6 May 2024

Program Checking.

Plan

* Program Checking
* Announcements
* Matrix Multiplication

In 4820

* Designed Algorithms for computational problems
  - Proof of correctness essential

In 4820

* Designed Algorithms for computational problems

  ↳ Proof of correctness essential

Outside of 4820

* May encounter programs w/ bugs.

> Can we make use of programs that may make mistakes?

Program Checking:

Given:

* a program P (as a black-box) that is supposed to compute function f.
* an input x

Program Checking:

Given:

* a program P (as a black-box) that is supposed to compute function f.
* an input x

Check:

Does $x \to \boxed{P} \to z = f(x)$?

* If $P(x) = f(x)$, accept
* If $P(x) \neq f(x)$, reject with high probability.

Check: Does $x \rightarrow \boxed{P} \rightarrow z = f(x)$?

* If $P(x) = f(x)$, accept
* If $P(x) \neq f(x)$, reject with high probability.

Note.

* Check should run faster than computing f
* Check should use randomness

Announcements

* FINAL EXAM. 16 May 2024, 7p.
  ↳ Watch Ed for any announcements
* OH. Check the online calendar.
  ↳ TODAY: My OH @ 2p
* Fill out Course Evals! by May 10th

Matrix Multiplication

$$A \cdot B = C$$

Matrix Multiplication

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & & & \vdots \\ \vdots & & & \vdots \\ A_{n1} & \cdots & & A_{nn} \end{bmatrix} \qquad B = \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1n} \\ B_{21} & & & \vdots \\ \vdots & & & \vdots \\ B_{n1} & \cdots & & B_{nn} \end{bmatrix}$$

$$A \cdot B = \begin{bmatrix} \langle \tilde{A}_{1}, B_1 \rangle & \langle \tilde{A}_{1}, B_2 \rangle & \cdots & \langle \tilde{A}_{1}, B_n \rangle \\ \langle \hat{A}_{2}, B_1 \rangle & & & \vdots \\ \vdots & & & \vdots \\ \langle \hat{A}_{n}, B_1 \rangle & \cdots & & \langle \tilde{A}_{n}, B_n \rangle \end{bmatrix}$$

where $\langle \hat{A}_i, B_j \rangle = \sum_{k=1}^{n} A_{ik} \cdot B_{kj}$

Matrix Multiplication

$A \cdot B$ // involves n^2 inner products

$$\sum_{k=1}^{n} A_{ik} \cdot B_{kj}$$

Naive Algo. $O(n^3)$ operations

Fastest Algo. $O(n^\omega)$ operations

where $\omega < 2.373$

Checking Matrix Mult

Given: Program P, $n \times n$ matrices A, B

$$P(A, B) \longrightarrow C$$

supposedly equal to $A \cdot B$.

Can we check that

$$C = A \cdot B$$

without running Matrix Mult?

Matrix - Vector Multiplication

$$\begin{bmatrix} — & \tilde{M}_1 & — \\ — & \tilde{M}_2 & — \\ & \vdots & \\ — & \tilde{M}_n & — \end{bmatrix} \cdot \begin{bmatrix} | \\ v \\ | \end{bmatrix} = \begin{bmatrix} \langle \tilde{M}_1, v \rangle \\ \langle \tilde{M}_2, v \rangle \\ \vdots \\ \langle \tilde{M}_n, v \rangle \end{bmatrix}$$

<u>Naive Algo</u>: $O(n^2)$ operations

Can we <u>check</u> Matrix Mult using calls to Matrix-Vector Mult?

Ideas?

$$A \qquad B = \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \cdots & b_n \\ | & | & & | \end{bmatrix}$$

$$j \sim [n].$$

$$A \cdot b_j \overset{?}{=} c_j \qquad n^2$$

$$\Omega(n)$$

$$C_{ij} \overset{?}{=} (AB)_{ij} = \langle \tilde{A}_i, B_j \rangle \qquad n$$

$$\Omega(n^2)$$

Ideas?

If $C = A \cdot B$, then

$$C \cdot r = \underbrace{(A \cdot B) \cdot r}$$

for all r!

$$\underbrace{A \, \underbrace{(B \cdot r)}}_{\text{MvMv}}$$

Random "spot checks" (aka Frievalds' Algo)

Given: A, B, and C supposedly equal to $A \cdot B$

Random "spot checks" (aka Frievalds' Algo)

Given: A, B, and C supposedly equal to $A \cdot B$

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

$$x \leftarrow C \cdot r$$

$$y \leftarrow B \cdot r$$

$$z \leftarrow A \cdot y$$

- if $x \neq z$, REJECT

Random "spot checks" (aka Frievalds' Algo)

Given: A, B, and C supposedly equal to $A \cdot B$

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

$$x \leftarrow C \cdot r$$
$$y \leftarrow B \cdot r$$
$$z \leftarrow A \cdot y$$

- if $x \neq z$, REJECT

Claim: for every $r \in \{0,1\}^n$ $\quad z = (A \cdot B) \cdot r$

Random "spot checks" (aka Frievalds' Algo)

Given: A, B, and C supposedly equal to $A \cdot B$

Repeat T times:

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

$$x \leftarrow C \cdot r$$
$$y \leftarrow B \cdot r$$
$$z \leftarrow A \cdot y$$

- if $x \neq z$, REJECT

ACCEPT

Repeat T times.

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

  $x \leftarrow C \cdot r$

  $y \leftarrow B \cdot r$

  $z \leftarrow A \cdot y$
- if $x \neq z$, REJECT

ACCEPT

Claim. If $C = A \cdot B$, then the checker ACCEPTS with probability 1.

Repeat T times.

- sample $r \leftarrow \{0,1\}^n$ uniformly at random
- Let

$$x \leftarrow C \cdot r$$
$$y \leftarrow B \cdot r$$
$$z \leftarrow A \cdot y$$

- if $x \neq z$, REJECT

ACCEPT

Suppose $C \neq A \cdot B$.

What is the probability Checker REJECTs?

Claim. If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n} \left[C \cdot r \neq A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

<u>Claim.</u> If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n} \left[C \cdot r \neq A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

<u>Corollary.</u> If $C \neq A \cdot B$, then

$$\Pr\left[\text{Checker ACCEPTs} \right] \leq \frac{1}{2^T}.$$

Claim. If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n}\left[C \cdot r = A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

Pf. By the Principle of Deferred Decisions.

Claim. If $C \neq A \cdot B$, then

$$\Pr_{r \leftarrow \{0,1\}^n}\left[C \cdot r = A \cdot B \cdot r \right] \geq \frac{1}{2}.$$

Pf. By the Principle of Deferred Decisions.

If $C \neq A \cdot B$, then there exists i, j s.t.

$$C_{ij} \neq (A \cdot B)_{ij}$$

$\Longrightarrow$ the j^{th} entry of r is crucial.

$$C_{ij} \neq (A \cdot B)_{ij}$$

Let $U = \sum_{k=1}^{n} C_{ik} \cdot r_k$ $\qquad$ $V = \sum_{k=1}^{n} (A \cdot B)_{ik} \cdot r_k$

$$C_{ij} \neq (A \cdot B)_{ij}$$

Let $U = \sum_{k=1}^{n} C_{ik} \cdot r_k$ $\qquad V = \sum_{k=1}^{n} (A \cdot B)_{ik} \cdot r_k$

Imagine we "defer" flipping the j^{th} bit of r until all others have been flipped.

$$r_1, r_2, \ldots r_{j-1}, r_{j+1} \ldots r_n$$

r_j?

(still random)

$$C_{ij} \neq (A \cdot B)_{ij}$$

Let $$U = \sum_{k=1}^{n} C_{ik} \cdot r_k \qquad V = \sum_{k=1}^{n} (A \cdot B)_{ik} \cdot r_k$$

Imagine we "defer" flipping the j^{th} bit of r until all others have been flipped.

$$r_1, r_2, \ldots r_{j-1}, r_{j+1} \ldots r_n$$

r_j?

(still random)

$$U_{\setminus j} = \sum_{k \neq j} C_{ik} \cdot r_k \qquad V_{\setminus j} = \sum_{k \neq j} (A \cdot B)_{ik} \cdot r_k$$

$$C_{ij} \neq (A \cdot B)_{ij}$$

$$U = U_{ij} + C_{ij} \cdot r_j \qquad V = V_{ij} + (A \cdot B)_{ij} \cdot r_j$$

$$\boxed{\text{What is } \Pr_{r_j \in \{0,1\}}[U \neq V] \text{ ?}}$$

$$C_{ij} \neq (A \cdot B)_{ij}$$

$$U = U_{ij} + C_{ij} \cdot r_j \qquad V = V_{ij} + (A \cdot B)_{ij} \cdot r_j$$

$$\boxed{\text{What is } \Pr_{r_j \in \{0,1\}}[U \neq V] \text{ ?}}$$

Case 1 $\quad U_{ij} = V_{ij}$

$$\Pr_{r_j \in \{0,1\}}[U \neq V] = \Pr[r_j = 1] = 1/2$$

$$C_{ij} \neq (A \cdot B)_{ij}$$

$$U = U_{ij} + C_{ij} \cdot r_j \qquad V = V_{ij} + (A \cdot B)_{ij} \cdot r_j$$

$$\boxed{\text{What is } \Pr_{r_j \in \{0,1\}}[U \neq V] \text{ ?}}$$

Case (1) $U_{ij} = V_{ij}$

$$\Pr_{r_j \in \{0,1\}}[U \neq V] = \Pr[r_j = 1] = 1/2$$

Case (2) $U_{ij} \neq V_{ij}$

$$\Pr_{r_j \in \{0,1\}}[U \neq V] \geq \Pr[r_j = 0] = 1/2$$

So.

* Checker always ACCEPTS when $C = A \cdot B$

* Checker REJECTS w.p. $1 - 1/2^T$ when $C \neq A \cdot B$

* Checker runs $3T$ Matrix-vector mults.

So.

* Checker always ACCEPTs when $C = A \cdot B$

* Checker REJECTs w.p. $1 - 1/2^T$ when $C \neq A \cdot B$

* Checker runs $3T$ Matrix-vector mults.

e.g. $T = 10 \log n$.

- Checker runs in $O(n^2 \log n)$ time

- Makes mistake with probability at most $1/n^{10}$.

Where do you go from here?

* More Algo? CS 6820

* More Hardness of Computation?
  ↳ Complexity Theory CS 4814

* Both?
  ↳ Cryptography CS 4830

Thanks for a great semester!