{O}(\log_\varepsilon n)$ prefixes **of the pattern**
- First prefix: Prefix of maximal length ℓ_1 with HD $\leq (1/\varepsilon)^2$
- Second prefix: Prefix of maximal length $\ell_2 \geq \ell_1$ with HD $\leq (1/\varepsilon)^3$
- ...

Simpler CC problem: **B** and **C** know the pattern



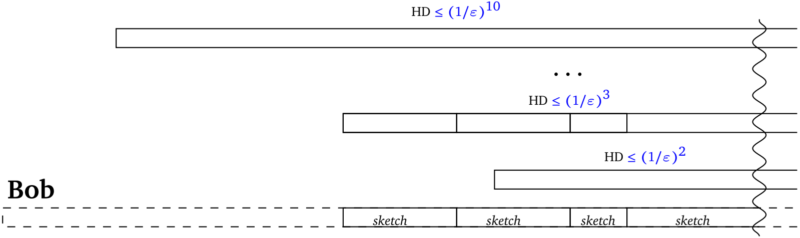
- Divide prefix j into $1/\varepsilon^2$ blocks with HD $\leq (1/\varepsilon)^{j-1}$

Simpler CC problem: **B** and **C** know the pattern



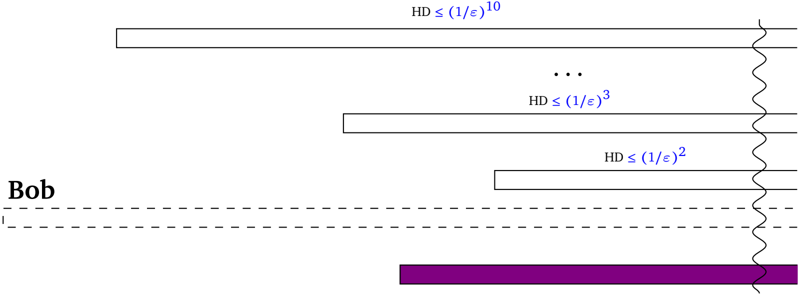
- Divide prefix j into $1/\varepsilon^2$ blocks with HD $\leq (1/\varepsilon)^{j-1}$
- Compute $\mathcal{O}(1/\varepsilon^2)$ sketches for the text

Simpler CC problem: **B** and **C** know the pattern

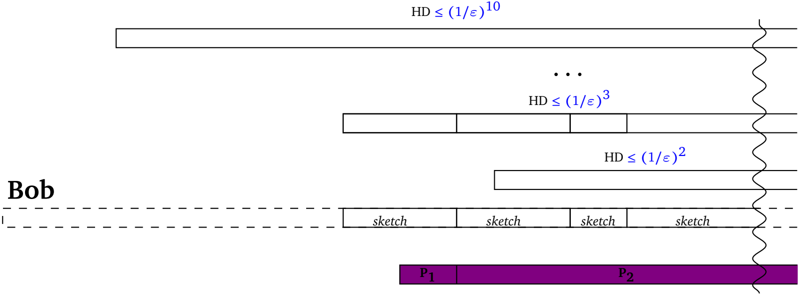


- Divide prefix j into $1/\varepsilon^2$ blocks with HD $\leq (1/\varepsilon)^{j-1}$
- Compute $\mathcal{O}(1/\varepsilon^2)$ sketches for the text
- Send the block borders and the sketches to **Charlie**

Simpler CC problem: **B** and **C** know the pattern

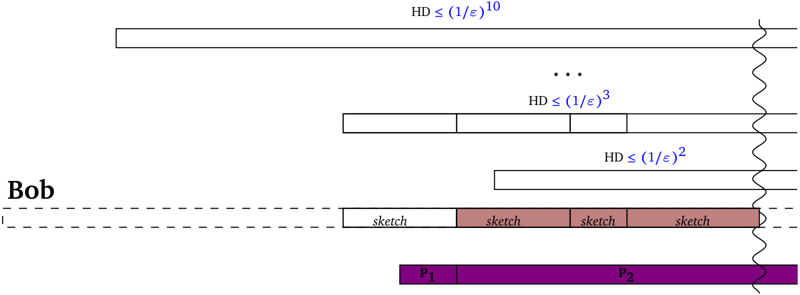


Simpler CC problem: **B** and **C** know the pattern



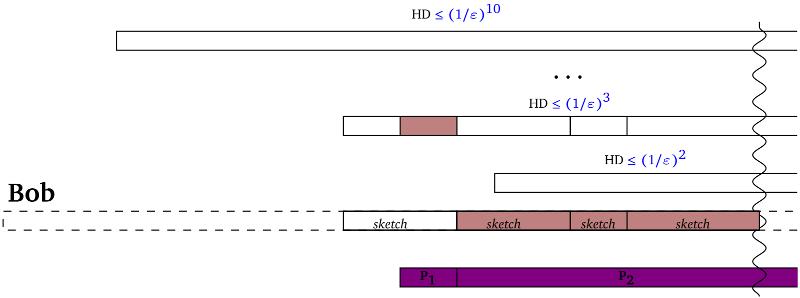
- Find the shortest prefix containing P

Simpler CC problem: **B** and **C** know the pattern



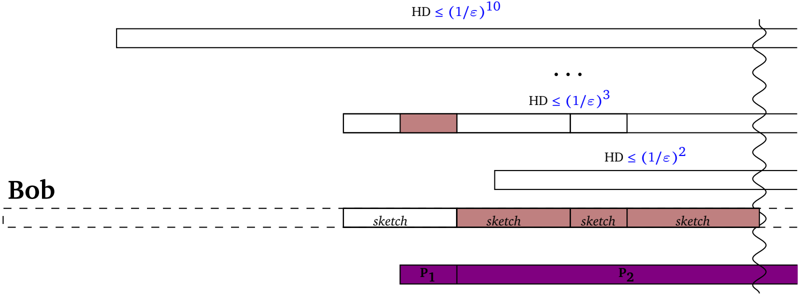
- Find the shortest prefix containing P
- $\mathrm{HD}(P_2, T)$: use sketches — $(1+\varepsilon)$-approximation

Simpler CC problem: **B** and **C** know the pattern

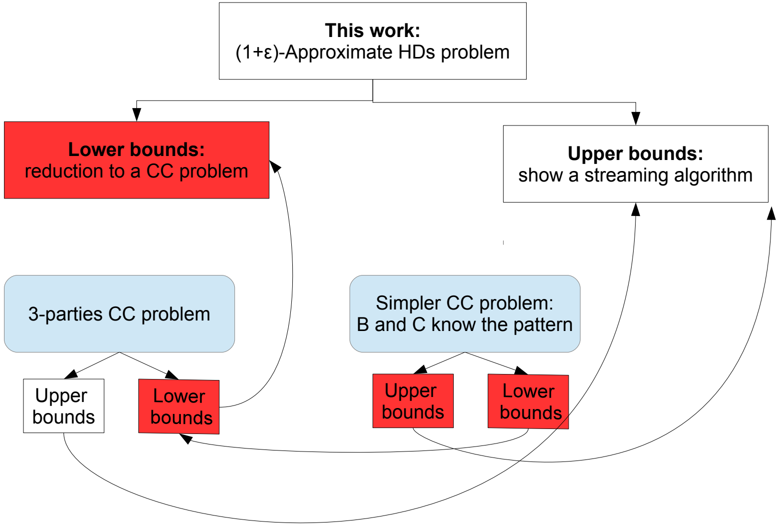


- Find the shortest prefix containing P
- $\mathrm{HD}(P_2, T)$: use sketches — $(1+\varepsilon)$-approximation
- $\mathrm{HD}(P_1, T)$: use the prefix's block — additive error $\leq \varepsilon \cdot HD(P,T)$

Simpler CC problem: **B** and **C** know the pattern

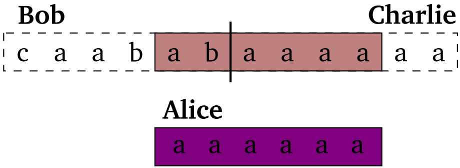


- Find the shortest prefix containing P
- $\mathrm{HD}(P_2, T)$: use sketches — $(1+\varepsilon)$-approximation
- $\mathrm{HD}(P_1, T)$: use the prefix's block — additive error $\le \varepsilon \cdot HD(P,T)$
- **CC** = $\mathcal{O}(\varepsilon^{-4}\log^2 n)$ **[Lower bound: $\Omega(\varepsilon^{-1}\log^2 \varepsilon^{-1} n)$]**



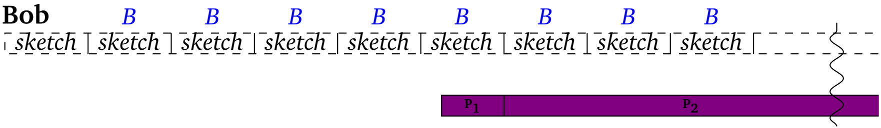
This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
3-parties CC problem
Simpler CC problem:
B and C know the pattern
Upper bounds
Lower bounds
Upper bounds
Lower bounds

3-parties CC problem



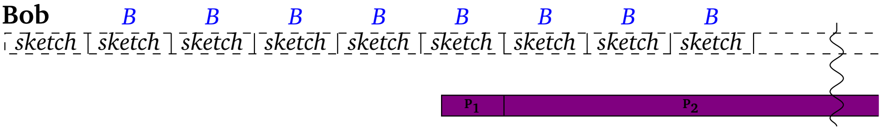
- **B** knows $T[1, n]$, **C** knows $T[n+1, 2n]$, only **A** knows P
- **Observation: C** doesn't need any information to compute HDs between suffixes of P and his part of the text
- Can't use prefixes of P to approximate T — **C** doesn't know P

3-parties CC problem



- Divide the text T into blocks of length $B = \sqrt{n}$
- Compute a sketch of each block
- Large Hamming distance: HD (prefix of P, T) $\geq B/\varepsilon$
  - HD(P_1, T): use sketches to compute $(1+\varepsilon)$-approx. H'
  - HD(P_2, T): ignore

3-parties CC problem



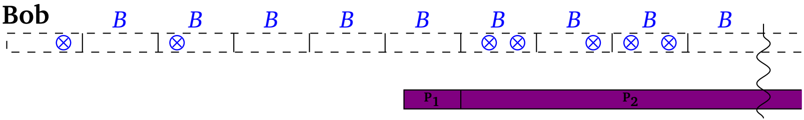
- Divide the text T into blocks of length $B = \sqrt{n}$
- Compute a sketch of each block
- Large Hamming distance: HD (prefix of P, T) $\geq B/\varepsilon$
  - HD(P_1, T): use sketches to compute $(1+\varepsilon)$-approx. H'
  - HD(P_2, T): ignore

Lemma
H' is a good approximation of HD

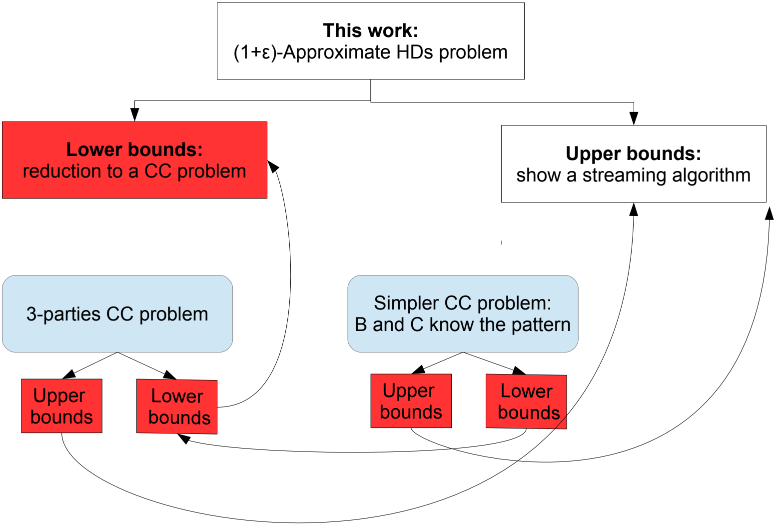
Proof

1. $H' \leq (1+\varepsilon) \cdot HD(P_2, T) \leq (1+\varepsilon) \cdot HD$
2. $H' \geq (1-\varepsilon) \cdot HD(P_2, T) \geq (1-\varepsilon) \cdot HD - HD(P_1, T) \geq (1-2\varepsilon) \cdot HD$

3-parties CC problem

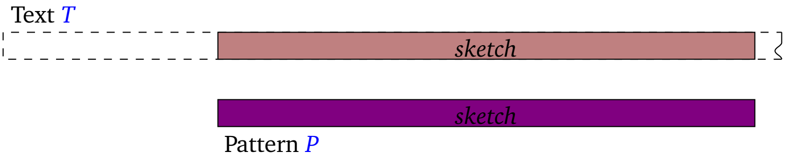


- Small Hamming distance: HD (prefix of P, T) $\geq B/\varepsilon$
  - If $\#(\otimes)$ in a block ≤ 1, **B** sends it to **C**
  - Starting from the first block where $\#(\otimes) \geq 2$, T and P can be encoded in small space (periodicity)
  - **C** can restore P and T from the encoding and compute HDs
- **CC** = $\mathcal{O}(1/\varepsilon^2\sqrt{n}\log n)$ ☺
  [Lower bound: $\Omega(\varepsilon^{-2}\log n + \varepsilon^{-1}\log^2 \varepsilon^{-1} n)$**]**


This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
show a streaming algorithm
3-parties CC problem
Simpler CC problem:
B and C know the pattern
Upper bounds
Lower bounds
Upper bounds
Lower bounds

Streaming algorithm

Streaming algorithm



Reminder

- $Y = 1/\varepsilon^2 \times n$ matrix of IID unbiased ± 1 random variables
- **sketch**$(\mathbf{S}) = Y \cdot S$

Problem

- How to maintain the sketch of T?
- We don't have random access to T and we can't store many of its characters

Streaming algorithm

B	B	B	B
sketch	*sketch*	*sketch*	*sketch*

super-sketch

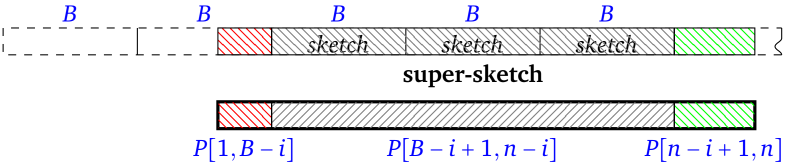
Reminder

- $Y = (1/\varepsilon^2) \times n$ matrix of IID unbiased ± 1 random variables
- $sketch(S) = Y \cdot S$

New notion: super-sketch

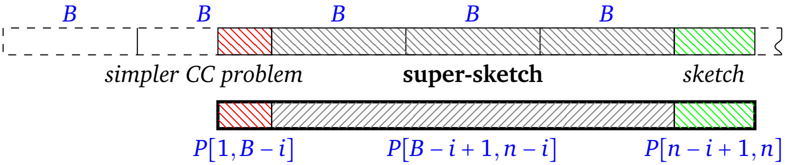
- σ_i — IID unbiased ± 1 variables
- **super-sketch** $= \sum \sigma_i \cdot sketch_i$
- Analysis: similar to sketches

Streaming algorithm



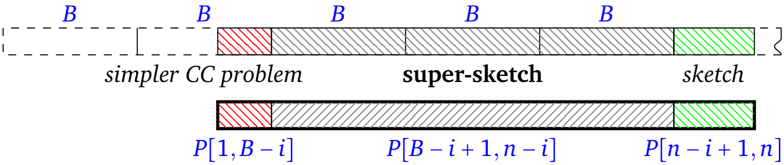
- HD between $P[B-i+1, n-i]$ and T: **super-sketch**
- Store a **super-sketch** for each $(n-B)$-length substring of P
  - $B = \sqrt{n}/\varepsilon$ **super-sketches** in total
- At each block border compute a **super-sketch** of the last n/B blocks from their sketches
  - $\mathcal{O}(n/B) = \mathcal{O}(\varepsilon\sqrt{n})$ time, can be de-amortized

Streaming algorithm

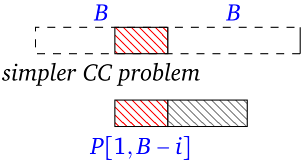


- HD between the **suffix** of P and T: sketch

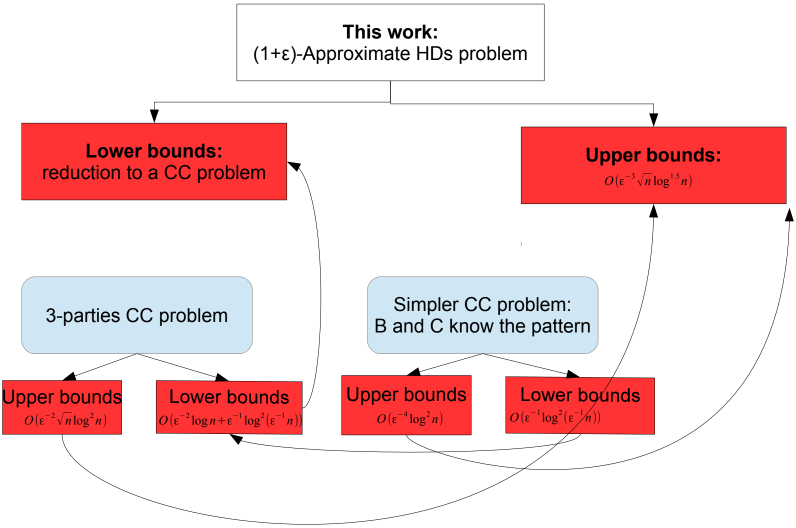
Streaming algorithm



- HD between the **suffix** of P and T: sketch
- HD between the **prefix** of P and T: similar to the simpler CC problem for the pattern $P[1, B]$



Complexity: $\mathcal{O}(1/\varepsilon^3 \sqrt{n} \log^2 n)$ bits of space, $\mathcal{O}(1/\varepsilon^2 \log^2 n)$ time ☺



This work:
(1+ε)-Approximate HDs problem
Lower bounds:
reduction to a CC problem
Upper bounds:
$O(\varepsilon^{-3}\sqrt{n}\log^{1.5} n)$
3-parties CC problem
Simpler CC problem:
B and C know the pattern
Upper bounds
$O(\varepsilon^{-2}\sqrt{n}\log^2 n)$
Lower bounds
$O(\varepsilon^{-2}\log n+\varepsilon^{-1}\log^2(\varepsilon^{-1}n))$
Upper bounds
$O(\varepsilon^{-4}\log^2 n)$
Lower bounds
$O(\varepsilon^{-1}\log^2(\varepsilon^{-1}n))$