CMPT 210: Probability and Computation

Lecture 10

Sharan Vaswani

June 10, 2022

Matrix Multiplication

Given two $n \times n$ matrices – A and B, if $C = AB$, then,

$$C_{i,j} = \sum_{k=1}^{n} A_{i,k} B_{k,j}$$

Hence, in the worst case, computing $C_{i,j}$ is an $O(n)$ operation. There are n^2 entries to fill in C and hence, in the absence of additional structure, matrix multiplication takes $O(n^3)$ time.

There are non-trivial algorithms for doing matrix multiplication more efficiently:

- (Strassen, 1969) Requires $O(n^{2.81})$ operations.
- (Coppersmith-Winograd, 1987) Requires $O(n^{2.376})$ operations.
- (Alman-Williams, 2020) Requires $O(n^{2.373})$ operations.
- Belief is that it can be done in time $O(n^{2+\epsilon})$ for $\epsilon > 0$.

Verifying Matrix Multiplication

For simplicity, we will focus on A, B being binary matrices (all entries are either 0 or 1), and matrix multiplication mod 2, i.e. $C_{i,j} = (\sum_{k=1}^{n} A_{i,k} B_{k,j}) \bmod 2$, implying that C is a binary matrix.

Example: $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ then $C = AB = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

Objective: Verify whether a matrix multiplication operation is correct.

Trivial way: Do the matrix multiplication ourselves, and verify it using $O(n^3)$ (or $O(n^{2.373})$) operations.

Frievald's Algorithm: Randomized algorithm to verify matrix multiplication with high probability in $O(n^2)$ time.

(Basic) Frievald's Algorithm

For $n \times n$ matrices A, B and D, is $D = AB (\text{mod } 2)$?

1. Generate a random n-bit vector x, by making each bit x_i either 0 or 1 *independently* with probability $\frac{1}{2}$. E.g, for $n = 2$, toss a fair coin independently twice with the scheme – H is 0 and T is 1). If we get HT, then set $x = [0\,;\,1]$.
2. Compute $t = Bx (\text{mod } 2)$ and $y = At = A(Bx)(\text{mod } 2)$ and $z = Dx (\text{mod } 2)$.
3. Output "yes" if $y = z$ (all entries need to be equal), else output "no".

Computational complexity: Step 1 can be done in $O(n)$ time. Step 2 requires 3 matrix vector multiplications and can be done in $O(n^2)$ time. Step 3 requires comparing two n-dimensional vectors and can be done in $O(n)$ time. Hence, the total computational complexity is $O(n^2)$.

(Basic) Frievald's Algorithm

Let us run the algorithm on an example. Suppose we have generated $x = [1\,;\,0]$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad ; \quad D = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$Bx = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad ; \quad y = A\,(Bx) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad ; \quad z = Dx = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Hence the algorithm will correctly output "no" since $D \neq AB (\text{mod } 2)$.

Q: Suppose we have generated $x = [1\,;\,1]$. What is y and z? Ans: $y = [0\,;\,1]$ and $z = [0\,;\,1]$.

In this case, $y = z$ and the algorithm will incorrectly output "yes" even though $D \neq AB (\text{mod } 2)$.

(Basic) Frievald's Algorithm

Let us run the algorithm on an example. Suppose we have generated $x = [1\,;\,0]$.

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad ; \quad C = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Bx = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad ; \quad y = A\,(Bx) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad ; \quad z = Cx = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Hence the algorithm will correctly output "yes" since $C = AB (\text{mod } 2)$.

Q: Suppose we have generated $x = [1\,;\,1]$. What is y and z? Ans: $y = [0\,;\,1]$ and $z = [0\,;\,1]$.

In this case again, $y = z$ and the algorithm will correctly output "yes".

(Basic) Frievald's Algorithm

Let us analyze the algorithm for general matrix multiplication (not necessarily (mod 2)).

Case (i): If $D = AB$, does the algorithm always output "yes"? Yes! Since $D = AB$, for any vector x, $Dx = ABx$.

Case (ii) If $D \neq AB$, does the algorithm output "no"?

Claim: For any input matrices A, B, D if $D \neq AB$, then the (Basic) Frievald's algorithm will output "no" with probability $\geq \frac{1}{2}$.

	Yes	No
$D = AB$	1	0
$D \neq AB$	$< \frac{1}{2}$	$\geq \frac{1}{2}$

Table 1: Probabilities for Basic Frievalds Algorithm

(Basic) Frievald's Algorithm

If $D \neq AB$, we wish to compute the probability that algorithm outputs "yes". Define $E := (AB - D)$ and $r := Ex = (AB - D)x = y - z$. If $D \neq AB$, then $\exists (i,j)$ s.t. $E_{i,j} \neq 0$.

$$\Pr[\text{Algorithm outputs "yes"}] = \Pr[y = z] = \Pr[r = \mathbf{0}] = \Pr[(r_1 = 0) \cap (r_2 = 0) \cap \ldots \cap (r_i = 0) \cap \ldots]$$

$$= \Pr[(r_i = 0)] \Pr[(r_1 = 0) \cap (r_2 = 0) \cap \ldots \cap (r_n = 0) | r_i = 0] \leq \Pr[r_i = 0]$$

$$r_i = \sum_{k=1}^{n} E_{i,k} x_k = E_{i,j} x_j + \sum_{k \neq j} E_{i,k} x_k = E_{i,j} x_j + \omega \qquad (\omega := \textstyle\sum_{k \neq j} E_{i,k} x_k)$$

$$\Pr[r_i = 0] = \Pr[r_i = 0 | \omega = 0] \Pr[\omega = 0] + \Pr[r_i = 0 | \omega \neq 0] \Pr[\omega \neq 0]$$

$$\Pr[r_i = 0 | \omega = 0] = \Pr[x_j = 0] = \frac{1}{2}$$

$$\Pr[r_i = 0 | \omega \neq 0] = \Pr[(x_j = 1) \cap E_{i,j} = -\omega] = \Pr[(x_j = 1)] \Pr[E_{i,j} = -\omega | x_j = 1] \leq \Pr[(x_j = 1)] = \frac{1}{2}$$

$$\implies \Pr[r_i = 0] \leq \frac{1}{2} \Pr[\omega = 0] + \frac{1}{2} \Pr[\omega \neq 0] = \frac{1}{2} \Pr[\omega = 0] + \frac{1}{2} [1 - \Pr[\omega = 0]] = \frac{1}{2}$$

$$\implies \Pr[\text{Algorithm outputs "yes"}] \leq \Pr[r_i = 0] \leq \frac{1}{2}.$$

(Basic) Frievald's Algorithm

Hence, if $D \neq AB$, the Algorithm outputs "yes" with probability $\leq \frac{1}{2} \implies$ the Algorithm outputs "no" with probability $\geq \frac{1}{2}$.

In the worst case, the algorithm can be incorrect half the time! We promised the algorithm would return the correct answer with "high" probability close to 1.

A common trick in randomized algorithms is to have m independent trials of an algorithm and aggregate the answer in some way, reducing the probability of error, thus *amplifying the success probability*.

Questions?

Frievald's Algorithm

By repeating the *Basic Frievald's Algorithm* (from slide 7) m times, we will amplify the probability of success. The resulting complete Frievald's Algorithm is given by:

1 Run the Basic Frievald's Algorithm for m independent runs.

2 If *any* run of the Basic Frievald's Algorithm outputs "no", output "no".

3 If *all* runs of the Basic Frievald's Algorithm output "yes", output "yes".

	Yes	No
$D = AB$	1	0
$D \neq AB$	$< \frac{1}{2^m}$	$\geq 1 - \frac{1}{2^m}$

Table 2: Probabilities for Frievald's Algorithm

If $m = 20$, then Frievald's algorithm will make mistake with probability $1/2^{20} \approx 10^{-6}$.

Computational Complexity: $O(mn^2)$

Probability Amplification

Consider a randomized algorithm $\mathcal{A}$ that is supposed to solve a binary decision problem i.e. it is supposed to answer either Yes or No. It has a one-sided error – (i) if the true answer is Yes, then the algorithm $\mathcal{A}$ correctly outputs Yes with probability 1, but (ii) if the true answer is No, the algorithm $\mathcal{A}$ incorrectly outputs Yes with probability $\leq \frac{1}{2}$.

Let us define a new algorithm $\mathcal{B}$ that runs algorithm $\mathcal{A}$ m times, and if *any* run of $\mathcal{A}$ outputs No, algorithm $\mathcal{B}$ outputs No. If *all* runs of $\mathcal{A}$ output Yes, algorithm $\mathcal{B}$ outputs Yes.

Q: What is the probability that algorithm $\mathcal{B}$ correctly outputs Yes if the true answer is Yes, and correctly outputs No if the true answer is No?

Probability Amplification - Analysis

$$
\begin{aligned}
&\Pr[\mathcal{B} \text{ outputs Yes} \mid \text{true answer is Yes}] \\
&= \Pr[\mathcal{A}_1 \text{ outputs Yes } \cap \mathcal{A}_2 \text{ outputs Yes } \cap \ldots \cap \mathcal{A}_m \text{ outputs Yes} \mid \text{true answer is Yes}] \\
&= \prod_{i=1}^{m} \Pr[\mathcal{A}_i \text{ outputs Yes} \mid \text{true answer is Yes}] = 1 \qquad \text{(Independence of runs)} \\
&\Pr[\mathcal{B} \text{ outputs No} \mid \text{true answer is No}] \\
&= 1 - \Pr[\mathcal{B} \text{ outputs Yes} \mid \text{true answer is No}] \\
&= 1 - \Pr[\mathcal{A}_1 \text{ outputs Yes } \cap \mathcal{A}_2 \text{ outputs Yes } \cap \ldots \cap \mathcal{A}_m \text{ outputs Yes} \mid \text{true answer is No}] \\
&= 1 - \prod_{i=1}^{m} \Pr[\mathcal{A}_i \text{ outputs Yes} \mid \text{true answer is No}] \geq 1 - \frac{1}{2^m}.
\end{aligned}
$$

When the true answer is Yes, both $\mathcal{B}$ and $\mathcal{A}$ correctly output Yes. When the true answer is No, $\mathcal{A}$ incorrectly outputs Yes with probability $< \frac{1}{2}$, but $\mathcal{B}$ incorrectly outputs Yes with probability $< \frac{1}{2^m} << \frac{1}{2}$. By repeating the experiment, we have "amplified" the probability of success.

Questions?