

CMPSCI 711: “Really Advanced Algorithms”

Lecture 14 – Finger Printing

Andrew McGregor

Schwartz-Zippel

String Equality and Some Communication Complexity

Pattern Matching

Readings

Outline

Schwartz-Zippel

String Equality and Some Communication Complexity

Pattern Matching

Readings

Schwartz-Zippel

Problem

Given three n variable polynomials P_1, P_2, P_3 over the field \mathbb{F} . Can you test if

$$P_1(x_1, \dots, x_n) \times P_2(x_1, \dots, x_n) = P_3(x_1, \dots, x_n)$$

faster than multiplying the polynomials? Equivalently, is

$$Q(x_1, \dots, x_n) = P_1(x_1, \dots, x_n) \times P_2(x_1, \dots, x_n) - P_3(x_1, \dots, x_n)$$

zero for all x_1, \dots, x_n ?

Schwartz-Zippel

Problem

Given three n variable polynomials P_1, P_2, P_3 over the field \mathbb{F} . Can you test if

$$P_1(x_1, \dots, x_n) \times P_2(x_1, \dots, x_n) = P_3(x_1, \dots, x_n)$$

faster than multiplying the polynomials? Equivalently, is

$$Q(x_1, \dots, x_n) = P_1(x_1, \dots, x_n) \times P_2(x_1, \dots, x_n) - P_3(x_1, \dots, x_n)$$

zero for all x_1, \dots, x_n ?

Theorem (Schwartz-Zippel)

Let $Q(x_1, \dots, x_n)$ be a non-zero multivariate polynomial \mathbb{F} of total degree d . Fix any finite set $S \subset \mathbb{F}$ and let r_1, \dots, r_n be chosen independently and uniformly at random from S . Then,

$$\mathbb{P}[Q(r_1, \dots, r_n) = 0] \leq d/|S|$$

Schwartz-Zippel Proof

- ▶ Induction on n : For $n = 1$, because Q has at most d roots,

$$\mathbb{P}[Q(r_1) = 0] \leq d/|S|$$

Schwartz-Zippel Proof

- ▶ Induction on n : For $n = 1$, because Q has at most d roots,

$$\mathbb{P}[Q(r_1) = 0] \leq d/|S|$$

- ▶ For induction step define Q_i for $0 \leq i \leq k$:

$$Q(x_1, \dots, x_n) = \sum_{i=0}^k x_1^i Q_i(x_2, \dots, x_n)$$

where k is maximum such that $Q_k(x_2, \dots, x_n) \not\equiv 0$

Schwartz-Zippel Proof

- ▶ Induction on n : For $n = 1$, because Q has at most d roots,

$$\mathbb{P}[Q(r_1) = 0] \leq d/|S|$$

- ▶ For induction step define Q_i for $0 \leq i \leq k$:

$$Q(x_1, \dots, x_n) = \sum_{i=0}^k x_1^i Q_i(x_2, \dots, x_n)$$

where k is maximum such that $Q_k(x_2, \dots, x_n) \not\equiv 0$

- ▶ Since total degree of Q_k is at most $d - k$,

$$\mathbb{P}[Q_k(r_2, \dots, r_n) = 0] \leq (d - k)/|S|$$

Schwartz-Zippel Proof

- ▶ Induction on n : For $n = 1$, because Q has at most d roots,

$$\mathbb{P}[Q(r_1) = 0] \leq d/|S|$$

- ▶ For induction step define Q_i for $0 \leq i \leq k$:

$$Q(x_1, \dots, x_n) = \sum_{i=0}^k x_1^i Q_i(x_2, \dots, x_n)$$

where k is maximum such that $Q_k(x_2, \dots, x_n) \neq 0$

- ▶ Since total degree of Q_k is at most $d - k$,

$$\mathbb{P}[Q_k(r_2, \dots, r_n) = 0] \leq (d - k)/|S|$$

- ▶ Consider $q(x) = Q(x, r_2, \dots, r_n)$,

$$\mathbb{P}[q(r_1) = 0 | Q_k(r_2, \dots, r_n) \neq 0] \leq k/|S|$$

Schwartz-Zippel Proof

- ▶ Induction on n : For $n = 1$, because Q has at most d roots,

$$\mathbb{P}[Q(r_1) = 0] \leq d/|S|$$

- ▶ For induction step define Q_i for $0 \leq i \leq k$:

$$Q(x_1, \dots, x_n) = \sum_{i=0}^k x_1^i Q_i(x_2, \dots, x_n)$$

where k is maximum such that $Q_k(x_2, \dots, x_n) \neq 0$

- ▶ Since total degree of Q_k is at most $d - k$,

$$\mathbb{P}[Q_k(r_2, \dots, r_n) = 0] \leq (d - k)/|S|$$

- ▶ Consider $q(x) = Q(x, r_2, \dots, r_n)$,

$$\mathbb{P}[q(r_1) = 0 | Q_k(r_2, \dots, r_n) \neq 0] \leq k/|S|$$

- ▶ Putting together gives $\mathbb{P}[Q(r_1, \dots, r_n) = 0]$ at most

$$\mathbb{P}[Q_k(r_2, \dots, r_n) = 0] + \mathbb{P}[q(r_1) = 0 | Q_k(r_2, \dots, r_n) \neq 0] \leq d/|S|$$

Bipartite Perfect Matching

Definition

Let $G = (U, V, E)$ be a bipartite graph on $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$. $M \subset E$ is a *matching* if each vertex occurs at most once in M . If $|M| = n$ then we say M is a *perfect matching*.

Bipartite Perfect Matching

Definition

Let $G = (U, V, E)$ be a bipartite graph on $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$. $M \subset E$ is a *matching* if each vertex occurs at most once in M . If $|M| = n$ then we say M is a *perfect matching*.

Theorem (Edmonds' Theorem)

Given G , let A be $n \times n$ matrix where

$$A_{i,j} = \begin{cases} x_{ij} & \text{if } (u_i, v_j) \in E \\ 0 & \text{if } (u_i, v_j) \notin E \end{cases}$$

Then $\det(A)$ is multivariate polynomial with maximum degree n .
 $\det(A) \equiv 0$ iff G has a perfect matching.

Bipartite Perfect Matching

Definition

Let $G = (U, V, E)$ be a bipartite graph on $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$. $M \subset E$ is a *matching* if each vertex occurs at most once in M . If $|M| = n$ then we say M is a *perfect matching*.

Theorem (Edmonds' Theorem)

Given G , let A be $n \times n$ matrix where

$$A_{i,j} = \begin{cases} x_{ij} & \text{if } (u_i, v_j) \in E \\ 0 & \text{if } (u_i, v_j) \notin E \end{cases}$$

Then $\det(A)$ is multivariate polynomial with maximum degree n .
 $\det(A) \equiv 0$ iff G has a perfect matching.

Schwartz-Zippel result gives randomized method for seeing if G has perfect matching. But it's actually not that interesting. . .

Outline

Schwartz-Zippel

String Equality and Some Communication Complexity

Pattern Matching

Readings

Verifying Equality of Strings

Problem

Suppose Alice has binary string (a_1, \dots, a_n) and Bob has binary string (b_1, \dots, b_n) . How many bits to this need to communicate to conclude (with high probability) that the strings are equal?

Verifying Equality of Strings

Problem

Suppose Alice has binary string (a_1, \dots, a_n) and Bob has binary string (b_1, \dots, b_n) . How many bits do they need to communicate to conclude (with high probability) that the strings are equal?

Protocol

- ▶ Alice and Bob define $a = \sum_{i \in [n]} a_i 2^{n-i}$ and $b = \sum_{i \in [n]} b_i 2^{n-i}$

Verifying Equality of Strings

Problem

Suppose Alice has binary string (a_1, \dots, a_n) and Bob has binary string (b_1, \dots, b_n) . How many bits to this need to communicate to conclude (with high probability) that the strings are equal?

Protocol

- ▶ Alice and Bob defines $a = \sum_{i \in [n]} a_i 2^{n-i}$ and $b = \sum_{i \in [n]} b_i 2^{n-i}$
- ▶ Alice randomly picks a prime $p \leq \tau = tn \log tn$

Verifying Equality of Strings

Problem

Suppose Alice has binary string (a_1, \dots, a_n) and Bob has binary string (b_1, \dots, b_n) . How many bits do they need to communicate to conclude (with high probability) that the strings are equal?

Protocol

- ▶ Alice and Bob defines $a = \sum_{i \in [n]} a_i 2^{n-i}$ and $b = \sum_{i \in [n]} b_i 2^{n-i}$
- ▶ Alice randomly picks a prime $p \leq \tau = tn \log tn$
- ▶ Alice transmits $F_p(a) = a \bmod p$ and p to Bob

Verifying Equality of Strings

Problem

Suppose Alice has binary string (a_1, \dots, a_n) and Bob has binary string (b_1, \dots, b_n) . How many bits do they need to communicate to conclude (with high probability) that the strings are equal?

Protocol

- ▶ Alice and Bob defines $a = \sum_{i \in [n]} a_i 2^{n-i}$ and $b = \sum_{i \in [n]} b_i 2^{n-i}$
- ▶ Alice randomly picks a prime $p \leq \tau = tn \log tn$
- ▶ Alice transmits $F_p(a) = a \bmod p$ and p to Bob
- ▶ Bob computes $F_p(b)$: Returns "equal" iff $F_p(a) = F_p(b)$

Verifying Equality of Strings

Problem

Suppose Alice has binary string (a_1, \dots, a_n) and Bob has binary string (b_1, \dots, b_n) . How many bits do they need to communicate to conclude (with high probability) that the strings are equal?

Protocol

- ▶ Alice and Bob defines $a = \sum_{i \in [n]} a_i 2^{n-i}$ and $b = \sum_{i \in [n]} b_i 2^{n-i}$
- ▶ Alice randomly picks a prime $p \leq \tau = tn \log tn$
- ▶ Alice transmits $F_p(a) = a \bmod p$ and p to Bob
- ▶ Bob computes $F_p(b)$: Returns “equal” iff $F_p(a) = F_p(b)$

Theorem

Protocol uses $O(\log(tn))$ bits of communication and is correct with probability $1 - O(1/t)$.

Verifying Equality of Strings: Analysis

- ▶ If $a = b$ then $F_p(a) = F_p(b)$

Verifying Equality of Strings: Analysis

- ▶ If $a = b$ then $F_p(a) = F_p(b)$
- ▶ If $a \neq b$ and $F_p(a) = F_p(b)$ then p divides $-2^n < a - b < 2^n$

Verifying Equality of Strings: Analysis

- ▶ If $a = b$ then $F_p(a) = F_p(b)$
- ▶ If $a \neq b$ and $F_p(a) = F_p(b)$ then p divides $-2^n < a - b < 2^n$

Fact

*There are at most n distinct prime dividing a number less than 2^n .
For any τ , the number of primes smaller than τ is $\pi(\tau) \sim \tau / \ln \tau$.*

Verifying Equality of Strings: Analysis

- ▶ If $a = b$ then $F_p(a) = F_p(b)$
- ▶ If $a \neq b$ and $F_p(a) = F_p(b)$ then p divides $-2^n < a - b < 2^n$

Fact

*There are at most n distinct prime dividing a number less than 2^n .
For any τ , the number of primes smaller than τ is $\pi(\tau) \sim \tau / \ln \tau$.*

- ▶ If $a \neq b$ then

$$\mathbb{P}[F_p(a) = F_p(b)] \leq \frac{n}{\pi(\tau)} = O\left(\frac{n \ln(tn)}{tn \log(tn)}\right) = O\left(\frac{1}{t}\right)$$

Verifying Equality of Strings: What about deterministic?

Theorem

Any deterministic protocol that involves one message from Alice to Bob requires n bits of communication.

Verifying Equality of Strings: What about deterministic?

Theorem

Any deterministic protocol that involves one message from Alice to Bob requires n bits of communication.

Proof.

- ▶ A length $k < n$ message m from Alice partitions set strings into 2^k sets:

$$S_m = \{a' : f(a') = m\}$$



Verifying Equality of Strings: What about deterministic?

Theorem

Any deterministic protocol that involves one message from Alice to Bob requires n bits of communication.

Proof.

- ▶ A length $k < n$ message m from Alice partitions set strings into 2^k sets:

$$S_m = \{a' : f(a') = m\}$$

- ▶ There exists a set S_m that has at least $2^{n-k} \geq 2$ elements.



Verifying Equality of Strings: What about deterministic?

Theorem

Any deterministic protocol that involves one message from Alice to Bob requires n bits of communication.

Proof.

- ▶ A length $k < n$ message m from Alice partitions set strings into 2^k sets:

$$S_m = \{a' : f(a') = m\}$$

- ▶ There exists a set S_m that has at least $2^{n-k} \geq 2$ elements.
- ▶ Let $a \in S_m$ and $b \in S_m$: Impossible for Bob to tell if $a = b$



Greater Than (1/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Greater Than (1/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Let $a[i, j] = (a_i, \dots, a_j)$ and $b[i, j] = (b_i, \dots, b_j)$. If we find max j with $a[1, j] = b[1, j]$ then value of a_{j+1} or b_{j+1} determines answer.

Greater Than (1/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Let $a[i, j] = (a_i, \dots, a_j)$ and $b[i, j] = (b_i, \dots, b_j)$. If we find max j with $a[1, j] = b[1, j]$ then value of a_{j+1} or b_{j+1} determines answer.

Protocol

- ▶ 1st message: Determine if $a[1, n/2] = b[1, n/2]$
- ▶ If equal: 2nd message determines if $a[1, 3n/4] = b[1, 3n/4]$
- ▶ If not equal: 2nd message determines if $a[1, n/4] = b[1, n/4]$

Greater Than (1/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Let $a[i, j] = (a_i, \dots, a_j)$ and $b[i, j] = (b_i, \dots, b_j)$. If we find max j with $a[1, j] = b[1, j]$ then value of a_{j+1} or b_{j+1} determines answer.

Protocol

- ▶ 1st message: Determine if $a[1, n/2] = b[1, n/2]$
- ▶ If equal: 2nd message determines if $a[1, 3n/4] = b[1, 3n/4]$
- ▶ If not equal: 2nd message determines if $a[1, n/4] = b[1, n/4]$
- ▶ Continue binary search in this manner...

Greater Than (1/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Let $a[i, j] = (a_i, \dots, a_j)$ and $b[i, j] = (b_i, \dots, b_j)$. If we find max j with $a[1, j] = b[1, j]$ then value of a_{j+1} or b_{j+1} determines answer.

Protocol

- ▶ 1st message: Determine if $a[1, n/2] = b[1, n/2]$
- ▶ If equal: 2nd message determines if $a[1, 3n/4] = b[1, 3n/4]$
- ▶ If not equal: 2nd message determines if $a[1, n/4] = b[1, n/4]$
- ▶ Continue binary search in this manner...

Theorem

Protocol uses $O(\log(tn) \log n)$ bits of communication and is correct with probability $1 - O((\log n)/t)$.

Greater Than (2/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

- ▶ Protocol uses $O(n^{1/p} p \log(tn))$ bits of communication and is correct with probability $1 - O(n^{1/p} p/t)$.

Greater Than (2/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Problem

What if we are only allowed p messages passed back and forth?

- ▶ Protocol uses $O(n^{1/p} p \log(tn))$ bits of communication and is correct with probability $1 - O(n^{1/p} p/t)$.

Greater Than (2/2)

Problem

Alice and Bob have non-equal binary strings (a_1, \dots, a_n) and (b_1, \dots, b_n) . Is $\sum_{i \in [n]} a_i 2^{n-i} < \sum_{i \in [n]} b_i 2^{n-i}$?

Problem

What if we are only allowed p messages passed back and forth?

- ▶ Instead of testing $a[1, n/2] = b[1, n/2]$, test

$$a[1, n^{1/p}] = b[1, n^{1/p}]$$

$$a[n^{1/p} + 1, 2n^{1/p}] = b[n^{1/p} + 1, 2n^{1/p}]$$

⋮

$$a[n - 2n^{1/p} + 1, n - n^{1/p}] = b[n - 2n^{1/p} + 1, n - n^{1/p}]$$

- ▶ Protocol uses $O(n^{1/p} p \log(tn))$ bits of communication and is correct with probability $1 - O(n^{1/p} p/t)$.

Other Communication Complexity Problems

Other Communication Complexity Problems

Theorem (Razborov 1990)

If Alice has $x \in \{0, 1\}^n$ and Bob has $y \in \{0, 1\}^n$, then determining if there exists i such that $x_i = y_i = 1$ with probability $9/10$ requires $\Theta(n)$ bits of communication.

Other Communication Complexity Problems

Theorem (Razborov 1990)

If Alice has $x \in \{0, 1\}^n$ and Bob has $y \in \{0, 1\}^n$, then determining if there exists i such that $x_i = y_i = 1$ with probability $9/10$ requires $\Theta(n)$ bits of communication.

Theorem (Brody and Charikrabarti 2009)

If Alice has $x \in \{0, 1\}^n$ and Bob has $y \in \{0, 1\}^n$, then determining Hamming distance up to additive error \sqrt{n} with probability $9/10$ requires $\Theta(n)$ bits of communication.

Outline

Schwartz-Zippel

String Equality and Some Communication Complexity

Pattern Matching

Readings

Pattern Matching (1/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$
(assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

Pattern Matching (1/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

- ▶ Define $X(j) = x_jx_{j+1} \dots x_{j+m-1}$

Pattern Matching (1/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

- ▶ Define $X(j) = x_jx_{j+1} \dots x_{j+m-1}$
- ▶ Brute force: test $X(j) = Y$ for each j in $O(mn)$ time

Pattern Matching (1/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

- ▶ Define $X(j) = x_jx_{j+1} \dots x_{j+m-1}$
- ▶ Brute force: test $X(j) = Y$ for each j in $O(mn)$ time
- ▶ Viewing $X(j)$ as integer $\sum_{i \in [m]} x_{j+m-i}2^{i-1}$

$$X(j+1) = 2[X(j) - 2^{m-1}x_j] + x_{j+m}$$

Pattern Matching (1/2)

Problem

Given “text” $X = x_1x_2 \dots x_n$ and “pattern” $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

- ▶ Define $X(j) = x_jx_{j+1} \dots x_{j+m-1}$
- ▶ Brute force: test $X(j) = Y$ for each j in $O(mn)$ time
- ▶ Viewing $X(j)$ as integer $\sum_{i \in [m]} x_{j+m-i}2^{i-1}$

$$X(j+1) = 2[X(j) - 2^{m-1}x_j] + x_{j+m}$$

- ▶ Therefore

$$F_p(X(j+1)) = 2[F_p(X(j)) - 2^{m-1}x_j] + x_{j+m} \pmod{p}$$

where p is some prime less than $\tau = n^2 m \log n^2 m$.

Pattern Matching (2/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

Theorem

Can perform pattern matching in $O(n + m)$ time with probability of false match $O(1/n)$

Pattern Matching (2/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

Theorem

Can perform pattern matching in $O(n + m)$ time with probability of false match $O(1/n)$

- ▶ Compute $F_p(X(j))$ for all j and $F_p(Y)$ in $O(n + m)$ time.

Pattern Matching (2/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

Theorem

Can perform pattern matching in $O(n + m)$ time with probability of false match $O(1/n)$

- ▶ Compute $F_p(X(j))$ for all j and $F_p(Y)$ in $O(n + m)$ time.
- ▶ Probability of false match of $X(j)$ and Y is $O(1/n^2)$

Pattern Matching (2/2)

Problem

Given "text" $X = x_1x_2 \dots x_n$ and "pattern" $Y = y_1y_2 \dots y_m$ (assume binary and $m < n$). Does there exist j with

$$x_jx_{j+1} \dots x_{j+m-1} = y_1y_2 \dots y_m$$

Theorem

Can perform pattern matching in $O(n + m)$ time with probability of false match $O(1/n)$

- ▶ Compute $F_p(X(j))$ for all j and $F_p(Y)$ in $O(n + m)$ time.
- ▶ Probability of false match of $X(j)$ and Y is $O(1/n^2)$
- ▶ Final error probability follows by union bound.

Outline

Schwartz-Zippel

String Equality and Some Communication Complexity

Pattern Matching

Readings

Readings

For next time, please make sure you've read:

- ▶ Chapter 7–7.3 [MR].